www.phmsociety.org

ISBN - 978-1-936263-06-6

PHM 2013

Proceedings of the Annual Conference of the Prognostics and Health Management Society 2013

New Orleans LA
October 14-17, 2013

ISBN - 978-1-936263-06-6

Edited by:
Shankar Sankararaman
Indranil Roychoudhury

phmsociety

Table of Contents

Full Papers

2 A Particle Filtering-based Framework for Real-time Fault Diagnosis of Autonomous Vehicles
Ioannis A. Raptis, Christopher Sconyers, Rodney Martin, Robert Mah, Nikunj Oza, Dimitris Mavris, and George Vachtsevanos

11 Maintenance Planning Optimization Based on PHM Information and Spare Parts Availability
Leonardo R. Rodrigues and Takashi Yoneyama

18 Rotor Track and Balance Improvements
Eric Bechhoefer, Austin Fang, and Ephrahim Garcia

27 Wireless Power and Data System for Integrated System Health Management of Systems Operating in the Harsh Environment of Deep Space
Jim Miller, Jon Patterson, and Ray Garbos

33 Processing for Improved Spectral Analysis
Eric Bechhoefer, Brandon Van Hecke, and David He

39 Lubricants Health Monitoring
Giovanni Jacazio, Marco Libraro, Andrea Mornacchi, and Massimo Sorli

49 Wear Prognostic on Turbofan Engines
Jerome Lacaille, Aurelie Gouby, and Olivier Piol

57 Electronic Returnless Fuel System Fault Diagnosis and Isolation: A Data-Driven Approach
Bharath Pattipati, Krishna Pattipati, Youssef A. Ghoneim, Mark Howell, and Mutasim Salman

66 Accommodating Repair Actions into Gas Turbine Prognostics
Zakwan Skaf, Martha A. Zaidan, Robert F. Harrison, and Andrew R. Mills

74 System Level RUL Estimation for Multiple-Component Systems
Joao P. P. Gomes, Leonardo R. Rodrigues, Roberto K. H. Galvao, and Takashi Yoneyama

83 A Stochastic Modeling Approach of Quantized Systems with Application to Fault Detection and Isolation of an Automotive Electrical Power Generation Storage System
Sara Mohon and Pierluigi Pisu

93 On Optimizing Anomaly Detection Rules for Gas Turbine Health Monitoring
Weizhong Yan, Lijie Yu, Jim Sherbahn, and Umang Brahmakshatriya

100 Development of Virtual Sensor Networks to Support Accident Monitoring System
Rizwan Ahmed, Pak Sukyoung, Gyungyoung Heo, Jung-Taek Kim, Seop Hur, Soo Y. Park, and Kwang-Il Ahn

107 Modeling Localized Bearing Faults Using Inverse Gaussian Mixtures
Pavle Boskoski and Dani Juricic

114 A Cryogenic Fluid System Simulation in Support of Integrated Systems Health Management
John P. Barber, Kyle B. Johnston, and Matthew Daigle

127 Anomaly Detection in Gas Turbine Compressor of a Power Generation Plant using Similarity-based Modeling and Multivariate Analysis
Tomas Carricajo, Felipe Kripper, Marcos E. Orchard, Luis Yacher, and Rodrigo Paredes

134 I/P Transducer Application of Model-BasedWear Detection and Estimation using Steady State Conditions
Christopher Teubert and Matthew Daigle

141 Probabilistic Delamination Diagnosis of Composite Materials Using a Novel Bayesian Imaging Method
Tishun Peng, Abhinav Saxena, Kai Goebel, Shankar Sankararaman, Yibing Xiang, and Yongming Liu

150 Heat Exchanger Fouling and Estimation of Remaining Useful Life
Tutpol Ardsomang, Wesley J. Hines, and Belle R. Upadhyaya

159 Practical Use of Accelerated Test Data for the Prognostics Methods
Dawn An, Joo-Ho Choi, and Nam H. Kim

167 Dynamic Bayesian Networks for Prognosis
Gregory Bartram and Sankaran Mahadevan

185 Fault Detection and Isolation for Autonomous Parafoils
Matthew R. Stoeckle, Amer Fejzic, Louis S. Breger, and Jonathan P. How

199 Ensemble Classifiers for Drift Detection and Monitoring in Dynamical Environments
Imen Khamassi, Moamar Sayed-Mouchaweh, Moez Hammani, and Khaled Ghedira

213 Distribution Free Prediction Interval for Uncertainty Quantification in Remaining Useful Life Prediction
Huimin Chen

220 Distilling the Verification Process for Prognostics Algorithms
Indranil Roychoudhury, Abhinav Saxena, Jose R. Celaya, and Kai Goebel

230 A Method for Anomaly Detection for Non-Stationary Vibration Signatures
Renata Klein

237 Online Monitoring of Plant Assets in the Nuclear Industry
Vivek Agarwal, Nancy Lybeck, Binh T. Pham, Richard Rusaw, and Randall Bickford

249 Electrochemistry-based Battery Modeling for Prognostics
Matthew Daigle and Chetan Kulkarni

262 Advanced Methods for Determining Prediction Uncertainty in Model-Based Prognostics with Application to Planetary Rovers
Matthew Daigle and Shankar Sankararaman

275 Improving the Diagnostic Performance for Dynamic Systems through the use of Conflict-Driven Model Decomposition
Anibal Bregon, Alexander Feldman, Belarmino Pulido, Gregory Provan, and Carlos Alonso-Gonzalez

288 Efficient Generation of Minimal Dynamic Bayesian Networks for Hybrid Systems Fault Diagnosis using Hybrid Possible Conflicts
Belarmino Pulido, Noemi Moya, Carlos Alonso-Gonzalez, and Anibal Bregon

300 Health Monitoring and Prognosis of Hybrid Systems
Said Zabi, Pauline Ribot, and Elodie Chanthery

312 An Efficient Model-based Diagnosis Engine for Hybrid Systems using Structural Model Decomposition
Anibal Bregon, Sriram Narasimhan, Indranil Roychoudhury, Matthew Daigle, and Belarmino Pulido

325 SIL/HIL Replication of Electric Aircraft Powertrain Dynamics and Inner-Loop Control for V&V of System Health Management Routines
Brian Bole, Christopher Teubert, Quach C. Chi, Edward Hogge, Sixto Vasquez, Kai Goebel, and George Vachtsevanos

337 Why is the Remaining Useful Life Prediction Uncertain?
Shankar Sankararaman and Kai Goebel

350 A Novel Computational Methodology for Uncertainty Quantification in Prognostics Using The Most Probable Point Concept
Shankar Sankararaman and Kai Goebel

363 An Energy-Based Prognostic Framework to Predict Fatigue Damage Evolution in Composites
Juan Chiachio, Manuel Chiachio, Abhinav Saxena, Guillermo Rus, and Kai Goebel

372 Integrated Sensor Arrays Based on PiezoPaint$^{\text{TM}}$Technology for SHM Applications
Karl Elkjaer, Konstantin Astafiev, Erling Ringgard, and Tomasz Zawada

381 Towards Real-Time, On-Board, Hardware-Supported Sensor and Software Health Management for Unmanned Aerial Systems
Johann Schumann, Kristin Y. Rozier, Thomas Reinbacher, Ole J. Mengshoel, Timmy Mbaya, and Corey Ippolito

402 Degradation Modeling and Remaining Useful Life Prediction of Electrolytic Capacitors under Thermal Overstress Condition Using Particle Filters
Hamed Khorasgani, Chetan Kulkarni, Gautam Biswas, Jose R. Celaya, and Kai Goebel

412 A Framework to Debug Diagnostic Matrices
Anuradha Kodali, Peter Robinson, and Ann Patterson-Hine

419 Prognostic Algorithm Verification
N. Scott Clements and David S. Bodden

430 Uncertainty in Impact Identification Applied to a Commercial Wind Turbine Blade
Raymond M. Bond and Douglas E. Adams

438 A Model-Based Approach for Predicting the Remaining Driving Range in Electric Vehicles
Javier A. Oliva, Christoph Weihrauch, and Torsten Bertram

449 A Modeling Framework for Prognostic Decision Making and its Application to UAV Mission Planning
Edward Balaban and Juan J. Alonso

461 Novelty Detection in Airport Baggage Conveyor Gear-Motors Using Synchro-Squeezing Transform and Self-Organizing Maps
Budhaditya Hazra, Shilpa Pantula, and Sriram Narasimhan

469 Towards an Online Prognostic System for Predicting the Axial Shrinkage of AGR Cores
Graeme M. West, Christopher J. Wallace, and Stephen D. J. McArthur

478 Physics based Prognostics for LCF Crack Nucleation Life of IMI 685 Aerospace Compressor Disc
Avisekh Banerjee, Jun Zhao, Ashok K. Koul, Amar Kumar, Alka Srivastava, and Nita Goel

486 A New Prognostic Approach for Hydro-Generators Stator Windings
Normand Amyot, Claude Hudon, Melanie Levesque, Mario Belec, France Brabant, and Francois-Xavier Frenette

492 Determination of Primary Chemical Constituents of PBX (AF)-108 Warhead Explosive Using microPHAZIR^{TM}NIR Handheld Platform
Sami Daoud, Michael J. Villeburn, Kevin D. Bailey, Gordon Kinloch, Lydia Biegert, and Craig Gardner

503 Battery Charge Depletion Prediction on an Electric Aircraft
Quach C. Chi, Brian Bole, Edward Hogge, Sixto Vasquez, Matthew Daigle, Jose R. Celaya, Adam Weber, and Kai Goebel

513 State of Charge Estimation of Lithium-ion Batteries Considering Model and Parameter Uncertainties
Zhimin Xi, Rong Jing, Xiao G. Yang, and Ed Decker

521 Fleet-wide Diagnostic and Prognostic Assessment
Alexandre Voisin, Gabriela Medina-Oliva, Maxime Monnin, Jean-Baptiste Leger, and Benoit Iung

531 Prognostics Health Management for Advanced Small Modular Reactor Passive Components
Ryan M. Meyer, Pradeep Ramuhalli, Jamie B. Coble, Mark R. Mitchell, David W. Wootan, Evelyn H. Hirt, Eric J. Berglin, Leonard J. Bond, and Chuck H. Henager Jr

540 Detection of CH-53 Swashplate Bearing Deformation - From a 3D Dynamic Model to Diagnostics
Mor Battat, Gideon Kogan, Alex Kushnirsky, Renata Klein, and Jacob Bortman

548 Multivariate Fault Detection using Vector Autoregressive Moving Average and Orthogonal Transformation in Residual Space
Francisco Serdio, Edwin Lughofer, Kurt Pichler, Thomas Buchegger, Markus Pichler, and Hajrudin Efendic

556 Simple Metrics for Evaluating and Conveying Prognostic Model Performance To Users With Varied Backgrounds
Michael E. Sharp

566 PHM for Astronauts - A New Application
Alexandre Popov, Wolfgang Fink, and Andrew Hess

573 Fault Monitoring Techniques for Nuclear Components
Gee-Yong Park and Jung T. Kim

Poster Papers

581 Signal Processing Techniques to Improve an Acoustic Emissions Sensor
Eric Bechhoefer, Yongzhi Qu, Junda Zhu, and David He

589 Decision Layer by Fusion of Diagnostic Algorithms
Jerome Lacaille and Tsirizo Rabenoro

598 A Fusion Framework with Nonlinear Degradation Improvement for Remaining Useful Life Estimation of Lithium-ion Batteries
Datong Liu, Limeng Guo, Jingyue Pang, and Yu Peng

608 Remaining Useful Life Prognostics for Lithium-ion Battery Based on Gaussian Processing Regression Combined with the Empirical Model
Shan Yin, Jingyue Pang, Datong Liu, and Yu Peng

616 Particle Filter Prognostic Applied in Landing Gear Retraction
Paula B. O. Cerdeira, Roberto K. H. Galvao, and Joao P. P. Malere

624 Filter Clogging Data Collection for Prognostics
O. F. Eker, F. Camci, and I. K. Jennions

633 Post Implementation CBM Benefit Analysis – U. S. Army AH-64D Apache Helicopter Main Transmission Accessory Sprag Clutch Endurance Project
Matthew C. Carter

637 Bringing the Aircraft System Management Tool to Life using the Informatica Tool Suite
Glenn Peters, Mark Mosher, and Chris Carlson

642 Options for Prognostics Methods: A Review of Data-Driven and Physics-Based Prognostics
Dawn An, Nam H. Kim, and Joo-Ho Choi

656 Experimental Tests of Wind Turbine Main Shaft Motion on a Laboratory Test Rig
Dirk Lutschinger and Ian Howard

664 Estimation of APU Failure Parameters Employing Linear Regression and Neural Networks
Renata M. Pascoal, Wlamir O. L. Vianna, Joao P. P. Gomes, and Roberto K. H. Galvao

671 Web-based Prognostics and 24/7 Monitoring
Miryam Strautkalns and Peter Robinson

677 Predictive Modeling of High-Bypass Turbofan Engine Deterioration
Christina Brasco, Neil Eklund, Mohak Shah, and Daniel Marthaler

685 Analysis of Data Quality Issues in Real-world Industrial Data
Thomas Hubauer, Steffen Lamparter, Mikhail Roshchin, Nina Solomakhina, and Stuart Watson

694 Learning Diagnoser and Supervision Pattern in Discrete Event System : Application to Crisis Management
Moussa Traore, Moamar Sayed-Mouchaweh, and Patrice Billauder

703 **Author Index**

Full Papers

A Particle Filtering-based Framework for Real-time Fault Diagnosis of Autonomous Vehicles

Ioannis A. Raptis[1], Christopher Sconyers[2], Rodney Martin[3], Robert Mah[4],
Nikunj Oza[5], Dimitris Mavris[6], and George J. Vachtsevanos[7]

[1] *University of Massachusetts Lowell, Lowell, Massachusetts, 01854, USA*
Ioannis_Raptis@uml.edu

[2,6,7] *Georgia Institute of Technology, Atlanta, Georgia, 30332, USA*
csconyers@gatech.edu
dimitri.mavris@aerospace.gatech.edu
george.vachtsevanos@ece.gatech.edu

[3,4,5] *NASA Ames Research Center, Moffett Field, California, 94035, USA*
Rodney.martin@nasa.gov
Robert.w.mah@nasa.gov
Nikunj.c.oza@nasa.gov

Abstract

Reliability and supervision of autonomous systems and their corresponding subsystems, can be significantly improved using advanced methods of anomaly detection and fault diagnosis. A reliable fault detection module can improve the autonomy of the vehicle itself, by leading to efficient fault tolerant control designs and mission scheduling. This paper presents a fault detection framework for incipient faults that take place on the actuators of an autonomous-small scale hovercraft. The enabling technologies borrow from the fields of modeling, data processing, Bayesian estimation theory and in particular a technique called particle filtering. Condition indicators or features are derived based on first principles modeling of the actuator's and the vehicle's dynamics. The fault detection algorithm fuses information obtained from different subsystems of the vehicle that use distinct sets of sensors, producing a robust degree of confidence. In addition, the algorithm is decoupled from the control system. This achieves the goal of minimizing the fault masking, or the compensation of fault-induced navigational errors by the control system, and allows for early detection even for small fault conditions. The efficacy of the diagnostic approach is demonstrated via simulation results. The proposed fault detection methodology can be easily leveraged to other types of autonomous vehicles.

1. Introduction

Autonomous vehicles have seen unprecedent levels of growth over the last decade. Although autonomous vehicles have been mainly used for military applications, there is a considerable and increasing interest for civilian applications as well. It is not an exaggeration to consider that as the technology matures and as autonomous vehicles become cost-effective with proven reliability and safety, civilian applications will dominate the field.

The practical motivation for pursuing this objective stems from the need to address an area of concern in the aviation industry that relates to power asymmetry resulting from a slow power loss, stuck throttle, or no response to throttle coupled with automatic controls. Flying aids, such as the auto-pilot and auto-throttle, can mask significant power asymmetry until a control limit is reached. At this point, the flight crew has to intervene, understand the malfunction, and assume control of an airplane which may be in an upset condition. Better indications and/or annunciations of power asymmetry could warn crews in advance and allow them time to identify the problem and apply the appropriate procedures (Sallee & Gibbons, 1998). Although this problem has in part been solved for a newer generation model B-787 capable of providing the pilot proprioceptive feedback based upon the Improved Thrust Asymmetry Compensation System (ITACS), older airplanes without ITACS are still going to remain in use for quite a long time so this scenario is still relevant (*http://aviationknowledge.wikidot.com/aviation:ergonomics-*

Christopher Sconyers et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

of-cae-7000-series-simulator-configured, n.d.). Thus, emulation of an asymmetric thrust event that causes a gradual drift of important system level safety parameters that go undetected by an operator/pilot due to compensation by a robust control algorithm is of great importance.

Specific to the issues of automatic controls masking the effects of degradations in thrust, there have been at least two accidents in recent years relating to Thrust Management Systems (TMS) effects. In both scenarios, data from the ensuing investigations have provided evidence that it is incorrect to assume that the flight crew will always detect and address potentially adverse TMS effects strictly from inherent operational cues. On March 31, 1995 in Bucharest, Hungary a Tarom Airbus Model A310-300 airplane crashed shortly after takeoff. The Romanian investigating team indicated that the probable cause of the accident was the combination of an auto throttle failure that generated asymmetric thrust and the pilot's apparent failure to react quickly enough to the developing emergency. On November 24, 1992 in Guilin, China, a China Southern Boeing Model 737-300 airplane crashed shortly before landing at Guilin. The Civil Aviation Administration of China team investigating the probable cause of the accident concluded that the right auto throttle did not react during descent and level off. As a result, the thrust asymmetry induced the airplane to roll to the right. The flight crew failed to recognize the abnormality and make correction in time, "followed by wrongful control input and crashed (Statement, 2022)." These two examples provide sufficient and defensible evidence that there is a pragmatic, safety-critical need for the results of the research problem to be investigated in this paper. Thus, it is important to recognize that the research efforts described in this paper are not purely academic in nature or purely as an exercise to demonstrate the utility of a particular approach.

Although most attention is given to ground and aerial vehicles, autonomous surface vessels are an emerging research topic. A special representative of the surface vessels family is the hovercraft. The main characteristic of the hovercraft is its ability to travel on different types of surfaces. This is achieved by sustaining a cushion between the hull of the vehicle and the surface with low-pressure air. The vehicle is moving almost frictionlessly since an air-lubricated layer is created between the cushion and the ground surface.

The main component that is responsible for providing the vehicle with autonomy during the execution of a mission, is the control system. An important requirement of advanced control systems is reliability. Reliability requirements necessitate that the control system is tolerant to incipient faults that affect the system. A key component of a fault tolerant control system is the anomaly detector. An anomaly detector is a module of the Prognostics and Health Management system intended to recognize as early as possible abnormal conditions or anomalies in the operation of a monitored system. In most real applications, an anomaly detector is required to perform this task while minimizing both the probability of false alarms and the detection time (time between the initiation of a fault and its detection), given a fixed threshold for false positives (Orchard, 2007; Zhang et al., 2011). In the case of multiple faults or alternate fault types, it is assumed that the fault signature indicating the fault of interest, a change in a single fan motor resistance value, is distinct from other fault signatures.

A block diagram illustrating the complete architecture that involves the control system with the detection algorithm can be seen in Figure 1. Additionally, Figure 2 shows the inner motor control loops that regulate fan speed and stabilize the thrust output for each motor. The combination of these inner control loops and the outer navigation control is what primarily masks the fault. Autonomous vehicles are typically large-scale systems composed of several subsystems. Each subsystem has its own set of sensors with varying accuracies. The proposed fault detection algorithm is capable of merging information from different subsystems of the autonomous vehicle. Finally, the detection scheme can be easily applied to other autonomous vehicle with different actuating mechanisms.

This paper presents the implementation of an online particle-filtering-based framework for fault diagnosis in a autonomous robotic hovercraft. The hovercraft under consideration is actuated by two independent thrusters. The fault to be detected is the increase to the effective resistance value of the motor that operates one of the thrusters. This fault is a gradual degradation of component health within the thruster motor and is chosen to allow the diagnostic tools time to analyze the fault and any available reconfigurable control scheme to adapt to the fault beyond the scope of immediate catastrophic failure. This resistance value is internal to the motor and can not be easily measured during operation of the hovercraft. A variety of components ranging from sensors and actuators to communication, computing, and control apparatus may experience a severe fault mode. We are focusing in this study on a thruster fault mode, i.e. driving motor insulation breakdown, which limits the operational capability of one of the two thrusters, resulting in a deviation of the vehicle from its designated trajectory. Moreover, the selected fault mode is masked by control loops and can be easily simulated or injected in the actual vehicle. The faulty thruster is producing a reduced thrust value compared to its healthy performance. The methodology considers a diagnostic model, and assumes the existence of fault indicators (for monitoring purposes) and the availability of real-time measurements. The fault detection module uses a hybrid state-space model of the system and a particle filtering algorithm to calculate the probability of detection of the fault, simultaneously computing the fault probability density function (PDF) estimates. The enabling technologies borrow from the fields of modeling, data processing, Bayesian estimation theory and in particular a technique called particle filtering. Once this information is available within the fault detection module, it is conveniently processed to generate proper fault alarms and to inform about the statistical confidence of the detection routine. These statistical outputs render the proposed algorithm significantly superior to classical approaches such as limit or trend checking that do not give a deeper insight into the fault diagnosis.

The fault detection algorithm can be executed on-line and in parallel with the control system. The decoupling of the fault detection routine from the control system allows an early detection, preventing the control system from “masking” the fault. By “masking” we express the suppression of the deviant system behavior–caused by the fault growth–from the control system (even for small fault levels). Therefore, the effects of the fault can be efficiently distinguished from external disturbances that act on the vehicle.

This paper is organized as follows: Section 2 presents the hovercraft motion dynamics and the model of the thrusters' motors. In Section 3 the generic description of the fault growth models is given. The description of the features (or condition indicators), used in the detection algorithm, are given in Section 4. The technical approach of the detection algorithm is presented in Section 5. Results in the form of numerical simulations are given in Section 6. Finally, concluding remarks are given in Section 7.

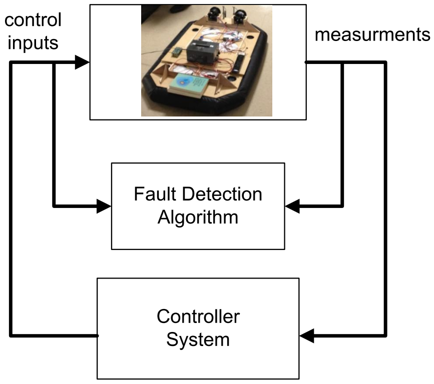


Figure 1. Block diagram illustrating the architecture of the control system enhanced with the proposed diagnostic algorithm.

2. Hovercraft Model

The hovercraft under consideration is actuated by two independent uni-directional thrusters that are symmetrically located with respect to the median plane of the vehicle. This configuration is generating the hovercraft's surge force and yaw moment. Since there is no direct control input applied to the sway motion the hovercraft is classified as an underactuated system. The input of each thruster is a voltage signal that controls an electrical motor. The motor speed is operating the thruster's propeller that generates the propulsion force. The hovercraft model can be divided into two subsystems. The first subsystem is related with the force and moment generation process. The second subsystem is associated with the hovercraft's motion dynamics. The two subsystems and their connections are shown in Figure 3.

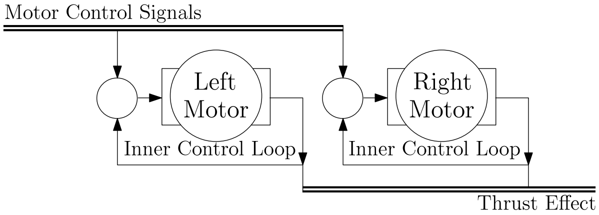


Figure 2. Inner motor control loops responsible for motor fault masking.

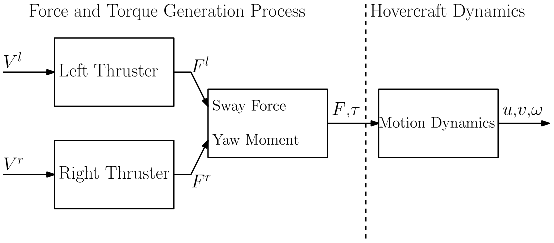


Figure 3. Hovercraft model subsystems

2.1. Hovercraft Dynamics

The first step towards the development of the hovercraft's equations of motion is the definition of two reference frames. Each frame is characterized by its center and three mutually orthonormal vectors. The first one is the inertial frame defined as $\mathcal{F}_I = \left\{O_I, \vec{i}_I, \vec{j}_I, \vec{k}_I\right\}$. A typical convention of the inertial frame is the North-East-Down system where $\vec{i}_I$ points North, $\vec{j}_I$ points East and $\vec{k}_I$ points at the center of the Earth. The second frame is the body-fixed reference frame defined as $\mathcal{F}_B = \left\{O_B, \vec{i}_B, \vec{j}_B, \vec{k}_B\right\}$, where the center O_B is located at the Center of Gravity (CG) of the hovercraft. The vector $\vec{i}_B$ points forward, $\vec{j}_B$ points at the aft right side of the hovercraft and $\vec{k}_B$ points downward such that $\left\{\vec{i}_B, \vec{j}_B, \vec{k}_B\right\}$ constitutes a right handed Cartesian coordinate frame ($\vec{k}_B = \vec{i}_B \times \vec{j}_B$). The direction of the body fixed frame orthonormal vectors $\left\{\vec{i}_B, \vec{j}_B, \vec{k}_B\right\}$ is shown in Figure 4.

We consider only the planar 2-D motion of the vehicle disregarding the pitch, roll, and heave motion. Denote the hovercraft angular velocity by ω and the surge (forward) and sway (lateral) velocities by u and v, respectively. From Figure 4, the hovercraft dynamic equations, with respect to the body

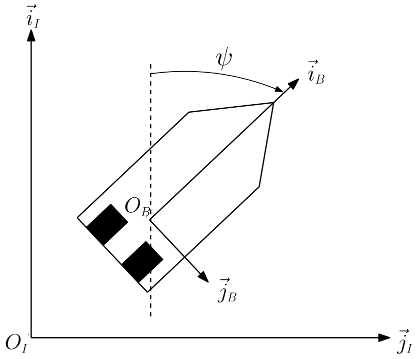


Figure 4. Direction of the hovercraft body fixed frame orthonormal vectors.

fixed frame, can be derived as

$$\begin{aligned} \dot{u} &= v\omega + F/m + n \\ \dot{v} &= -u\omega + n \\ \dot{\omega} &= \tau/\mathcal{I} \end{aligned} \tag{1}$$

where F denotes the net surge force, τ the net yaw moment, m is the mass of the hovercraft, $\mathcal{I}$ is the inertia of the hovercraft (assuming symmetry with respect to the principal axis), and n is additive noise due to disturbances.

2.2. Force and Torque Generation

Let f_l, f_r denote the thrust force of the left and right fan respectively. The association of the net surge force and yaw moment with the thrusts of the two fans are given by a linear one-to-one mapping, namely

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ d & -d \end{bmatrix} \begin{bmatrix} f_r \\ f_l \end{bmatrix} \tag{2}$$

where d is a parameter related to the moment arm of each force. The two propulsion thrusts are produced by two identical fans that are operated by common motors. The last step of the modeling process is to include a simplified model of these motors. A reasonable assumption is to consider the motor dynamics significantly faster compared to the motion dynamics of the vehicle. Therefore the motors can be represented by a static instead of a dynamic model. In this section we are going to present the generic model of each motor. The distinction between the left and right motor will be take place by using the lower scripts l and r in each case. Denote by V the voltage applied to the fan motor. This voltage is the output from the control system. From standard results, the electrical part of the motor is described by the following equation:

$$V = I_m R_m + V_{emf} \tag{3}$$

where I_m is the motor current, R_m is the effective internal motor resistance and V_{emf} is the back-emf voltage of the motor. The back emf voltage is given by

$$V_{emf} = K_\Omega \Omega \tag{4}$$

where Ω is the angular velocity of the motor and K_Ω is a positive constant. From the mechanical part of the motor, the produced torque is given by

$$T = K_t I_m - b\Omega \tag{5}$$

where T is the produced torque, b is the friction coefficient of the motor and K_t a positive constant. Finally, we assume that the produced thrust is proportional to the angular velocity Ω. Namely, we assume a linear relationship at the desired operating conditions;

$$F = K_f \Omega \tag{6}$$

where K_f is a positive constant. K_t and K_f are identified as part of the system ID in Section 2.3. From (3)-(6) the input-output description of the thrusters is given by

$$F = \Pi\left(R_m\right) V \tag{7}$$

where

$$\Pi\left(R_m\right) = \frac{K_t K_f}{bR_m + K_t K_\Omega} \tag{8}$$

The available measurements are all the states related to the motion of the vehicle (u, v, ω), the applied voltages to each motor (V^l, V^r) and the produced currents (I_m^l, I_m^r). Since the produced current of each motor is considered a measured quantity, the current-voltage mapping is required by the FDI approach. Therefore,

$$I_m = \frac{b}{bR_m + K_t K_\Omega} V \tag{9}$$

2.3. Simulation-Hardware Matching and System ID

A collaboration software framework is adapted for simulation and visualization studies. The vehicle dynamics are simulated in Gazebo. Rviz is used for the visualization of the vehicle's motion/environment accompanied by sensor/navigation relevant information. ROS is the operating system running the vehicle's software routines. The selected simulation/visualization environment facilitates troubleshooting and debugging of the control algorithms before installation on the actual platform, exploits realistic sensor and actuator models exhibiting the limitations of available measurements. An attempt is made to cope as accurately as possible with uncertainty observed on the actual vehicle.

A dynamic environment overview of the ROS/Gazebo simulator is shown in Figure 5.

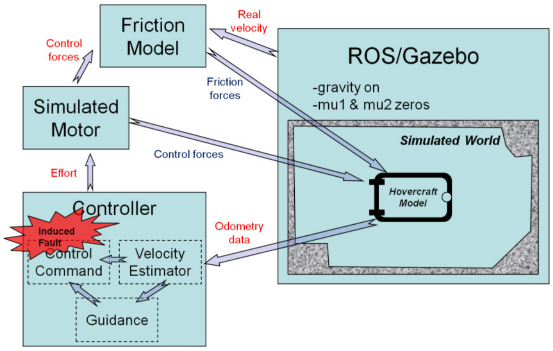


Figure 5. Dynamic environment overview of the ROS/Gazebo simulator.

The Gazebo hovercraft model incorporates component weights as measured individually. The friction model presents significant difficulties because of the inherent uncertainty associated with a moving hovercraft—the skirt/environment interface. Translational coefficients are estimated from strain gauge measurements while damping coefficients are tuned on the basis of hardware test data.

Hovercraft tests focused on way-point following (five way points) with a line-of-sight guidance law and a dynamic inversion control scheme. Several forward maneuvers were tested with a varying effect for several seconds. Acceleration, deceleration, and velocity rates were recorded. Simulation model parameters were tuned using a mixture of system ID and parameter optimization tools to maximize agreement between the simulation model and the hovercraft behavior.

Simulation versus test results are shown in Figure 6. Distance and velocity profiles for various percentage efforts are compared between simulated and experimental results. Good agreement is observed between the two cases.

Finally, 7 shows a comparison of test and simulation results in the way point trajectory following controller, using a five point trajectory that generates response for each unique hovercraft maneuver (left turn, right turn, straight). Discrepancies are attributed to parameter (such as friction) uncertainties, disturbances impacting the vehicle, and other unmodeled vehicle dynamics and environmental effects. As more test data is accumulated, the parameters are fine-tuned to better match experimental and simulation results.

3. Fault Model

The fault under consideration is increase of the effective internal resistance value of one of the motors, a value that cannot be measured directly without dismantling the motor. Additional simultaneous faults, including the case of a similar fault occurring on both motors, are not considered in this paper. The proposed analysis involves faults that are monotonically increasing functions of the load conditions. In this case study the load variable is the faulty motor current I_m. Therefore, the generic growth rate of the fault under consideration is given by the following differential equation:

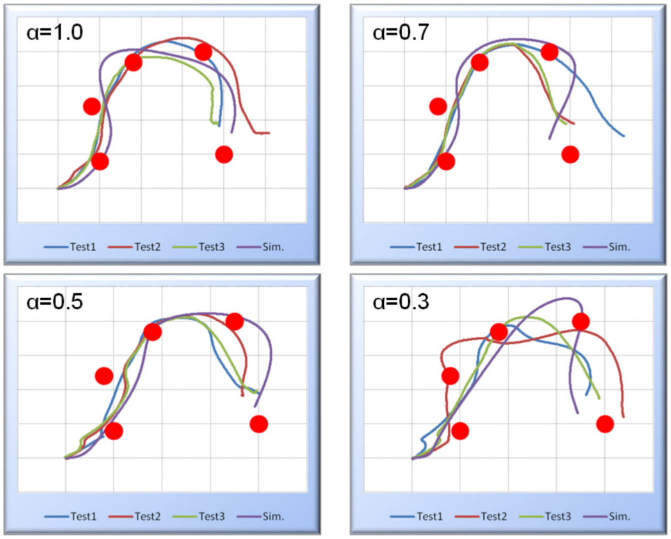


Figure 7. Example of trajectory following tests, simulation versus test results. α is a tuned parameter representing friction.

$$\dot{R}_m^f = g(t, R_m^f, I_m) \tag{10}$$

with $R_m^f(t_f) = R_m$. By t_f we denote the time instant that the fault occurs, while R_m^f is the value of the faulty resistance. Furthermore, $g(t, R_m^f, I_m) > 0\,\forall t, R_m^f, I_m \in \Re_+$. The latter condition guarantees that the fault value is non-decreasing over time. Hence, the faulty resistance can be written as $R_m^f(t) = R_m + \Delta R_m^f(t)$, where R_m is the healthy value of the resistance and $\Delta R_m^f(t) \geq 0 \quad \forall t \geq t_f$.

4. Feature Extraction

Feature or condition indicator selection and extraction constitutes the cornerstone for accurate and reliable fault diagnosis. A feature or condition indicator is an extracted value from a signal that describes the status of the process that fault diagnosis is applied to. Fault diagnosis depends mainly on extracting a set of features from sensor data that can distinguish between fault classes of interest, detect and isolate a particular fault at its early initiation stages. Feature evaluation and selection metrics include the monotonicity of the relationship between the feature and the true fault size and the variance (or covariance) of the feature at discrete fault levels compared to the feature range. A feature is sufficient if it shows a similar growth pattern to that of the ground truth data.

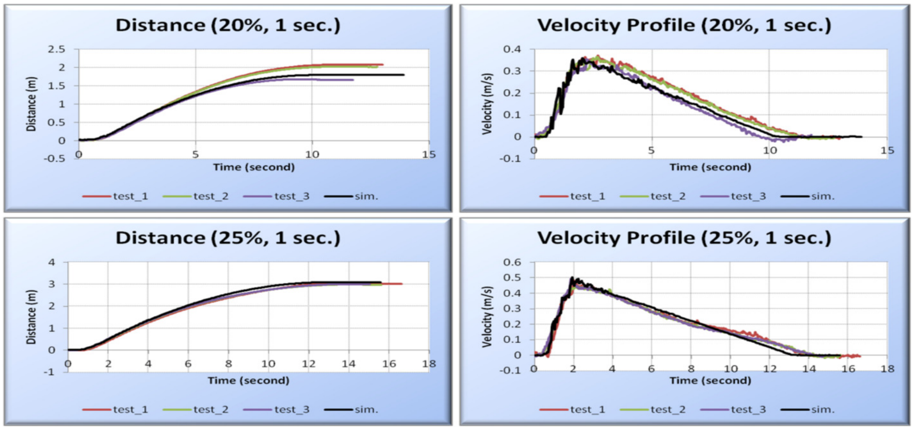


Figure 6. Hovercraft performance, simulated versus test results.

With the possibility of alternate fault types or of multiple simultaneous faults, it is assumed that a feature or set of features may be designed to identify only the fault of interest. Alternatively, a single feature may accomodate a set of similar fault types, such as a set of faults that have similar characteristics and effects at the component or system levels. For this paper, only one fault is being examined.

As indicated earlier the fault under consideration is the change in the resistance value of one of the two motors. From Section 2, the hovercraft model is composed by two interconnected subsystems: the force/moment generation and the motion dynamics subsystems. The faulty resistance affects the force/moment generation directly, and sequentially the vehicle's motion. The goal of this paper is to use features extracted from signals generated by both of these subsystems. The first feature, that belongs to the force/moment generation subsystem, is the resistance value itself. In particular, rearranging (9) one has

$$f_{em}(R_m^f) = R_m^f = \frac{V}{I_m} - \frac{K_\omega K_t}{b} \tag{11}$$

The second feature is derived by the motion dynamic subsystems. Both of the two features presented are based on the dynamic equation of motion of the vehicle given in (1). Combining (1), (2) and (7) the dynamics of the surge velocity can be written as

$$\dot{u} = v\omega + \left(\Pi\left(R_m^l\right)V^l + \Pi(R_m^r)V^r\right)/m \tag{12}$$

Assuming that we monitor the left motor for a fault ($R_m^f\left(0\right) = R_m = R_m^l$) and considering the above equation, the second feature is

$$f_{cr1}(R_m^f) = \Pi\left(R_m^f\right) = \frac{m\left(\dot{u} - v\omega\right) - \Pi(R_m^r)V^r}{V^l} \tag{13}$$

The second feature is the mapping from the voltage-to-thrust. This feature is valid only when $V^l \neq 0$ indicating the intuitive notion that the faulty motor must be operating in order to diagnose the fault. Similarly the dynamics of the angular motion are given by

$$\dot{\omega} = \frac{d}{\mathcal{I}}\left(\Pi\left(R_m^l\right)V^l - \Pi(R_m^r)V^r\right) \tag{14}$$

The voltage-to-thrust mapping can be also derived by the angular motion as well. More specifically,

$$f_{cr2}(R_m^f) = \Pi\left(R_m^f\right) = \frac{(\mathcal{I}/d)\,\dot{\omega} + \Pi(R_m^r)V^r}{V^l} \tag{15}$$

The features f_{cr1} and f_{cr2} are monitoring the same quantity. However, in each case different sensors are used. The goal of this paper is to conduct fault diagnosis by monitoring two subsystems of the vehicle. Therefore, from a theoretical perspective, either the tuple (f_{em}, f_{cr1}) or (f_{em}, f_{cr2}) can be used by the detection algorithm. Typically, for the sensing of the vehicle's motion an Inertial Measurement Unit (IMU) is used. In such cases, it is preferable to monitor motion variables related to the angular motion of the vehicle since they typically have better accuracy compared to variables related to the linear motion and are less affected by disturbances and varying environment conditions. The Advanced Systems Design Laboratory at the Georgia Institute of Technology has assembled a small-scale autonomous hovercraft platform in order to serve as a testbed for the evaluation of the algorithm. For indoor operation of the vehicle, the variables related with the motion of the vehicle are obtained by a LIDAR sensor using SLAM. Measurements from the GPS and IMU can be taken and combined with the LIDAR measurements for improved accuracy. The measurements related with the thrust generation are obtained by the PC unit onboard the vehicle. The autonomous hovercraft can be seen in Figure 8.

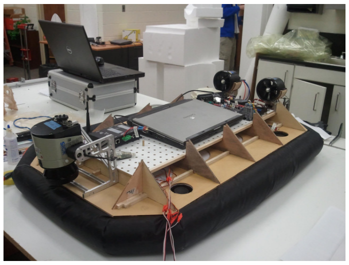


Figure 8. This figure illustrates the small-scale autonomous hovercraft developed by the Advanced Systems Design Laboratory at the Georgia Institute of Technology.

5. The Diagnostic Algorithm

A fault diagnosis procedure involves the tasks of fault detection and isolation (FDI), and fault identification (assessment of the severity of the fault). In general, this procedure may be interpreted as the fusion and utilization of the information present in a feature vector (measurements), with the objective of determining the operating condition (state) of a system and the causes for deviations from particularly desired behavioral patterns. Several ways to categorize FDI techniques can be found in literature. FDI techniques are classified according to the way that data is used to describe the behavior of the system: data-driven or model-based approaches.

Data-driven FDI techniques usually rely on signal processing and knowledge-based methodologies to extract the information hidden in the feature vector (also referred to as measurements). In this case, the classification/prediction procedure may be performed on the basis of variables that have little (or sometimes completely lack of) physical meaning. On the other hand, model-based techniques, as the name implies, use a description of a system (models based on first principles or physical laws) to determine the current operating condition.

A compromise between both classes of FDI techniques is often needed when dealing with complex nonlinear systems, given the difficulty of collecting useful faulty data (a critical aspect in any data-driven FDI approach) and the expertise needed to build a reliable model of the monitored system (a key issue in a model-based FDI approach).

From a nonlinear Bayesian state estimation standpoint, this compromise between data-driven and model-based techniques may be accomplished by the use of a Particle Filter (PF) based module built upon the dynamic state model describing the time progression or evolution of the fault (Orchard, 2007; Orchard & Vachtsevanos, 2007, 2009). The fault progression is often nonlinear and, consequently, the model should be nonlinear as well. Thus, the diagnostic model is described by:

$$\begin{aligned} x_d\left(t+1\right) &= f_d\left(x_d\left(t\right), n\left(t\right)\right) \\ x_c\left(t+1\right) &= f_t\left(x_d\left(t\right), x_c\left(t\right), \omega\left(t\right)\right) \\ y\left(t\right) &= h_t\left(x_d(t), x_c\left(t\right), v\left(t\right)\right) \end{aligned}$$

where f_d, f_t, and h_t are nonlinear mappings, $x_d\left(t\right)$ is a collection of Boolean states associated with the presence of a particular operating condition in the system (normal operation and multiple fault types), $x_c\left(t\right)$ is a set of continuous-valued states that describe the evolution of the system given those operating conditions, $y(t)$ denotes the available measurements, $\omega\left(t\right)$ and $v\left(t\right)$ are non-Gaussian distributions that characterize the process and feature noise signals including disturbances, respectively. Since the noise signal $n\left(t\right)$ is a measure of uncertainty associated with Boolean states, it is advantageous to define its probability density through a random variable with bounded domain. For simplicity, $n\left(t\right)$ may be assumed to be additive uniform white noise (Orchard, 2007). The PF approach using the above model allows statistical characterization of both Boolean and continuous-valued states, as new feature data (measurements) are received. As a result, at any given instant of time, this framework provides an estimate of the probability densities associated with each fault mode, as well as a PDF estimate for meaningful physical variables in the system. Hypothesis testing through calculating current and baseline PDFs is used to generate fault alarms, and other statistical analysis tools may be used to extract additional information about the detection and diagnostic results (discussed further in Section 6). One particular advantage of the proposed particle filtering approach is the ability to characterize the evolution in time of the above mentioned nonlinear model through modification of the probability masses associated with each particle, as new data from fault indicators are received.

The PF based FDI module is implemented accordingly using the non-linear time growth model given in (10) to describe the faulty motor's resistance value. The goal is for the algorithm to make an early detection of the increase to the effective resistance value (leading to an open-circuit). Two main operating conditions are distinguished: The normal condition reflects the fact that there is no fault in the motor while a faulty condition indicating an unexpected growth to the resistance value. Denote by $x_{d,1}$ and $x_{d,2}$ two Boolean states that indicate normal and faulty conditions respectively. The nonlinear model is given by:

$$\begin{aligned} \left[\begin{array}{c} x_{d,1}(t+1) \\ x_{d,2}(t+1) \end{array}\right] &= f_b\left(\left[\begin{array}{c} x_{d,1}(t) \\ x_{d,2}(t) \end{array}\right] + n(t)\right) \\ \dot{R}_m^f(t) &= x_{d,2}(t)g(t, R_m^f, I_m) + \omega(t) \qquad (16) \\ y(t) &= h(R_m^f(t)) + v(t) \end{aligned}$$

where

$$f_b(x) = \begin{cases} [1 \quad 0]^T & \text{if } \left\| x - [1 \quad 0]^T \right\| \leq \left\| x - [0 \quad 1]^T \right\| \\ [0 \quad 1]^T & \text{else} \end{cases} \tag{17}$$

$$\left[x_{d,1}(t_o) \quad x_{d,2}(t_o) \quad R_m^f(t_o)\right]^T = [1 \quad 0 \quad R_m]^T$$
$$h(R_m^f(t)) = \left[f_{em}(R_m^f) \quad f_{cr1}(R_m^f)\right]^T$$

In the above equations $R_m$is the initial healthy value of the resistance. In (17), the condition indicators $x_{d,1}(t)$ and $x_{d,2}(t)$, after the addition of noise $n(t)$, are thresholded to restrict them to Boolean values, with the possibility of changing to new values at $t+1$. The above system can be written in a more compact form as:

$$\mathcal{X}(t+1) = \Phi\left(t, \mathcal{X}(t), \mathcal{N}(t+1)\right) \tag{18}$$
$$\mathcal{Y}(t) = H\left(\mathcal{X}(t), t\right) + v(t) \tag{19}$$

where $\mathcal{X} = \left[x_{d,1} \quad x_{d,2} \quad R_m^f\right]^T$, $\mathcal{Y} = [f_{em} \quad f_{cr1}]^T$ and $\mathcal{V} = [n \quad \omega]^T$. The steps of the PF algorithm execution are described below:

1. From (18) generate N state estimates (particles) denoted by $\hat{\mathcal{X}}^i(t)$ where $i = 1, \ldots, N$.
2. From (19) calculate the feature estimates, substituting the particles $\hat{\mathcal{X}}^i(t+1)$ to the mapping $H(\cdot)$.
3. Calculate the N errors $\varepsilon^i = H\left(\hat{\mathcal{X}}(t), t\right) - \mathcal{Y}$ with $\varepsilon^i = \left[\varepsilon_1^i \quad \varepsilon_2^i\right]^T$, and assign to each particle $\hat{\mathcal{X}}^i(t)$ a weight $w^i(t) = \phi\left(\varepsilon_1^i\right)\phi\left(\varepsilon_2^i\right)$, where $\phi(\cdot)$ denotes the standard normal distribution.
4. Normalize the weights $w^i(t)$. The normalized weights $\bar{w}^i(t)$ represent the discrete probability masses of each state estimate.
5. Calculate the final state estimate $\widetilde{\mathcal{X}}(t)$ using the weighted sum of all the states $\hat{\mathcal{X}}^i(t)$.

An important part of the PF algorithm is the resampling procedure. Resampling is an action that takes place to counteract the degeneracy of the particles caused by estimates that have very low weights. A block diagram of the PF algorithm is given in Figure 9.

6. Simulation Results

The performance of the proposed FDI algorithm was tested via numerical simulations. The hovercraft dynamics are described in (1) and the thrusters model in (3)-(7). The resistance fault is seeded to the left motor according to (10). The actual fault can be seen in Figure 10. The vehicles parameters used for the simulation are summarized in Table 1. In real-life applications these parameters can be obtained by the thruster specifications and simple system identification tests.

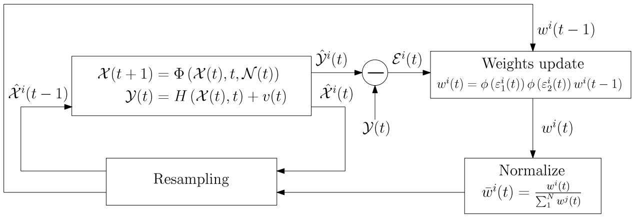

$w^i(t-1)$
$\hat{\mathcal{X}}^i(t-1)$
$\mathcal{X}(t+1) = \Phi\left(\mathcal{X}(t), t, \mathcal{N}(t)\right)$
$\mathcal{Y}(t) = H\left(\mathcal{X}(t), t\right) + v(t)$
$\hat{\mathcal{Y}}^i(t)$
$\hat{\mathcal{X}}^i(t)$
$\mathcal{E}^i(t)$
$\mathcal{Y}(t)$
Weights update
$w^i(t) = \phi\left(\varepsilon_1^i(t)\right)\phi\left(\varepsilon_2^i(t)\right)w^i(t-1)$
$w^i(t)$
Resampling
Normalize
$\bar{w}^i(t) = \frac{w^i(t)}{\sum_1^N w^j(t)}$

Figure 9. Block diagram of the PF algorithm for state estimation

Test experiments were designed to closely mimic a fault continuum over time. Initially, a fault is not present in the system for the purposes of establishing a baseline. At $t = 100sec$, the fault is seeded with an initial value close to zero and grows at a constant rate until, at $t = 200sec$, the fault growth ceases and the vehicle behavior and feature characteristics can be analyzed at a static fault level.

With the desired vehicle trajectory given, position and orientation of the hovercraft is extracted from a simulated LIDAR/SLAM visual odometry sensor. Motor control signals are extracted from the motor control software.

The number of particles used for the estimator were $N = 100$. The estimator fault value can be seen in Figure 10. Although the diagnostic particle filter retains an accurate estimate of the fault after the maximum fault level is reached ($t >= 200sec$), the upper and lower feature bounds exhibit large spikes as the control algorithm fails to compensate for the motor fault during vehicle maneuvers.

Besides detecting the faulty condition, it is desired to obtain some measure of the statistical confidence of the alarm signal. For this reason, an additional output will be extracted from the FDI module. This output is the statistical confidence needed to declare the fault via hypothesis testing (H_0: The motor is healthy vs H_1: The motor is faulty). The latter output needs another PDF to be considered as the baseline. The statistical parameters of the baseline PDF are derived from known healthy data, typically collected from the beginning of a component's lifecycle when it is known that no fault exists or any fault is negligible. In this case, a normal distribution $N(0, \sigma)$ is used to define this baseline data. This indicator is essentially equivalent to an estimate of type II error, or equivalently the probability of detection. The statistical confidence can be seen in Figure 11. Customer specifications are translated into acceptable margins for the type I and II errors in the detection routine. If additional information is required, it is possible to compute the value of the Fisher Discriminant Ratio. The baseline PDF of the faulty resistance and the estimated one at times $t = 107sec$ and $t = 200sec$ can be seen on Figures 12 and 13, respectively.

m	$21\,kg$	$\mathcal{I}$	$0.0948\,kg \cdot m^2$
d	$0.3\,m$	R_m	$1.779\,\Omega$
K_Ω	$0.1066\,V/RPM$	b	$0.000114\,N \cdot m \cdot sec$
K_t	$0.1066\,V/RPM$	K_f	$5.1991 \cdot 10^{-4}\,N \cdot sec$

Table 1. Simulator parameters

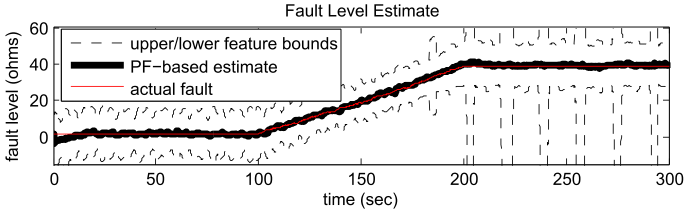


Figure 10. This figure illustrates the actual fault value and the estimated value from the PF algorithm.

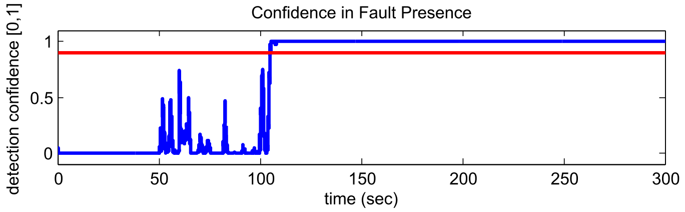


Figure 11. This figure illustrates the confidence metric.

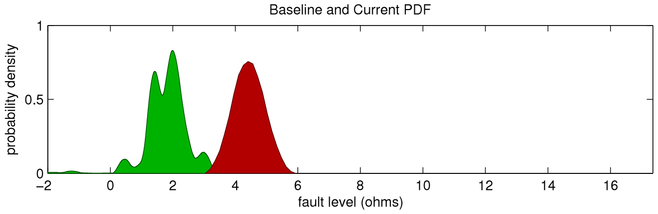


Figure 12. This figure illustrates the baseline (green) and estimated PDF (red) of the faulty resistance at $t = 107 sec$.

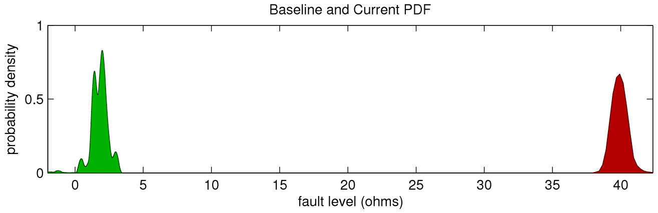


Figure 13. This figure illustrates the baseline (green) and estimated PDF (red) of the faulty resistance at $t = 200 sec$.

7. Conclusions

This paper is introducing a method for the development, implementation, testing and assessment of a particle-filtering-based framework for FDI . The proposed algorithm for FDI has been successful and very efficient in pinpointing abnormal conditions in very complex and nonlinear processes, such as the detection of a faulty motor to an autonomous hovercraft. A series of features has been derived using first principles modeling of the vehicles and thruster dynamics. The FDI algorithm can be executed on-line and is decoupled from the autonomous vehicle's control system. This allows for early fault detection minimizing the "masking" that can be potentially caused by the control system. In addition, the algorithm is capable of fusing information obtained from different subsystems of the vehicle.

Acknowledgment

This work was sponsored by NASA's Ames Research Center and Impact Technologies. We gratefully acknowledge their support and contributions to the research effort. We would also like to recognize the contributions of Edward Balaban and Brian Bole for their valuable input and feedback in reviewing this paper.

References

http://aviationknowledge.wikidot.com/aviation:ergonomics-of-cae-7000-series-simulator-configured. (n.d.).

Orchard, M. (2007). *A particle filtering-based framework for on-line fault diagnosis and failure prognosis.* Unpublished doctoral dissertation, Georgia Institute of Technology.

Orchard, M., & Vachtsevanos, G. (2007). A particle filtering-based framework for real-time fault diagnosis and failure prognosis in a turbine engine. In *Control automation, 2007. med '07. mediterranean conference on* (p. 1 -6). doi: 10.1109/MED.2007.4433871

Orchard, M., & Vachtsevanos, G. (2009). A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurment and Control*, *31*, 221-246.

Sallee, G. P., & Gibbons, D. M. (1998). *Aia/aecma project report on propulsion system malfunction plus inappropriate crew response (psm+icr)* (Vol. I&II). Aerospace Industries Association.

Statement, F. P. (2022, March). Faa policy on type certification assessment of thrust management systems. *FAA Policy Statement Number ANM-01-02*.

Zhang, B., Sconyers, C., Byington, C., Patrick, R., Orchard, M. E., & Vachtsevanos, G. (2011, May). A probabilistic fault detection approach: Application to bearing fault detection. In *Ieee transactions on industrial electronics* (Vol. 58).

Maintenance Planning Optimization Based on PHM Information and Spare Parts Availability

Leonardo Ramos Rodrigues[1] and Takashi Yoneyama[2]

[1] *EMBRAER S.A., São José dos Campos, São Paulo, 12227-901, Brazil*
leonardo.ramos@embraer.com.br

[2] *ITA – Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo, 12228-900, Brazil*
takashi@ita.br

ABSTRACT

Maintenance planning plays an important role in assets management because it directly affects assets availability. In the aviation industry, maintenance planning becomes even more important due to the high availability expectations from aircraft operators and the high costs incurred when an aircraft becomes out of service. Gathering and combining all the relevant information to generate an optimized maintenance planning is not a simple task because the number of variables to be considered is high. The aim of this paper is to present a new model to plan maintenance interventions, using RUL (Remaining Useful Life) estimations obtained from a PHM (Prognostics and Health Monitoring) system. This information is used to verify whether spare parts will be available when the next failures are expected to occur. Since spare parts are finite resources, the goal of the proposed model is to reduce the probability that multiple similar components will fail in a short period of time because, when it happens, there is not enough time to repair all failed components and fleet availability is penalized. To avoid this situation, the model suggests the anticipation of some replacements. This paper presents a simulation comparing a situation in which PHM information is not available with the proposed model in terms of fleet availability and investment in spare parts. Life cycle cost considering a time horizon of 15 years was also computed in simulations. The results showed that the proposed model allowed an increase in fleet availability and a reduction in the lifecycle cost.

1. INTRODUCTION

In a previous work, the authors presented an algorithm that uses PHM information for non-repairable items spare parts inventory control (Rodrigues & Yoneyama, 2012). In this paper, repairable items are addressed. Mathematical models for optimizing the performance of repairable components based on maintenance interventions have been widely discussed in the literature. Dekker (1996) presented an overview of many maintenance models for repairable items.

Planning maintenance interventions can be a complex task because there are many variables involved. An efficient maintenance plan must take into account information obtained from different sources. Gathering and combining all this information to generate an optimized maintenance planning is a challenge faced by maintenance planners.

This work presents a maintenance planning algorithm to support maintenance planning optimization. The proposed algorithm combines PHM information and spare parts availability estimations in order to schedule maintenance intervention with minimum impact on fleet availability.

2. SPARE PARTS INVENTORY SYSTEM FOR REPAIRABLE ITEMS

Repairable items are components or assets that, after a failure, are submitted to a repair cycle to be used again instead of been discarded (Fritzsche & Lasch, 2012; Lee, Chew, Teng & Chen, 2008). It implies that a repairable item spare part inventory system must have a repair shop where failed components are repaired, as well as a warehouse where spare parts are stocked (Perlman & Levner, 2010). An example of a typical spare parts inventory system for repairable item is shown in Figure 1.

Rodrigues, L. R. et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

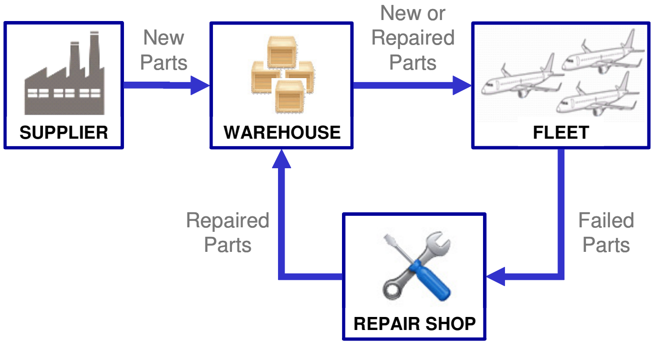


Figure 1. Spare parts inventory system for repairable items

In this inventory system, it is considered that spare parts are always bought from the same supplier and delivered at a single warehouse. When a component installed on an aircraft fails, it is removed and sent to the repair shop to be repaired. The faulty component is replaced by a new one from the warehouse. If there is no spare part in the warehouse, we assume that the aircraft is grounded until a new part is provided.

Once a faulty component arrives in the repair shop, it is submitted to the repair process. If limitation on repair shop capacity is considered, then a priority policy must be established. When the repair process ends, the repaired component is sent to the warehouse and stays there until a new failure occurs in the field. The repair process can be considered to be perfect (if repaired components returns to an “as good as new” condition) or imperfect (if repaired components keep a residual degradation). Imperfect repair models were presented by Do Van, Voisin, Levrat & Iung (2012) and Doyen & Gaudoin (2004). In this work, we consider that the repair shop has infinite capacity and that the repair process is perfect. We also consider that no degradation occurs to spare parts while they are in the warehouse.

2.1. Investment in Spare Parts versus Fleet Availability

One important decision to be made by inventory managers is related to the number of spare parts that will be bought in order to support fleet operation. In most real applications, the inventory system comprises multiple items, and the number of spare parts of each component must be defined.

The determination of how many spare parts of each component shall be bought must consider two conflicting variables: investment in spare parts and fleet availability.

Sherbrooke (2004) described a methodology called marginal analysis that can be used in order to determine the optimum sequence of spare parts to be bought in order to maximize the expected fleet availability.

3. Maintenance Planning

As a general rule, all assets demand maintenance interventions during their operational life. Maintenance planning plays an important role in assets management because it helps maintenance planners to schedule maintenance interventions with minimum impact in operation.

3.1. PHM Information

In order to identify the best moment to perform maintenance tasks, monitoring the health condition of assets can provide valuable information about how long an asset can operate before a failure occurs (Sandborn & Wilkinson, 2007).

PHM (Prognostics and Health Monitoring) is the ability of assessing the health state, predicting impending failures and forecasting the expected RUL (Remaining Useful Life) of a component or system based on a set of measurements collected from the aircraft systems (Vachtsevanos, Lewis, Roemer, Hess & Wu, 2006).

Based on measurements collected from the aircraft, a PHM system estimates the degradation level of monitored components. The degradation index is zero when the monitored component is new. During operation, degradation process starts and the degradation index increases. If the degradation index threshold that defines the failure is known, it is possible to extrapolate the curve generated by the evolution of the degradation index over time and estimate a time interval in which the failure is likely to occur (Leão, Yoneyama, Rocha & Fitzgibbon, 2008). This estimation is usually represented as a probability density function, as shown in Figure 2.

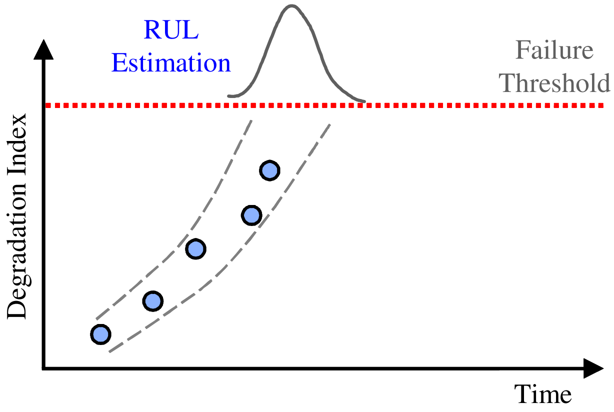


Figure 2. RUL estimation

3.2. Spare Parts Availability

In order to plan maintenance interventions, maintenance planners must verify the availability of all required resources such as technicians, spare parts, tools, etc. In this work, PHM information will be used to estimate the availability of spare parts. We consider that all maintenance interventions require a spare part. We also consider that all other resources are always available.

Spare parts in the repair shop are unavailable and can not be installed in an aircraft. They become available when the repair process ends and they are sent to the warehouse. Fleet availability is affected when a failure occurs and there are no spare parts in the warehouse.

Suppose S_X is the number of spare parts of component *X* and $R_X(t)$ is the number of components *X* in the repair shop at instant *t*. The number of aircraft grounded waiting for a component *X* at instant *t*, $G_X(t)$, can be calculated as a function of $R_X(t)$ and S_X as follows (Sherbrooke, 2004).

$$G_X(t) = \begin{cases} 0 & ;R_X(t) \le S_X \\ R_X(t) - S_X & ;R_X(t) > S_X \end{cases} \quad (1)$$

In Eq. (1), we can observe that fleet availability is affected by component *X* only when there are more than S_X components simultaneously in the repair shop.

4. Proposed Model

In the proposed model, PHM information is used to estimate when failures are likely to occur. Using the RUL estimations for the monitored components and their MTTR (Mean Time to Repair), it is possible to build an expected repair shop time schedule for each component type, as illustrated in Figure 3.

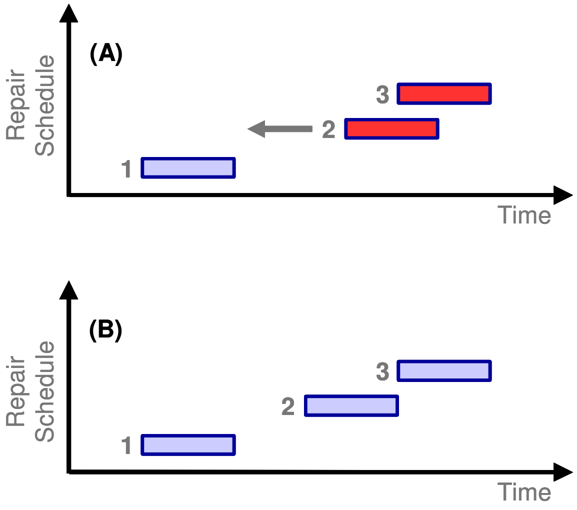


Figure 3. Repair shop time schedule estimation

Figure 3(A) shows an example of a repair shop time schedule for component *X*. Each bar in Figure 3 represents the repair cycle of one component. Let’s assume that the number of spare parts for component *X*, S_X, is 1. PHM information is used to determine when a failure is expected to occur and, consequently, when a faulty component is expected to be sent to the repair shop. MTTR is used to determine how long the faulty components will stay in the repair shop.

We can observe in Figure 3(A) that the third component is expected to arrive in the repair shop while the second component is still being repaired. In this situation, there will be two components simultaneously in the repair shop. During this time, $R_X(t)$ is 2, and according to Eq. (1), $G_X(t)$ is 1. In other words, there will be one aircraft grounded waiting for a component *X*.

In order to reduce the probability that multiple similar components will be simultaneously in the repair shop, some components can be replaced earlier. When some replacements are anticipated, the period of time in which aircraft are grounded can be reduced or even eliminated. In the example illustrated in Figure 3(A), if the replacement of component 2 is anticipated, we generate a new time schedule in which the maximum number of components in the repair shop never exceeds 1. This new time scheduled is shown in Figure 3(B).

The identification of concentrations of failure events, as well as the preventive anticipation of maintenance interventions, is possible only when PHM information is available. In a situation without PHM, the effects of the concentrations of failure events can not be reduced.

5. Numerical Example

In order to analyze the potential increase in fleet availability provided by the anticipation of some replacements to avoid the concentration of similar components in the repair shop at the same time, a set of simulations were run. The spare parts inventory system shown in Figure 1 was used in this example.

Two identical fleets were simulated. In the simulation of the first fleet, PHM information was not used. In the simulation of the second fleet, PHM information and spare parts availability estimations were used to anticipate maintenance tasks whenever a high concentration of similar spare parts in the repair shop was detected.

Four LRUs (Line Replaced Units) were considered in the simulation. Table 1 shows the price and the reliability data for each LRU. It is considered that an aircraft is available only if all its components are working properly. In other words, a failure of any LRU puts the aircraft to an AOG (Aircraft on Ground) condition.

Table 1. LRU data

LRU	A	B	C	D
Price [Monetary Units]	400	250	150	100
MTTF [days]	300	150	200	120
MTTR [days]	30	20	25	25

The decision of anticipating a maintenance task or not is made based on a cost criteria. The cost parameters used in the simulation are shown in Table 2.

Table 2. Cost data

Parameter	Value
Holding Cost	30% of component price per year
Repair Cost	30% of component price per repair
Stockout Cost	3.3 M.U. per day per aircraft

The PHM system estimates the RUL for all components installed in the fleet. The estimated RUL for each component is given as a normal distribution. In other words, for each component the PHM system informs the estimated RUL and a standard deviation. Table 3 shows the maximum and the minimum values for the error in the RUL estimation and for the standard deviation used in the simulation.

Table 3. PHM system data

Parameter	Value
Minimum RUL Error [days]	0
Maximum RUL Error [days]	20
Minimum RUL Standard Deviation [days]	5
Maximum RUL Standard Deviation [days]	20

Figure 4 illustrates the relation between the date of failure and the RUL estimation provided by the PHM system.

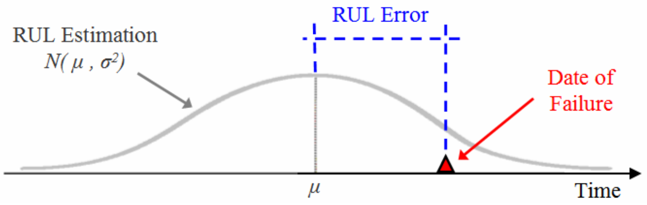


Figure 4. Failure date and RUL estimation

5.1. Scenario Description

The spare parts inventory system shown in Figure 1 is used in this example. The two identical fleets simulated will be compared in terms of investment in spare parts and expected fleet availability.

A spare part list must be defined in the beginning of each simulation. Once defined, we consider that all spare parts are bought from the supplier and are stored at the warehouse in the beginning of the simulation. When a failure occurs, a spare part is sent from the warehouse to replace the failed component, which is sent to the repair shop to be repaired. Once repaired, the component is sent to the warehouse and stays there until a new failure occurs in the field. In this work, we consider that components can always be repaired, and that repaired components are as good as new.

Sherbrooke (2004) developed a methodology to determine the optimum sequence of spare parts to be added to the spare parts list in order to maximize the expected fleet availability. We applied this methodology and defined the sequence of spare part to be bought. For each new spare part list, we repeated the simulation. Table 4 shows the optimum sequence of spare parts. PHM information is not necessary to calculate the optimum sequence of spare parts to be acquired.

The sequence of spare parts shown in Table 4 indicates that, for the group of LRUs considered in this example, if the inventory manager decided to support fleet operation with only one spare part, the best choice would be to have a spare part of LRU D. Another example: if inventory manager decides to invest 1,500 monetary units in spare parts, the optimum choice would be to buy the first eight spare parts listed in Table 4 (1 spare part of LRU A, 2 spare parts of LRU B, 2 spare parts of LRU C and 3 spare parts of LRU D).

Table 4. Optimum spare parts acquisition sequence

Spare Part	LRU	Cumulative Investment	Spare Part	LRU	Cumulative Investment
1st	D	100	9th	D	1,600
2nd	D	200	10th	C	1,750
3rd	C	350	11th	A	2,150
4th	B	600	12th	B	2,400
5th	D	700	13th	D	2,500
6th	C	850	14th	C	2,650
7th	A	1,250	15th	A	3,050
8th	B	1,500	16th	B	3,300

5.2. Simulation Results

After defining the optimum sequence of spare parts to be bought, a set of simulations without using PHM information were run, considering a fleet of 10 aircraft. The time horizon for each simulation was 15 years.

First of all, fleet operation was simulated with no spare parts at all. In this simulation, every time a component failed, the aircraft stayed out of service until the repair was completed. After that, the spare part list was incremented, following the sequence presented in Table 4. For each spare part list, 20 repetitions of the simulation were run. Figure 5 shows the average fleet availability obtained with each spare part list, including the first set of simulations with no spare parts.

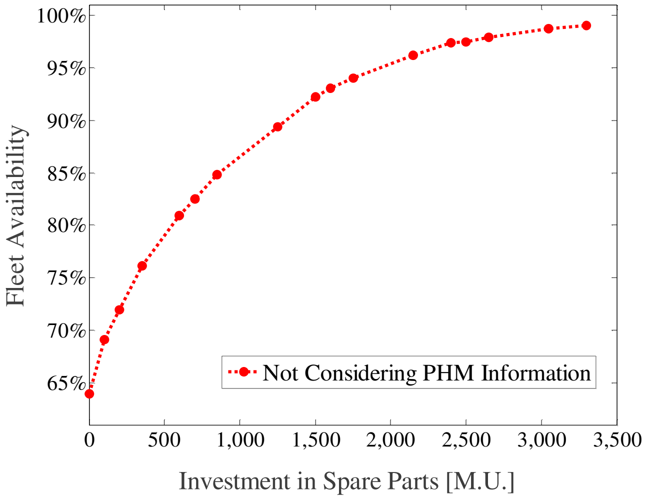


Figure 5. Average fleet availability versus investment in spare parts not considering PHM information

PHM information was then introduced in the simulation. The procedure of increasing the spare parts list according to Table 4 was repeated. Again, 20 repetitions of the simulation were run for each spare parts list. Figure 6 shows the average fleet availability obtained with each spare part list using PHM information and spare part availability (solid blue). For comparison purposes, the fleet availability curve obtained without PHM information – shown in Figure 5 – was also plotted in Figure 6 (dotted red).

Cost information presented in Table 2 was used in each simulation to calculate the expected maintenance life cycle cost. Since fleet availability and operational cost are conflicting variables, the purpose of this simulation was to investigate whether the increase in fleet availability obtained by the use of PHM information did not cause an increase in the maintenance life cycle cost.

Figure 7 shows the average life cycle cost computed during simulations. The investment in spare parts is shown in the horizontal axis, following the sequence presented in Table 4. For each spare part list, the bar on the left is the life cycle cost obtained without using PHM data, while the bar on the right is the life cycle cost obtained considering PHM information. In Figure 7, total life cycle cost is broken into four terms: investment in spare parts (black), holding cost (dark gray), repair cost (light gray) and stockout cost (white).

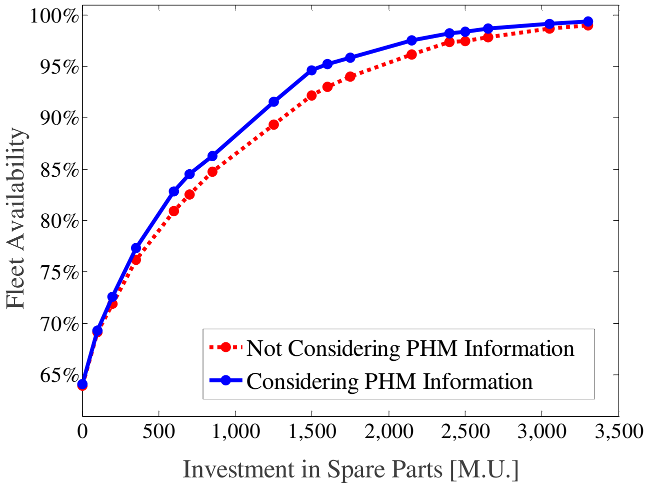


Figure 6. Increase in fleet availability when PHM information and spare parts availability is considered

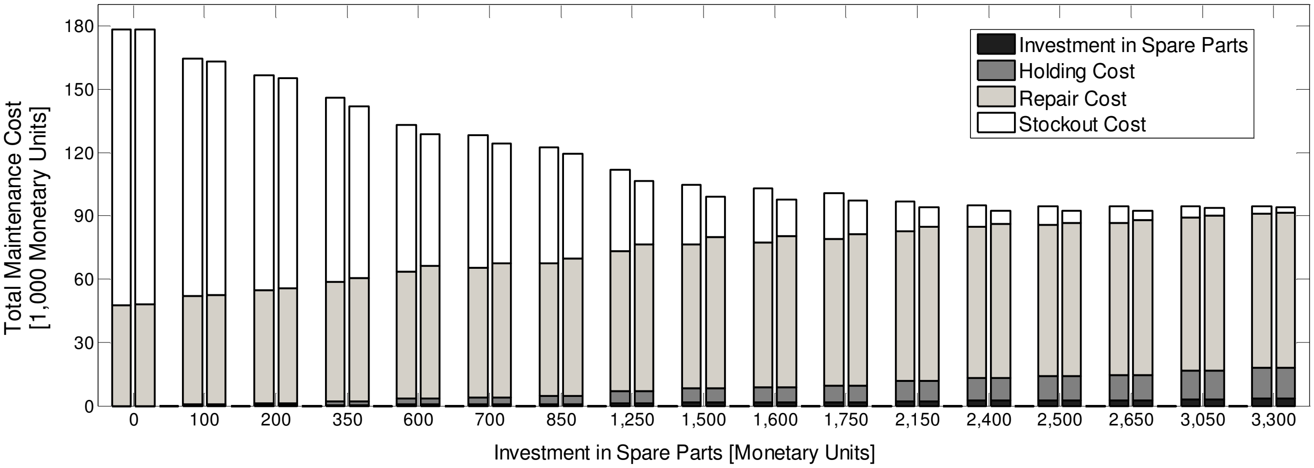


Figure 7. Average life cycle cost breakdown

6. Conclusions

We found that combining PHM data and spare parts availability estimations allowed us to improve fleet availability without additional investments in spare parts. RUL estimations provided by the PHM system were used in order to anticipate some maintenance actions. It avoided multiple similar components of being simultaneously in the repair shop and caused an increase in fleet availability.

The proposed model presented the best results when the expected fleet availability was around 92%. In this situation, the proposed model allowed and increase of 2.4 percentage points (from 92.2% to 94.6%). In all other situations, the proposed model allowed to achieve a better fleet availability in comparison with the situation in which PHM data are not used.

When maintenance tasks are anticipated, the number of maintenance interventions performed during fleet operational life is higher, and an increase in repair cost is expected. The computation of life cycle cost confirmed this expectation. However, the increase in fleet availability reduced the stockout cost, compensating the increase in repair cost.

Although the numerical increase in the availability achieved by using PHM information is small, the result is relevant considering that in the aviation industry the cost of an AOG event is usually very high. Intangible aspects associated to AOG events such as company reputation and damage to customer relationship are also relevant for aircraft operators.

Future research may extend the proposed model by considering the limitations associated with other resources such as technicians and tools.

Acknowledgment

The authors acknowledge the support of FAPESP (grant 2011/17610-0) and CNPq (research fellowships).

References

Dekker, R. (1996). Applications of Maintenance Optimization Models: A Review and Analysis. Reliability Engineering and System Safety, Volume 51.

Do Van, P., Voisin, A., Levrat, E. & Iung, B. (2012). Condition-Based Maintenance with both Perfect and Imperfect Maintenance Actions. In *Proceedings of Annual Conference of the Prognostics and Health Management Society*.

Doyen, L. & Gaudoin, O. (2004). Classes of Imperfect Repair Models Based on Reduction of Failure Intensity or Virtual Age. Reliability Engineering and System Safety, Volume 84.

Fritzsche, R. & Lasch, R. (2012). An Integrated Logistics Model of Spare Parts Maintenance Planning within the Aviation Industry. World Academy of Science, Engineering and Technology, Volume 68.

Leão, B. P., Yoneyama, T., Rocha, G. C. & Fitzgibbon, K. T. (2008). Prognostics Performance Metrics and their Relation to Requirements, Design, Verification and Cost-Benefit. In *Proceedings of International Conference on Prognostics and Health Management, Denver*.

Lee, L. H., Chew, E. P., Teng, S. & Chen, Y. (2008). Multi-Objective Simulation-Based Evolutionary Algorithm for an Aircraft Spare Parts Allocation Problem. European Journal of Operational Research, Volume 189.

Perlman, Y. & Levner, I. (2010). Modeling Multi-Echelon Multi-Supplier Repairable Inventory Systems with Backorders. Journal of Service Science and Management, Volume 3.

Rodrigues, L. R. & Yoneyama, T. (2012). Spare Parts Inventory Control for Non-Repairable Items Based on Prognostics and Health Monitoring Information. In *Proceedings of Annual Conference of the Prognostics and Health Management Society*.

Sandborn, P. A. & Wilkinson, C. (2007). A Maintenance Planning and Business Case Development Model for the Application of Prognostics and Health Management (PHM) to Electronic Systems. Microelectronics Reliability, Volume 47, Issue 12, Electronic system prognostics and health management.

Sherbrooke, C. C. (2004). Optimal Inventory Modeling of Systems: Multi-Echelon Techniques. In *2nd. ed. Springer*.

Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A., & Wu, B. (2006). Intelligent Fault Diagnosis and Prognosis for Engineering Systems. In *1st ed. Hoboken.*

BIOGRAPHIES

Leonardo Ramos Rodrigues holds a bachelor’s degree in Electrical Engineering from Universidade Federal do Espírito Santo (UFES, 2003), Brazil, and a Master Degree in Aeronautical Engineering from Instituto Tecnológico de Aeronáutica (ITA, 2008), Brazil. He is currently pursuing his doctorate in Aeronautical Engineering at ITA. He is with EMBRAER S.A. since 2006. His current research interests are the application of health monitoring techniques for electronic components and the usage of PHM information for inventory optimization.

Takashi Yoneyama is a Professor of Control Theory with the Electronic Engineering Department of ITA. He received the bachelor’s degree in electronic engineering from Instituto Tecnológico de Aeronáutica (ITA), Brazil, the M.D. degree in medicine from Universidade de Taubaté, Brazil, and the Ph.D. degree in electrical engineering from the University of London, U.K. (1983). He has more than 250 published papers, has written four books, and has supervised more than 50 theses. His research is concerned mainly with stochastic optimal control theory. Prof. Yoneyama served as the President of the Brazilian Automatics Society in the period 2004-2006.

Rotor Track and Balance Improvements

Eric Bechhoefer[1], Austin Fang[2] and Ephrahim Garcia[2]

[1]*Green Power Monitoring Systems, LLC, Vermont, 05753, USA*
ebechhoefer@gmail.com

[2]*Cornell University, Ithaca, New York, 14853, USA*
fredo0709@gmail.com
af55@cornell.edu

ABSTRACT

Vibration derived from the main rotor dynamics and imbalance causes premature wear to the aircraft components, and can cause pilot fatigue. While improvements have been made in rotor track and balance (RTB) techniques; there is room to enhance the quality of the recommended RTB adjustments.

One aspect that limits the development of RTB algorithms is the difficulty in quantifying the performance of new algorithms. This is because there are limited data sets to work on, and no agreed upon metrics on which to measure RTB performance.

This paper develops a methodology to simulate the vibration due to injecting a fault into the rotor system, and demonstrates metrics to evaluate the performance of a RTB algorithm. A new Bayes RTB method is evaluated against a standard least squares technique. In addition, a technique is presented to automate the selection of active adjustments.

1. INTRODUCTION TO ROTOR TRACK AND BALANCE

Vibrations in helicopters result in:

- Crew fatigue,
- Increased fatigue of mechanical parts,
- Higher probability of avionics malfunctions,
- And potential limits on the operational envelope (Rosen and Ben-Ari, 1997).

Failure rates for components in fixed-wing aircraft are lower than the rates for similar components installed in rotary-wing aircraft. The impact of vibrations on overall aircraft health was demonstrated in a study conducted by Sikorsky for the U.S. Army Air Mobility Research and Development Laboratory (Veca, 1973). In the study a squadron of H-3 helicopters were configured with rotor-mounted bifilar vibration absorbers and compared to a similar squadron that did not have the device. The two squadrons were similar in size and mission and flew approximately the same number of flight hours over the period of the study. The results were significant: overall helicopter failure rate and corrective maintenance requirements were reduced by 48% and 38.5%, respectively. Additionally, life-cycle cost showed a significant reduction of approximately 10% for the overall aircraft.

Vibration in helicopters is divided three general categories:

- High frequency vibration associated with the engine/gearbox and drivetrain. Typically, the frequencies are between 100 Hz to 10,000 Hz. Improvements to engine/transmission mounts and improved gear designs have greatly reduced this source of vibration.
- Medium frequency vibration, associated with the tail rotor, and to a lesser extent, high harmonics of the main rotor, are the main source of these vibration.
- Low frequency vibration, caused exclusively by the main rotor. This has the most severe effect on flight crew and equipment fatigue.

Main rotor vibration can be characterized as either vibration that is inherent due to the asymmetric nature of rotor dynamics in forward flight (present even with identical blades), and vibration due to the non-uniformity of the blades. The non-uniformity is due to the variation in manufacturing, and uneven wear/fatigue of the blade as a result of usage.

The vibration caused by non-uniformity results in ongoing maintenance and inspection by ground crews, and is the

Bechhoefer, E. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

focus of the development of improved Rotor Track and Balance algorithms.

2. Rotor Phenomenology

A rotor blade rotates with a constant angular rate Ω, with the root of the blade attached to the hub. The blade position for the k^{th} blade is: Ψ_k. The motion of the blade includes a flapping angle β_k, a lead-lag angle ζ_k, and a pitch angle θ_k, where k is the index of the blade. If elastic deformations are small, then β_k practically determines the blade tip path (Figure 1).

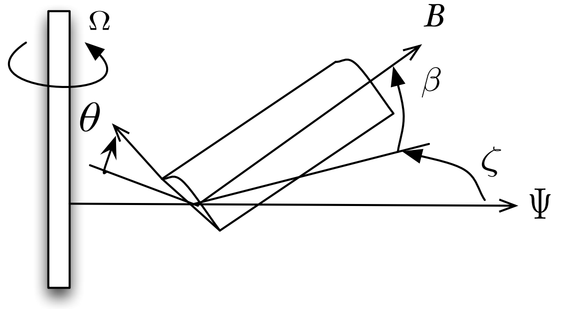


Figure 1 Blade Motion and Coordinates

The loads from the blades are transferred to the hub. If the blades are articulated, then moments acting on the hub are theoretically negligible. The force of the k^{th} blade on the hub is then:

$$F_k^H = X_\alpha sin(\psi_k) + X_\alpha cos(\psi_k) \qquad (1)$$

where X_α are the loads along the aircrafts x,y and z axis and the force due to blade k on the hub is F_k^H. In the case of identical blades, the sum of all forces on the hub would then be:

$$F^H = \sum_{k=0}^{b-1} F_k^H = 0 \qquad (2)$$

where b is the total number of blades in the rotor system. Deviation from a nominal blade will result in a non-zero force, which is measured as accelerations in the helicopter.

The relationship between perturbations between blades and the resulting track deviation and vibration is complex. Consider the case where the mass balance of all blades is identical, but the flapping angle, β_k, is different. By adjusting pitch of the k^{th} blade, an identical track/flapping angle could be reached, for a given helicopter airspeed. However, a change in pitch of that blade would:

- Affect that blade's lift and drag,
- Which would change the blade's lead/lag,
- That would in turn change the mass balance of the hub,
- Resulting in accelerations that increased vibration.

Initially, all efforts to decrease the non-uniformity of the blades started as an effort to reduce track split errors. But as many maintainers know, a flat track does not always result in a smooth helicopter. Since the primary goal of rotor track and balance is to reduce vibration, solving the problem efficiently is an underlying motivation.

2.1. Modeling Helicopter Vibration

The non-uniformity of the blades results in aerodynamic, mass imbalance, and track errors. To correct for these non-uniformities, rotor blades are manufactured with devices to purposely induce non-uniformity to cancel the effect of the naturally occurring blade errors. These devices include:

- Weights (WTS), which are attached at specific locations (hub and rotor tip) to change the blade moment,
- Pitch control rod (PCR) setting, which by changing length of the pitch rod, changes the angle of attack of that blade relative to the other blades, and
- Trailing edge tabs (TAB), which effectively change the blade's camber when bent. This in turn affects the aerodynamic loads/moments on the blade.

The acceleration due to blade induced vibration is measured at specific points in the aircraft, such as the:

- Pilot/Copilot vertical acceleration. These can be combined vectorially to derive cockpit vertical (A+B) or cockpit roll (A-B),
- Cabin Vertical
- Cabin Lateral

or other location where vibration deleteriously effects equipment or passengers. The levels of vibration will also be affected by the regime (airspeed) of the helicopter. For example, there is no flapping motion (β_k) when the helicopter is in ground or hover. Typical regimes for helicopter might be: Ground, Hover, 90 knots, 120 knots and 150 knots. The Fourier coefficients to describe the change in vibration then need to account for: adjustment type (a), sensor location (s), aircraft regime (r) and order (e.g. harmonic order, o).

For b blades, the Fourier representation of an adjustment, A_a, is the multiplication of the time domain representation of the adjustment (e.g. blade k) multiplied by the discrete Fourier transform matrix $\boldsymbol{D}_{k,o}$,

$$\boldsymbol{D}_{k,o} = exp(-i2\pi \times k \times o \,/\, b) \qquad (3)$$

And

$$\boldsymbol{A}_o = \boldsymbol{D}_{k,o} \times \boldsymbol{a} \qquad (3)$$

For example, the Fourier representation of an adjustment of 2 on blade 1, and 3 on blade 2, of a 3 bladed rotor is:

$$\begin{bmatrix} 2+1.73i \\ 2-1.73i \\ 5 \end{bmatrix} = \begin{bmatrix} -.5-.86i & -.5+.86i & 1 \\ -.5+.86i & -.5-.86i & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix} \quad (5)$$

The vector $\boldsymbol{A}_o$, is indexed by order: the 1st index is first harmonic (e.g. shaft order 1), the 2nd index is the 2nd harmonic, while the 3rd index is DC (static value, which for WTS in the sum of all weights, while for PCR/TAB is the coning angle of the blades). However, Equation (1) is by a given order, for an adjustment type. Measured vibration is for a given order, over sensor and regime. This means that an adjustment vector, over adjustment type, is built by calculating the DFT adjustment for a given adjustment type, then building and adjustment vector for a given order.

Consider a WTS adjustment of [1 1 0], a PCR adjustment of [0 2 3], and a TAB adjustment of [-5 5 0]. Then the DFT adjustment for order 1 is:

$$\boldsymbol{A_1} = [-1-0i \quad 2+1.73i \quad 0+8.66i] \quad (6)$$

In other words, the first term of each $\boldsymbol{A}_o$ vector calculated by multiplying $\boldsymbol{D}_{k,o}$ with each adjustment type vector, is combined into a new vector $\boldsymbol{A}_1$.

The acceleration, for a given sensor location and regime for order 1, in the matrix representation of Equation (1), is expressed as:

$$\boldsymbol{F}_1 = \boldsymbol{X}_1 \times \boldsymbol{A}_1 \quad (7)$$

Note that this is a linear model. It is assumed that the perturbation induced by $\boldsymbol{A}_o$, is small relative to the nominal blade, such that the Taylor series of $\boldsymbol{X}_o$ is dominated by the first derivative (e.g. slope). The concept that adjustment coefficients are linear has, been presented by other researchers (Ferrer, 2001., Dimarco, 1990).

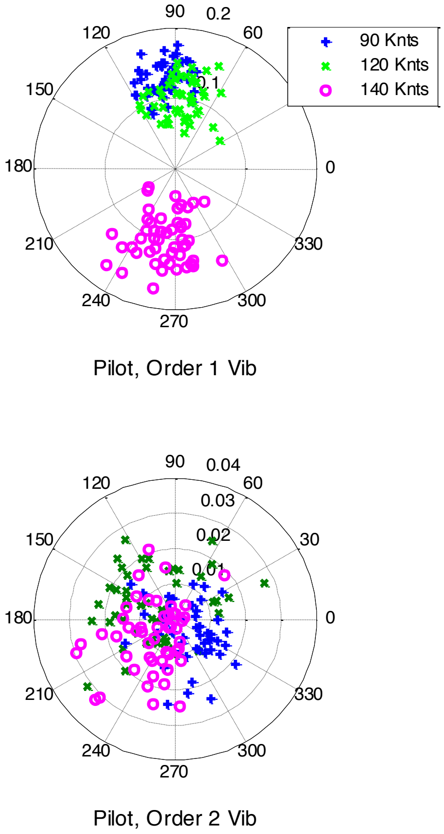
Equation (7) explicitly describes a system of equations that can generate vibration, for a given set of adjustment types, over a given order. This also suggest that:

- Implicitly, this means that there is no control over the *b*th order vibration (e.g. forth harmonic of a 4-blade rotor cannot be controlled passively).
- That vibration is operated on by order (e.g. one cannot solve a system of equations for order 1 and order 2 simultaneously. Meaning, when implementing a 2 blade solution on a 4 bladed rotor for vibration on the 1st order, if the vibration coefficients are not zeros for the 2nd order, the 2 blade solution affects the 2nd order vibration.

2.2. Development of a Vibration Simulation

Simulation provides a power tool to for RTB research. Because of restriction concerning software developed (FAA 2008), it is difficult to develop, test and mature algorithms, such as RTB, on aircraft. By modeling the vibration associated with the rotor, it is possible to test algorithms without the large expenses associated with on-aircraft development. This allows for quicker deployment of new features, and reduces risk of associated with deviating from an existing practice. Additionally, it allows testing that would be deemed to risky for on aircraft use.

Further, simulation will allow the development of metrics for algorithm performance evaluation. Consider a typical scenario to test a new RTB algorithm. Using Equation (3) and Equation (7), a known adjustment will derive a known vibration $\boldsymbol{F}$, plus measurement process noise. The test algorithm generates a solution, from which a residual error is derived (e.g. input adjustment – calculated adjustment, or difference in measurements prior to the adjustment and after the adjustment). This experiment can be run in Monte Carlo fashion to derive performance statistics. Hence, one can now develop probabilistic models on how well one RTB algorithms perform against another algorithm.



Pilot, Order 1 Vib

Pilot, Order 2 Vib

Figure 2. Simulated Pilot Sensor Vibration, 1st and 2nd Orders

Of course, simulation is only as good as the data that drives it. In this study, rotorcraft data from a 4-blade helicopter was used to model the vibration coefficients, $\boldsymbol{X_1}$, $\boldsymbol{X_2}$, with measured process noise. Process noise was modeled as a stochastic process, where a Gaussian random variable, $N(0, \sigma_{s,r,o})$ was added to the real and complex values of the vibration coefficients. These process noises where estimated from flight data. Figure 2 shows the simulated vibration for the pilot sensor at 90 knots, 120 knots and 140 knots, for orders 1, and 2. This is the estimated vibration as a result of injecting the these blade faults on a 4-blade rotor

- WTS: [0 5 10 0]. Because this is Hub WTS, there is no effect on the 2nd order, hence only 2 blades.
- PCR: [5 7 2 0]
- TAB: [5 0 3 -3]

3. Rotor Track and Balance Solution Strategies

RTB solutions present an unusually difficult challenge in solving. While optimization on Equation (7) using least squares or some other methodology, the solution is in the Fourier domain. In the conversion from Fourier to “time domain” or “real solution”, the result has multiple equivalent solutions. Consider an order 1 solution for a 4 blade rotor of: -8 -4i. There are four possible real solutions:

- [-8 -4 0 0], [0 -4 8 0], [-8 0 0 4] or [0 0 8 4]

These four solutions are equivalent in the Fourier domain. The best solution would be based externalities, such as: if an adjustment can be pulled off the blade, or an adjustment that minimizes track, or the preference of the maintainer.

3.1. Details on Converting the Adjustment from Fourier to Time Domain

For this discussion, the following convention for blade identification is used for a notional, 4 bladed rotor:

Black ⇔ $k = 0$, Yellow ⇔ $k = 1$, Blue ⇔ $k = 2$, Red ⇔ k=3

Expanding on the prior blade solution example, assume that the order 2 solution was: 2.0. From Equation (3), one should observe that the order 2 solution for a 4-blade solution is always real, and that the resulting time domain solution is: [2 -2 2 -2] . The implemented adjustments are the superposition of the order 1 and order 2 solutions:

Black	Yellow	Blue	Red
-8	-4	0	0
2	-2	2	-2
Adj: 6	**-6**	**2**	**-2**

However, no maintainer would implement this adjustment, as it is equivalent to an [8 -4 4 0] blade adjustment. Why touch 4 blades when 3 blades will do? In effect, the 4 blade solution captures DC in the Fourier domain, but in time, adds nothing to reducing the order 1 and order 2 vibrations, hence it should be removed. The DC component would, however, affect the helicopter rigging for such things as auto rotation, which is not a desired result of an RTB event.

3.2. A Procedure to Develop Real Blade Adjustments

Multiplying the inverse of Equation (3) to solve for the real blade adjustments results a solution with a DC component. In order to get a solution that would be implemented, an automated procedure must be used in order to provide appropriate adjustment solution. This will depend on the adjustment order and type.

A balance solution (either vibration or track) requires solving for Equation (7) for the number of blades–1 orders (recall that in the DFT, only the number of blades–1 order are available, as k^{th} blade is DC). Additionally, the solutions are conjugate (Ventres, 2000): the order 3 solution is the conjugate to the order 1 solution on a 4-blade rotor. For a 5-blade rotor, order 1 and 4 are conjugate, just as order 2 and 3 are conjugate. Thus, the RTB analysis calls the for the solution of Equation (7) for order 1 and order 2 (assuming a 4 blade rotor) then sets order 3 as the conjugate of the order 1 solution.

For WTS solution, since there is no flapping motion, there is no order 2 solution. The real blade solution is the set of all possible 2-blade solutions. This is found by multiplying the DFT solution of Equation (7) by the partitioned inverse of (3).

Set of possible 2-blade combinations:

- $\boldsymbol{B_1}$ = [1 2], $\boldsymbol{B_2}$ = [1 4], $\boldsymbol{B_3}$ = [2 3], and $\boldsymbol{B_4}$=[3 4]

Note that solutions such as [1 3] or [2 4] do not exist, as this is equivalent to adding weights on opposing blades. Since there is no order 2 solution, for each set (e.g. i = 1 through 4), the real blade solution would be:

$$\boldsymbol{a}[\boldsymbol{B_i}]_i = \boldsymbol{D}\left[\begin{matrix}1\\3\end{matrix}\quad \boldsymbol{B}_i^T\right]^{-1}\boldsymbol{A} \tag{8}$$

Recall that $\boldsymbol{A}[2] = \boldsymbol{A}[1]^*$ and that $\boldsymbol{a_i}$ is a real valued vector, where the index $\boldsymbol{B_i}$ is the blade adjustment value (say -8 on the black blade, and -4 on the yellow blade, for solution $\boldsymbol{B_1}$).

For a three-blade solution, which is appropriate for adjustments that are affected by blade flapping in forward flight, the set of all possible solutions is:

- $\boldsymbol{B_1}$ = [1 2 3], $\boldsymbol{B_2}$ = [1 2 4], $\boldsymbol{B_3}$ = [2 3 4], and $\boldsymbol{B_4}$ = [1 3 4]

The order 2 solution is real: for each set (e.g. i = 1 through 4), the real blade solution would be:

$$\boldsymbol{a}[\boldsymbol{B_i}]_i = \boldsymbol{D}\left[\begin{matrix}1\\2\\3\end{matrix}\quad \boldsymbol{B}_i^T\right]^{-1}\boldsymbol{A} \tag{9}$$

Here $\boldsymbol{A}[3] = \boldsymbol{A}[1]^*$ and that, again, $\boldsymbol{a_i}$ is a real valued vector. A comment on the values of $\boldsymbol{a}$ in both Equation (8), and

Equation (9) is that generally, adjustments for weights are an integer values. Similarly for PCR (number of "clicks" or "notches"), TABS are in mils of bend against a jib or dial caliper fixture. Thus, it is implied in Equation (8) and Equation (9) that the values are rounded to the nearest integer.

3.3. The Least Square Solution

The least squares solution (LSS) is a relatively simple solution strategy. The optimization object is to minimize the sum of squares residual error. In effect, this is the dual problem to the solution strategy implement by (Bechhoefer, 2011), in which the objective function was to minimize the adjustment size given a constraint on allowable vibration after the adjustment.

The LSS is a naïve implementation, in that is sensitive to outlier data, especially at the "end points". For non-Gaussian residuals (difference between the measured and predicted vibration), this could be problematic. That said, the solution to Equation (7) is simply implemented as:

For Each Order, *i*:

$$\boldsymbol{A}_i = (\boldsymbol{X}_i^T \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^T \boldsymbol{F}_i \qquad (10)$$

Then the set of real blade adjustments are calculated as per Section 3.2.

3.4. The Bayes Least Square Solution

One strategy to add robustness to Equation (10) is to weight the coefficients by some appropriate metric. One method would be to weight $\boldsymbol{F_i}$ by the Fisher's information matrix, which is a measure of the information carried in $\boldsymbol{F_i}$ (Fukunaga, 1990). This becomes Bayes least squares solution, where:

For Each Order, *i*:

$$\boldsymbol{A}_i = (\boldsymbol{X}_i^T \, \Sigma_i^{-1} \boldsymbol{X}_i)^{-1} \, \Sigma_i^{-1} \boldsymbol{X}_i^T \boldsymbol{F}_i \qquad (11)$$

And $\boldsymbol{\Sigma}$ is the measured covariance of $\boldsymbol{F_i}$.

3.5. Quantifying Solution Strategies

Given these solutions strategies, Equation (10) or Equation (11), one can now determine, stochastically, which algorithm will give the best performance given some objective. For this experiment, the norm residual vibration for order 1 and order 2 will be used. The scenario consists of 10 acquisitions for the 3 sensors, at 90 knots, 120 knots and 140 knots. The experiment will be run for 500 trials, and the PDF of the norm residuals well be evaluated. The norm residual is calculated by estimating the vibration given a proposed adjustment solution. That solution will use integer value adjustments from Equation (8) or Equation (9), where the estimated vibration of the solution is calculated using Equation (3) and Equation (7). The results are given in Figure 3.

Clearly, the Bayes solution strategy provides a more robust solution, as both the order 1 and order 2 norm residual vibration error is approximately 40% smaller.

3.6. Issues with Track

Typically, the object of RTB is to reduce vibration, and as noted, a flat track does not mean low vibration. However, there are cases where minimizing Track split is a requirement. For example, after a blade change and prior to flight, a flat track maintenance event is performed. This is primarily the result of established procedures but also serves the purpose of providing a better field of view for the pilots.

The solution strategy for track is identical to vibration. This is done by converting the track into its Fourier representation ($\boldsymbol{T}$) using Equation (3), replacing $\boldsymbol{a}$ in Equation (3) with blade track height, then replacing $\boldsymbol{F}$ in Equations (10) and Equation (11) with $\boldsymbol{T}$, and solving for the time domain adjustment per Section 3.2. Track is in fact a simpler solution. This is because for track, there is always only one sensor. Care must be taken in that, for one regime (ground), only one adjustment can be solved (typically a PCR adjustment).

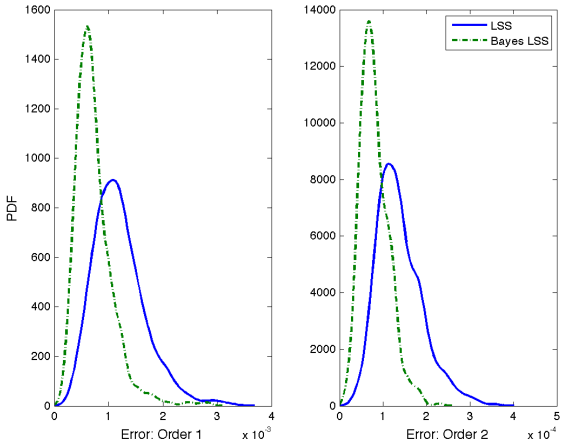


Figure 3. Comparison of LSS to Bayes LSS, for Order 1 and Order 2 Norm Error

4. Improving User Experience

In addition to reducing vibration, the RTB algorithm should present the maintainer with a solution that is easy to implement. Most commercial systems (Renzi, 2004) provide only a 2-blade solution (as they only solve for order 1 vibration). For track, a 2-blade solution can introduce some additional complexity in attaining a flat track in 1 adjustment (Keller, 2007). Additionally, order 2 or higher harmonics do occur and require maintenance adjustments to restore the helicopter into normal operational limits. Ideally, the RTB algorithm should be able to determine the most appropriate solution based on the measured vibration or track.

In (Bechhoefer, 2011), an expert system was developed in an attempt filter the options used by the RTB algorithm, based on the current set of measurements. The solution was not ideal in that it required an extensive library of *a priori* data. Essentially, configuration was needed to model to decision space, which selected the adjustment type (WTS, PCR, in board TAB, out board TAB), and adjustment order (1, or 1 and 2). The decision space encompassed 27 sets of configuration items.

An alternative method is proposed for the selection of adjustment type and adjustment order based on the estimated outcome of an adjustment. Because one can use Equation (7) to predict the vibration as a result of an adjustment, it is possible to estimate the residual vibration error post adjustment. This allows hypothesis testing for the adjustment/order options.

Consider that a full adjustment (WTS, PCR, TAB) is selected, and vibration order 1 and 2 are solved for (assuming a 4-bladed rotor). The residual error variance is then calculated. Say that an alternative adjustment is selected, in which only an order 1 (2-blade solution) is selected. Then one can test the hypothesis that the error variances are the same. If the test fails to reject the null hypothesis, then the simpler (2-blade solution) is selected over the 3-blade solution. Formally, as per (Wackerly, 1996), the test is derived as:

$$H_0: \sigma_1^2 = \sigma_2^2$$

$$H_a: \sigma_1^2 < \sigma_2^2$$

where the test statistic is:

$$F = {S_2^2}/{S_1^2}$$

The rejection region of the test is: F > Fα, where Fα is chosen so that P(F> Fα) = α when F have $v_1 = n_1$-1 degrees of freedom in the denominator, and $v_2 = n_2$ - 1 degrees of freedom in the numerator. This test is easily performed online, and requires only the selection of the probability of false alarm, α, which was set at 0.05.

Given the simulation capability developed in Section 2.0, and the vibration generated by the adjustments used in Figure 2, the probability distributions were calculated for order 1 norm error, order 2 norm error, and the track split. Multiple hypothesis test were conducted, where the null hypothesis was a full adjustment: [WTS/PCR/TAB], and the alternative hypothesis were reduced adjustment sets: [WTS/PCR], [WTS/TAB], [PCR/TAB] or WTS alone.

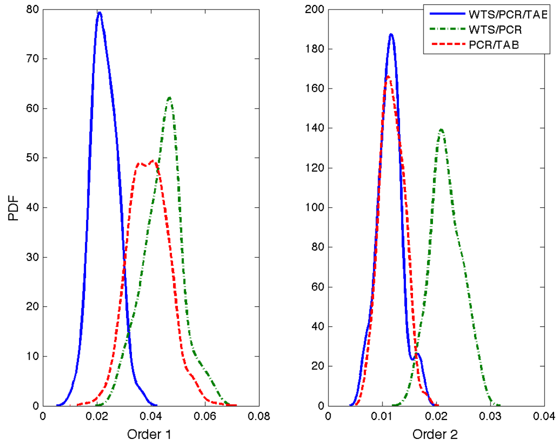


Figure 4. Order 1 and Order 2 Residual Vibration for Different Adjustments

The scenario assumed 10 acquisitions at 90 knots, 120 knots and 140 knots. Figure 4 shows that the algorithm selected between WTS/PCR/TAB (13% of the trials), WTS/PCR (75% of the trials) and PCR/TAB (12% of the trials). Figure 5 shows the difference in Track Split between the different adjustment sets.

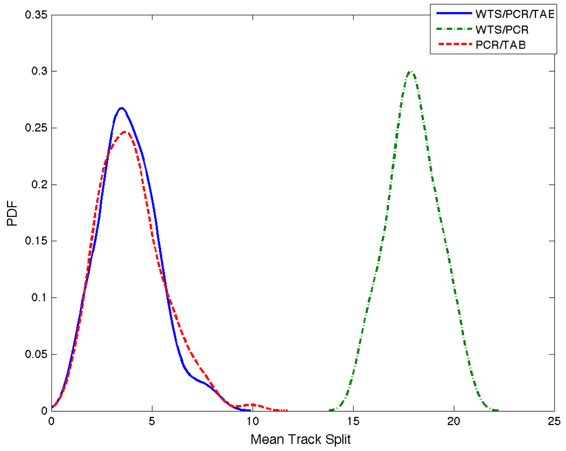


Figure 5. Track Split for Different Adjustment Sets

The algorithm did not select WTS alone, or a WTS/TAB solution, which as a general practice, reflects reality. We can note that the full adjustment results in a lower vibration in both vibration orders. The order 2 results are sensitive to the presence of a WTS solution. This is similar to the track performance issue – seeing as WTS has no effect on track, when the PCR or TAB adjustment is removed, the track split is larger. This is an important observation: improving Order 2 reduced the track split, even though optimization objective was vibration and not track. This suggests that a 2-blade solution (no reduction in Order 2) will always result in larger track split than, as seen in Figure 6.

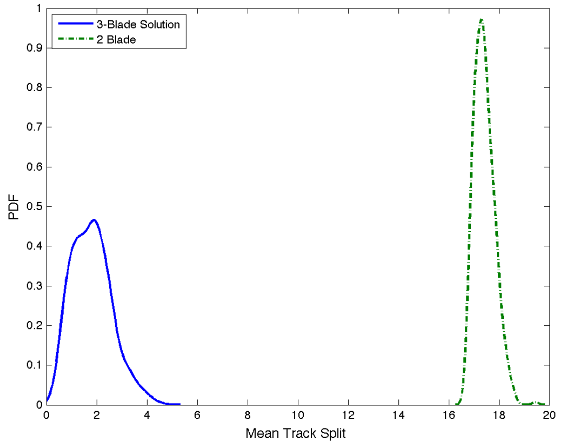


Figure 6. Effect of 2-Blade vs. 3-Blade Solution on Track Split

Given how the adjustments are selected, when the estimate of vibration is poor (due to stochastic nature of vibration), adding an adjustment, statically, does not improve the results. This hypothesis was tested by increasing the number of acquisitions per trial to 50 (Figure 7), as increasing the sample size improves the estimate by sqrt(*n*). This suggests that increasing the number of acquisitions in give time period will improve the overall quality of the adjustment and lower overall vibration.

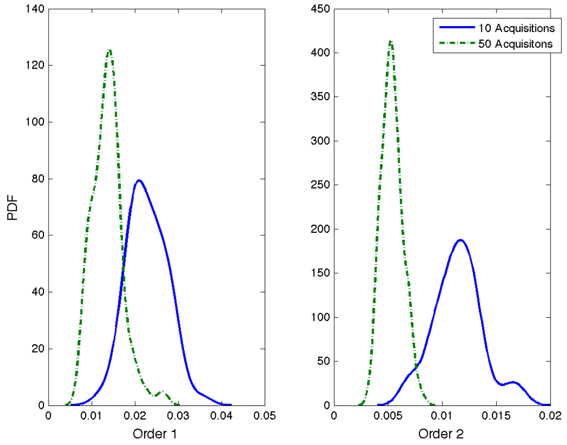


Figure 7. Order 1 and 2 Vibration for 10 vs. 50 Acquisitions per Trial

In the 50 acquisitions per trial case, because the estimate of the vibration was improved, the calculated adjustment results in a lower residual vibration error. Additionally, because the information was better, adding an adjustment improved the solution. This was seen in that the full adjustment set was selected 97% of the time, vs. 13% when only 10 acquisitions were used.

4.1. Methods to Reduce "Selection Fatigue"

Because each adjustment type has a large number of equivalent adjustments (see example in Section 3.0), even a WTS/PCR adjustment presented too many options for most maintainers. In some cases, it caused confusion and "selection fatigue". Additionally, both the helicopter manufacturer and the operator may have preferences as to what is a good adjustment. Subjectively, a good adjustment:

- Touches as few blades as possible
- Tries not to change the rigging of the helicopter
- Does not recommend adjustments which are too small to implement (e.g. minimum TAB is greater than 3 mils)

These preferences need to filter the adjustment such that the initial view to the maintainer is one set of WTS, PCR and TAB, which encompasses the rules or preference of the maintainer. A proposed rule set would be:

- Minimum DC offset on PCR. This ensures that, over time, the changes in PCR does not effect the helicopter rigging, and therefore the main rotor RPM during autorotation.
- Minimum TAB of +/- 3 mils. If mathematical solutions are less than 1.5 mils, zero the adjustment, if greater than +/- 1.5 and less than +/- 3 mils, round to 3 mils (sign appropriate)..
- Only add WTS. For a 4-blade rotor, since adding weights on one blade is the same as removing weights on the opposing blades, it's relatively easy for the maintainer to implement this.
- If there are two equivalent sets for an adjustment type, pick the adjustment set that intersects with a set of another adjustment type. This attempts to minimize the number of on which maintenance is performed.

Example: Generated Adjustments for WTS/PCR/TAB

WTS	Set 1	Set 2	Set 3	Set 4
Black	8	8	0	0
Yellow	-4	0	-4	0
Blue	0	0	-8	-8
Red	0	4	0	4

PCR	Set 1	Set 2	Set 3	Set 4
Black	-6	-2	2	0
Yellow	-8	-4	0	-2

Blue	-4	0	4	2
Red	0	4	8	6

TAB	Set 1	Set 2	Set 3	Set 4
Black	-13	-2	-8	0
Yellow	-5	6	0	8
Blue	-11	0	-6	2
Red	0	11	5	13

For PCR, the DC Offset is the sum of blade adjustments by set:

PCR	Set 1	Set 2	Set 3	Set 4
DC Offset	-18	-2	14	6

Set 2 for PCR affects the rigging the least, and touches the Black, Yellow and Red blades. For WTS, the positive adjustments are on the Black and Red blades. For TAB, corresponding adjustments are Black: -2, Yellow: 6, and Red: 11. Because the Black is -2, and violates the minimum adjustment for TAB rule, it is rounded to -3 with little effect on the vibration. Thus, the "best" adjustment presented to the maintainer is:

Adjustments	Black	Yellow	Blue	Red
WTS	8	0	0	4
PCR	-2	-4	0	4
TAB	-3	6	0	-4

5. CONCLUSION

In the paper, we present a methodology to simulate vibration on a helicopter for the purpose of developing, testing and, ultimately, improving Rotor Track and Balance (RTB) performance. Low frequency (e.g. order 1 and order 2, corresponding to the first and second harmonics of the main rotor) vibration is known increase the rate of component failure and to cause pilot fatigue. RTB maintenance is designed to reduce these vibrations.

Two potential solver strategies were presented, and using simulation procedure that was developed: the Bayes Least Squares solution was found to be superior to the Ordinary Least Squares in reducing vibration. Techniques were presented to automatically select the best adjustments based on the measured vibration. Additionally, the relationship between 2nd order vibration (e.g. the second harmonic of the main rotor) and blade track split was observed.

Most importantly, it was observed that increasing the number of acquisitions used in an adjustment reduced the post adjustment vibration. This could impact future RTB design requirements. Instead of sampling helicopter vibration once every 6 to 10 minutes (a limit imposed by the processing power of the onboard vibration monitoring system), the key to a rotor tuning may be sampling the helicopter once per minute. Simulation results showed that increasing the number of samples from 10 to 50 acquisitions reduced mean vibration error and track split by 45%.

NOMENCLATURE

Ω	angular rate of the main rotor shaft
β	blade flapping angle
ζ	blade lead-lag angle
θ	blade pitch angle
B	blade tip path
b	number of blades in the rotor system
F^H	force exerted on the rotor hub
$\boldsymbol{D}_{k,o}$	Fourier transform matrix
$\boldsymbol{a}$	time domain adjustment
$\boldsymbol{A}$	Fourier domain adjustment
$\boldsymbol{X}$	vibration adjustment coefficients
$\boldsymbol{F}$	measured vibration over regimes and sensors
DC	static load (sum of weights for WTS adjustment or conning angle for PCR/TAB adjustment).

REFERENCES

Veca, A., (1973). Vibration Effects on Helicopter Reliability and Maintainability. U.S. Army Air Mobility Research and Development Laboratory Technical Report 73-11.

Rosen, A., Ben-Ari, R., (1997). Mathematical Modeling of Helicopter Rotor Track and Balance: Theory, Journal of Sound and Vibration, vol. 200, No. 5.

Ferrer, E., Aubourg, P., Krysinksi, T., Bellizzi S. (2001). New Methods for Rotor Tracking and Balance Tuning and Defect Detection Applied to Eurocopter Products, American Helicopter Society International, Forum 57.

Dimarco, W., Floyd, M., Hayden, R., Ventres, C., (1990) "Method and Apparatus for Reducing Vibration over the Full Operating Range of Rotor and a Host Device" US Patent 4937758/

FAA, (2008), Certification of Transport Category Rotorcraft, AC-29-2C.

Bechhoefer, E., Fang, A., and Van Ness, D., (2011). Improved Rotor Track and Balance Performance, American Helicopter Society, Annual Forum 67.

Ventres, S., Hayden, R., (2000) "Rotor Tuning Using Vibration Data Only" American Helicoper Society Annual Forum, Virginia Beach.

Fukunaga, K., (1990). Introduction to Statistical Pattern Recognition. Academic Press, San Diego.

Renzi, M., (2004). An Assessment of Modern Methods for Rotor Track and Balance. Air Force Institute of Technology, Wright-Patterson Air Force Base.

Keller, J., Brower, G., (2007). Coefficient Development for Linear Rotor Smoothing on the MH-6 Main Rotor. http://www.rmc98.com/Coefficient%20Development%20for%20Linear%20Rotor%20Smoothing%20on%20the%20MH-6%20Main%20Rotor.pdf

Wackerly, D., Mendenhall, W., and Scheaffer, R., (1996) Mathematical Statistics with Applications, Duxbury Press, Belmont. Pg. 451-453.

BIOGRAPHIES

Eric Bechhoefer received his B.S. in Biology from the University of Michigan, his M.S. in Operations Research from the Naval Postgraduate School, and a Ph.D. in General Engineering from Kennedy Western University. His is a former Naval Aviator who has worked extensively on condition based maintenance, rotor track and balance, vibration analysis of rotating machinery and fault detection in electronic systems. Dr. Bechhoefer is a board member of the Prognostics Health Management Society, and a member of the IEEE Reliability Society.

Austin Fang is a senior engineer at Sikorsky Aircraft Corporation. After graduating Cornell University with an M.S. in Mechanical Engineering in 2005, Austin joined the Noise, Vibration, and Harshness group as a dynamics engineer. Austin holds a fixed wing private pilot certificate and enjoys working on condition monitoring applicable to rotorcraft.

Ephrahim Garcia is a Professor of Mechanical and Aerospace Engineering at Cornell University, College of Engineering. Dr. Garcia's is interested in dynamics and controls, especially sensors and actuators involving smart materials. Current projects include: Modeling and Analyses of Flapping Wings, Design and Control of Nanoscale Smart Material Actuators, Control of Reconfigurable Morphing Aircraft , Energy Harvesting for Biological Systems: Lab-on-a-Bird,Aeroelastic Energy Harvesting Modeling with Applications to Urban Terrain, Artificial Muscles for a Bipedal Walking Robot, and Mesoscale Hydraulics for Bio-inspired Robots

Wireless Power and Data System for Integrated System Health Management of Systems Operating in the Harsh Environment of Deep Space

Jim Miller[1], Jon Patterson[2], and. Ray Garbos[3]

[1,2]*NASA Marshall Space Flight Center, ISHM Engineering, Huntsville, AL 35812, USA*
jim.miller[1]@nasa.gov
jon.patterson[2]@nasa.gov

[3]*AMI Research and Development, LLC, 161 Brook Road, Mont Vernon, NH 03057, USA*
rgarbos[3]@ami-rd.com

ABSTRACT

Large and complex deep space platforms such as the Deep Space Habitat (DSH) being developed by NASA will require a robust, on-platform, Integrated System Health Management (ISHM) function. Currently the DSH is contemplated to be stationed at the L2 Lagrangian point outbound from the lunar orbit. This will provide a vantage point of the back side of the moon as well as to serve as a jumping off platform for manned trips to Mars, the Moon, or near Earth asteroids. The ISHM function includes the monitoring, diagnostics, prognostics, and failure mitigation strategies and capabilities for any viable failure modes of the DSH. To evaluate a prototype of this approach, NASA has assembled a full scale, ISS derived, DSH prototype at the Marshall Space Flight Center (MSFC), involving a wired ISHM sensor network of over 80 sensors located at various points where early system failure mechanisms may be detected and analyzed. However, it is anticipated that a wired, distributed architecture could involve many pounds of complex cable harnesses and connectors, along with the commonly encountered problems of accessibility, flexibility and maintainability. In the high likelihood that modifications or upgrades are needed, these complexities result in higher design and build cost along with increased operational costs as in-flight anomalies occur that could require the addition of different sensors or different sensor locations. To address these issues, the ISHM team at MSFC is studying a wireless, distributed architecture with on-platform, advanced prognostic and diagnostic capabilities. The approach being considered is based on the X-33 ISHM system which consisted of hardware identical remote health nodes (RHN) and a central vehicle health management computer. Each RHN was very flexible and reprogrammable to enable it to interface directly with all the health monitoring sensors. For application on the DSH, modifications to the RHN are being considered. These changes and resulting upgraded capabilities are described in this paper. As ISHM sensor technology gets smaller, more robust, and includes wireless interfaces for communication and power, the approach will be to connect these wireless sensors by adding state-of-the-art wireless technology to the X-33 developed RHN. This wireless approach eliminates connectors and cables, thus reducing development, installation and life cycle costs while improving reliability and flexibility of the ISHM systems.

1. DESCRIPTION OF ISHM ON THE DSH

As we move into deep space and establish long life systems for human occupancy, the attributes of ISHM systems become more valuable and border on being an enabling capability. In the case of the DSH prototype being considered for long life at the L2 Lagrangian point, parked in Cis-Lunar space, a large amount of sensing, diagnosing, and prognostication will be required. The state of health data and the algorithms that drive ISHM functions will be crucial to the survivability of the crew and the assurance of mission success at the lowest life cycle cost. These ISHM algorithms will involve structural health monitoring including the effects of micrometeoroid and orbital debris (MMOD) hits, monitoring of accumulated radiation dosage, and air quality monitoring along with other human protective systems such as the Environmental Conditioning and Life Support Systems (ECLSS). To feed these algorithms with system states of health (SOH) information will require numerous types of sensing, analyzing, and prognosticating elements. Because of the large physical size of the facility, these elements will be separated at significant distances in a network that must be extremely reliable as

Jim Miller et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

well as flexible and maintainable. This paper discusses the types of data needed to detect the onset of the several failure mechanisms, along with the diagnostic and prognostic algorithms that will assure the maintenance of a safe and functionally healthy system. The backbone of the ISHM system will be the sensors, along with several RHN boxes that enable end to end availability of all pertinent data needed to assess and control the overall state of health of the DSH facility. The specific design of the first DSH is still to be determined but will likely be based on heritage from the International Space Station (ISS) because much thought and investment has gone into that asset already and program re-use will be highly desirable. Figure 1 shows the concept that NASA is considering that uses the ISS laboratory and the multipurpose logistics module (MPLM) connected by a tunnel to serve as living quarters with all the essential elements for habitation, including a capability to grow green leafy plants for salad-type food. On the other hand, Figure 2 shows a futuristic structure that looks like the Star Trek science fiction version of a DSH. In either instance one thing is very clear; they both have a need for integrated system health management which will be critical for the long term support of human presence in deep space.

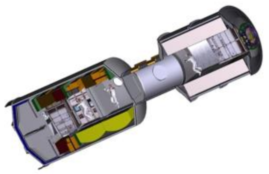


Figure 1. ISS Derived

Figure 2. Fiction Version

The possibility of abort, or mission abandonment, from the L2 Lagrangian orbit in Cis-Lunar space is not a trivial matter, making time to criticality (TTC) of utmost importance. Therefore, accurate monitoring of individual, as well as integrated systems states of health, along with diagnostics and prognostics related thereto is paramount to the viability of such an endeavor. Assuring that the life support environment is monitored and maintained constitutes only part of the equation. The ability to prognosticate and deal with future changes in that environment is equally necessary. The ISS-derived prototype being evaluated by NASA is populated with sensors dealing with determining states of health (SOH) and state of health trends that drive preemptive fault mitigation strategies and algorithms. This is necessary to assure the total monitoring and SOH understanding of the afore mentioned structural, communication, thermal, air quality, space radiation attributes, and the complete environmental conditioning and life support hardware and software over its full, functional, cradle to grave life. This not only drives the overall safety and reliability of the DSH but is the primary factor determining its life cycle cost.

To sense all the parameters that feed into the failure detection algorithms, the DSH prototype at MSFC presently consists of 84 various sensors that are highly distributed throughout the facility. These ISHM sensors are sampled by the data system on a two second sampling interval and data are analyzed according to the algorithm shown in Figure 3.

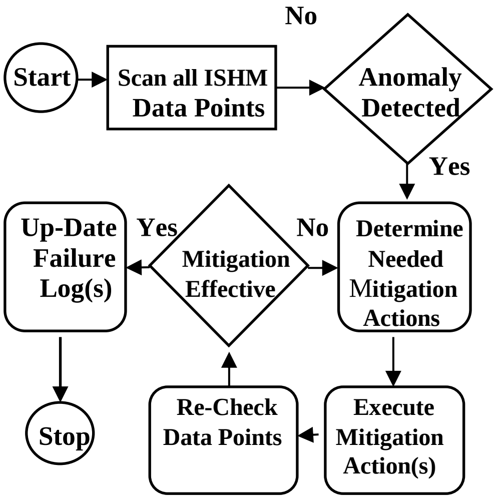


Figure 3. Failure Detection and Analysis Algorithm

The sub-tier algorithms noted in the diagnostic and prognostic boxes are specifically designed to accommodate the myriad of failure types and failure mechanisms that may arise. They are based on the failure modes and effects analyses (FMEAs) developed by the subsystem design engineers along with the fault management (FM) criteria established for the program. One basic assumption is that these algorithms are exhaustive and that all possible failures will be detected and mitigated in some fashion, including the possibility of mission abort and abandonment of the DSH if necessary. A data log of all triggers of the fault detection and analysis algorithm will be maintained. The major life critical subsystems of the DSH are shown below.

Critical elements of failure mechanisms are:

- Pressure vessel temperature and heat distribution
- Pressure vessel micrometeoroid and orbital debris hits
- Attitude determination and/or control
- Ability to communicate with Earth
- Air quality including O2 and/or hazardous gases
- Cabin temperature and rates of change
- Cabin air pressure and rates of change
- Aggregate of space radiation levels

An obvious engineering concern, and a major motive for the writing of this paper, is the consideration for the complexity of the sensor networks and associated cable harness. One very valuable lesson learned on previous programs, including the X-33, is that complex sensor cable networks are unreliable, heavy, intrusive, and expensive to modify. The application of a wireless system based on the X-33 developed RHN and the more recently upgraded REU should significantly improve the system reliability and cost by eliminating unnecessary sensor cable harnesses and reduce complexity. The result will be a very flexible design that lends itself to easy changes made necessary through growth in system complexity, parts obsolescence, or subsystem failures. Even after the DSH is placed in service at a deep space location such as the L2 Lagrangian point, the ISHM system based on a wireless sensor network will easily accommodate architectural changes as needed.

2. DESCRIPTION OF THE RHN

2.1. X-33 Generation 1 RHN Description

In the 1990s, NASA began development of a reusable launch vehicle (RLV) called the X-33 space plane. The integrated vehicle health management (IVHM) system for the X-33 consisted of a pair of host processors and 50 Generation 1 (Gen 1) Remote Health Nodes (RHN) distributed around the periphery of the X-33 to collect data from a variety of sensor types, (Garbos, Childers, & Jambor, 1997, and Garbos, & Mouyos, 1998). The RHN's interface to the health sensors and amplify, filter, and sample sensor signals before converting the data to digital format. Next, they analyze and perform some local prognostics decisions and store the data. The data is then packed with other information and tagged (e.g., ID and time sample) and transmitted to the vehicle health management (VHM) central processor.

The DSH will require a considerable amount of sensing, diagnosing, and on-platform prognostication to predict and ensure facility availability and safety. Because of the DSH size and possibility of on-facility maintenance and upgrading, a distributed ISHM architecture is proposed using wireless remote electronic units (REU's) derived from the RHN and shown in Figure 4 installed around the platform and interfacing to the state-of-health sensors. As discussed elsewhere in this paper, the wireless REU's were developed by NASA under the Extreme Temperature SiGe ETDP program, (Cressler, 2008, and Berger, Garbos, & Cressler, 2008, and Garbos, 2011) and are therefore much more robust and reliable for deep space use than the original RHN units, shown in Figure 5, developed for the X-33 program.

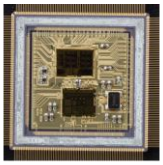


Figure 4. REU

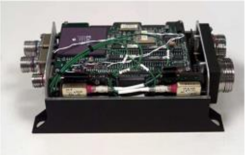


Figure 5. RHN

The Generation 1 RHN was an 11 kilogram box that dissipated 17 watts of power. The mixed-signal data acquisition unit was assembled from a combination of commercial integrated circuits and custom hybrids for the analog front-end arranged on three cards interconnected by ribbon cables. On one end of the box, a pair of large multi-pin connectors provided the sensor interface. On the other end, a pair of optical connections provided a redundant interface to the host computers. The RHN communicated with the hosts via a token-ring network topology, and power was provided through a +28 VDC bus.

The original RHN design sought modularity by combining multiple types of sensor interfaces into a single hardware implementation that was usable throughout a spacecraft or exploration vehicle to provide mission critical environmental and health data to engineers in an efficient, reliable manner. A key feature of the Gen 1 RHNs is the flexibility and re-programmability that enables one hardware design to interface with a wide variety of health sensors (e.g., temperature, strain, pressure, acceleration, vibration, acoustic, heat flux, position, rate and flow). Samples rates, gains, resolution, differential/single, full, half bridge inputs were all selectable for each sensor. In addition a programmable supply current was available.

On X-33, these sensors serviced all the vehicle health management requirements of the different vehicle subsystems (Propulsion, Thermal Protection, Cryo Tanks, Structures and Landing Systems, etc.). For the X-33 IVHM system 50 hardware identical Gen 1 RHNs interfaced to over 1200 health sensors were developed, tested, qualified and delivered to the program. The DSH wireless REU will incorporate the same basic flexibility as the previous designs but add wireless interfaces to eliminate cables and connectors where appropriate. This will lower the life cycle cost, increase reliability, and add flexibility for maintenance and possible upgrades over the life of the facility.

While the X-33 space plane program was terminated prior to flight, RHN box operation was demonstrated on an F/A-18 aircraft. Over 50 missions were flown during 1999 and 2000 with over 25 Gigabytes of information gathered which demonstrated the functionality and reliability of the box.

2.2. SiGe Gen 2 RHN Description

Whereas the X-33 Gen 1 RHN represented a significant improvement over the state-of the-art sensor networks of the day, there were obvious improvements that could be made in the miniaturization of the box. It was estimated that miniaturization would result in roughly two orders of magnitude improvement in volume, a 10x improvement in weight, and a 5x decrease in power dissipation. Extensive miniaturization as well as enhanced toughness led to the application of the more robust Gen 2 RHN (the REU) using SiGe technology which was funded by NASA under the

Extreme Temperature Development Program (ETDP) contract NNL06AA29C using mixed signal SiGe technology, (Cressler, 2008 and Berger, et. al., 2008, and Garbos, 2011). This development was led by a multi-center NASA ETDP team with support from multiple universities and demonstrated a Gen 2 SiGe System on a Chip (SOC) RHN based on the design of the X-33 Gen 1 Remote Health Node and implementing the same functionality. Two of the 16 channel REU's using mixed signal Application Specific Integrated Circuits (ASICs), are functionally equivalent to one X-33 thirty two channel unit.

The focus of this work was on a monolithic 16-channel system with integrated data conversion. It occupied an area of 10 x 14 mm and consumed 0.5 W + 0.25 W per universal/high-speed channel. These metrics represent an approximate 10x reduction in power consumption and a 100x reduction in form factor when compared to the Gen 1 RHN described above. Individual blocks that comprise the Gen 2 RHN were also flown in space as part of a Materials International Space Station Experiment (MISSE) Project in an effort to validate total-dose hardness and wide-temperature operability. The Gen 2 RHN was also radiation tested at cryogenic temperature under the ETDP program

2.3. Wireless Gen 2 RHN Description

As technology evolves, the power requirements for both sensors and health nodes will be greatly reduced. Also, a fully wireless sensor to Gen 2 RHN connection can be developed along with a fully wireless connection between RHNs and the central ISHM computer. The result will be that the need for connectors and cables for most of the distributed ISHM system will be eliminated and the entire system will be lighter and more reliable.

Depending on the type of smart sensor and required sample rate, there are different techniques for developing the wireless sensors. Some sensors that are sampled at a very low rate can self-generate and temporarily store data by scavenging energy from the local environment (vibration, temperature, etc.) then wirelessly transmit the data to a Gen 2 RHN. Also, these very low sampled sensors could be designed to respond to an RHN "ping" much like Radio Frequency Identification (RFID) tags operate today. These sensors could be easily interfaced to a wireless Gen 2 RHN.

For sensors that require a high sample rate and therefore more power than can be self-generated, the wireless Gen 2 RHN would be required to wirelessly transmit energy to the sensor. The sensor would then convert and store the energy within the device for response to the interrogation. Once the Gen 2 RHN is modified for a wireless interface and can provide energy to the ISHM sensors it can also be designed for wireless interface to the central ISHM computer. In some cases, however, the RHN may still require a wired power interface. However, this power would be a standard power that would be available around the platform and, therefore, not add much weight to the system. Since most of the technical solutions are already state-of-the-art in other applications, this would not be considered a high technical risk. Also, the Generation 2 RHN should be able to act as a relay station for other remotely located RHN's.

One big challenge will be to ensure that both ISHM wireless communication and wireless energy transmitting must not interfere with other subsystems, including other RHNs on the same platform. As the DSH ISHM work continues to define the sensor requirements, these issues can be addressed. Once these devices become available, the existing DSH prototype can be used to validate the approach. The benefits of eliminating most ISHM connectors and cables greatly exceed the risk.

Since the Gen 2 RHNs will be distributed throughout the platform they can also serve as the host for sensors, such as those for air quality and cabin temperature, since these will also be highly distributed. These sensors can be mounted directly to the RHN hardware.

3. CONCEPT OF OPERATONS FOR THE ISHM SYSTEM

Being at least three days away from a rescue and/or a re-supply from Earth, the crew of the deep space habitat will be totally reliant on the system's on-board health management system for mitigation of any credible failure scenario. In addition to such consumables as food, water, and oxygen, the crew must have spare parts, repair procedures, and materials on hand to deal with life threatening situations, such as those discussed earlier in this paper. Each failure scenario will be accompanied with a time to criticality, and the needed failure mitigation strategies must be tailored to fit within that TTC window. This means that at the time of failure detection the crew must be alerted as to the type, location, and TTC of the particular failure mechanism and the location and mitigation procedure(s) of needed actions to prevent the failure from causing life threatening consequences. Normally the time to criticality will be sufficiently long as to allow for a more leisurely repair. This would be true of a communication failure or cabin temperature rise. However, the TTC for a more severe hazard such as an MMOD puncture of the pressure vessel would be very short and would require rapid, emergency procedures.

In either case stated above, the crew would be notified initially by a caution and warning alarm consisting of both audible and on-screen annunciation(s). These would also appear in the personal hand held device(s) of each crew member. Once the caution and warning alert is received, the responsible crew member would use the touch screen techniques to drill down into the system diagrams and schematics to pinpoint the nature and location of the fault. One step further in the drill down procedure would bring the crew member to the mitigation procedure along with the location of spares, materials, and procedures needed to fix the problem. A very severe failure such as MMOD puncture may require emergency donning of space suits to survive the declining atmospheric pressure until the proper repairs are completed and the system is re-pressurized. A warning of an oncoming solar storm that will inflict life threatening radiation levels would require the crew to gather in a radiation shielded room, such as the exercise room. The radiation shield may be a heavy metallic enclosure or perhaps a water barrier surrounding the safe haven. In either event, the ISHM system will give the "all clear" signal when the threat has passed and/or dropped to a safe level.

The final action in all failure and/or life threatening situations that the ISHM system is responsible for is to update the event log and notify mission control back on Earth as to the nature of the event, the mitigation strategies employed, and the spares and materials used to fix the problem. The mission control personnel will then prepare the next supply vessel to carry replacements for all that was consumed during the failure event.

4. CONCLUSIONS

Travel and exploration in deep space within the next decade is likely and providing a habitat for long stopover and rendezvous for trips to Mars, the Moon, or Asteroids are the most likely missions. Whatever the state of technology of materials, structures, communications, or avionics and power, one requirement is conspicuous: the need for an integrated system health management (ISHM) capability. The sensors needed to provide total health monitoring will constitute the equivalent of the human nervous system and their sensed information must be gathered in a diagnostic and prognostic location where caution and warning alerts and mitigation actions are determined. Desirable attributes of such a system will be flexibility, maintainability, and reliability, all of which point to a wireless network free of electrical harness complexities and problems experienced in systems of the past. To provide the technology base for such systems, the remote health nodes described in this paper will prove to be the backbone of the ISHM system. Marginal improvements in the RHN circuitry to add wireless capabilities and improve robustness will assure the availability of the technology needed as the deep space habitat and other deep space systems are developed.

ACKNOWLEDGMENT

The Gen 2 RHN (SiGe REU) work was supported by NASA ETDP under contract NNL06AA29C. The program was managed by a NASA Team - A. Keys, M. Watson, D. Frazier, M. Beatty, D. Hope, and C. Moore. The researchers and students who contributed to the NASA SiGe ETDP program were led by a world class Georgia Tech Team of university and industry partners includes: Auburn University, Vanderbilt University, University of Tennessee, University of Maryland, JPL, BAE Systems, Aura Instrumentation, Inc. (working under BAE Systems), Boeing, Lynguent, and IBM.

REFERENCES

Berger, R. W. , Garbos, R. J. , Cressler, J. D. , et.al. (2008). A Miniaturized Data Acquisition System for Extreme Temperature Environments in Space. *2008 IEEE Aerospace Conference*, Montana, March 2008.

Cressler, J. D. (2008). Silicon-Germanium as an Enabling IC Technology for Extreme Environment Electronics. P*roceedings of the 2008 IEEE Aerospace Conference*, pp. 1-7 (on CD ROM), 2008.

Diestelhorst, R. M., England, T., Berger, R., Garbos, R., et. al. (2012). A New Approach to Designing Electronic Systems for Operation in Extreme Environments: Part I – the SiGe Remote Sensor Interface. *IEEE Aerospace and Electronic Systems Magazine*, vol. 27, no. 6, pp. 25-34, 2012.

England, T., Diestelhorst, R. M., Berger, R., Garbos, R., et. al. (2012). A New Approach to Designing Electronic Systems for Operation in Extreme Environments: Part II – the SiGe Remote Electronics Unit. *IEEE Aerospace and Electronic Systems Magazine*, vol. 27, no. 7, pp. 29-41, 2012.

Garbos, R. J., Childers, L., & Jambor, B. (1997) System Health Management/ Vehicle Health Management for Future Manned Space Systems. *AIAA DASC Conference, SHM/VHM*. Oct. 1997.

Garbos, R. J., & Mouyos, W. (1998). X-33/RLV System Health Management/ Vehicle Health Management. *Proceedings of the IAA/ASME/ ASCE/ASC Thirty-ninth Structures, Structural Dynamics, and Materials Conference and Exhibit*, Long Beach, California, April 20–23, 1998. AIAA-98-1928.

Garbos, R. J. (2011). Mixed Signal Silicon Germanium (SiGe): An Enabling Technology for Distributed Architectures in Extreme Environment Applications. *GOMAC Session 30-22*, Mar 2011.

BIOGRAPHIES

Jim Miller is a NASA Manager responsible for Integrated System Health Management (ISHM) in the Spacecraft and Vehicle Systems Department of the Marshall Space Flight Center. He holds a BSEE from Tennessee Technological University, an MSEE from the University of Alabama in Huntsville, and is a 1974 Sloan Fellow from Stanford University in Palo Alto, CA. He began his NASA career with the Langley Research Center in 1959 and later relocated to the Marshall Center following a three year tour of duty as a commissioned officer in the U. S. Army. During his time on active duty he taught radar and computer to NATO field grade officers in the Officer Training Division of the U. S. Army Ordnance Guided Missile School at Redstone Arsenal, AL. His entire career has been in the domain of avionic systems and advanced avionics technologies with special emphasis on ISHM and structural health management systems. He has served in team lead, Branch Chief, Division Manager, and Assistant Department Manager positions over the past several years. He has received several awards including the NASA Exceptional Service Medal, the Engineering Directorate's Technology Achievement Award, the Director's Commendation Award from the MSFC Center Director, Dr. Petrone, and the Award of Achievement from the MSFC Center Director, Dr. Werner Von Braun. Mr. Miller has authored and/or co-authored several technical papers in the domain of advanced avionics systems and ISHM. His most recent was presented at the International Workshop on Structural Health Monitoring (IWSHM) at Stanford University in September 2011. He is currently working on the ISHM system for the ISS Derived Deep Space Habitat.

Jon Patterson is a member of the Integrated Systems Health Management (ISHM) and Automation Branch, at the Marshall Space Flight Center. He holds Bachelor and Master of Science degrees in Computer Science from the College of Engineering at Louisiana Tech University in 1983 and 1984. In addition, he completed the coursework in the PhD Computer Science program in 1992 in the College of Engineering at the University of Alabama in Huntsville with an emphasis on Artificial Intelligence. Jon has worked in software development in industry with SCI, Inc. and General Electric prior to coming to NASA. He came to NASA, Marshall Space Flight Center in 1991 in the Avionics Department Simulation Division. Shortly after coming to NASA, he served as the chairperson for the MSFC Artificial Intelligence Working Group (AIWG). In 1995, he was selected as the Chief for the Design and Implementation Branch in the Software Division, where he provided technical software oversight for the Tethered Satellite System Re-Flight (TSS-R) Skip-rope Observer project and the Advanced X-Ray Astrophysics Facility (AXAF) Science Center (ASC). Jon has also served as a branch chief or team lead until he was selected to support NASA's X-33 program in Palmdale, California, in 1999 as NASA's X-33 Lead Software Engineer. Following his return to Huntsville, he continued to provide technical leadership in the areas of software and ISHM for numerous NASA programs. Throughout his 30 year career he has supported programs such as the Advanced X-Ray Astrophysics Facility for Spectroscopy (AXAF-S), the Atlas/Centaur Automated Diagnostic System (ACADS), the X-33 Health Management System, the Space Launch Initiative (SLI) Integrated Vehicle Health Management Task Area, and the Orbital Space Plane (OSP) program. Jon has most recently served as the technical lead for the Ares I Failure Detection, Notification, and Response (FDNR) system. He is currently serving as the Space Launch System (SLS) Vehicle Management (VM) lead for the development of the SLS Mission and Fault Management (M&FM) capabilities. He is deeply involved in the promotion and improvement of ISHM definition and development across the agency, including numerous collaborative efforts with other NASA centers.

Ray Garbos is an Engineering Fellow. He is responsible for the development of advanced Avionics/ISHM concepts, architectures and technologies for aerospace applications. He has over forty years of circuit and system architecture design experience. He was an Engineering Fellow for Sanders Associates (1984-85), Lockheed Martin (1986 -2000) and BAE Systems (2001-06). He was VP and Chief Engineer of Aura Instrumentation Inc. (1998-2010). He was the X-33 IVHM lead and the Reusable Launch Vehicle Avionic IPT lead for Lockheed Martin and has participated in many Advanced Space Avionics Studies supporting MSFC. He was the technical lead for the BAE Systems lead ARES I Avionics Proposal Team. Mr. Garbos was a charter member of the NASA initiated Strategic Avionics Technology Working Group (SATWG) circa 1988 that evolved into the Aerospace Technology Working Group (ATWG) where he participated until 2008. He received a BSEE/MSEM degrees from Northeastern University, Boston, MA, in 1968/1971 and a Math (MAT) degree from Rivier College, Nashua, NH, in 2001. He was an Adjunct Assistant Professor at the University of New Hampshire. He has been involved with Science Technology Engineering and Mathematics (STEM) outreach for over 20 years.

Processing for Improved Spectral Analysis

Eric Bechhoefer[1], Brandon Van Hecke[2] and David He [2]

[1]*Green Power Monitoring Systems, LLC, Vermont, 05753, USA*
eric@gpms-vt.com[1]

[2]*Dept of Mechanical and Industrial Engineering, The University of Illinois at Chicago, Illinois, 60607 USA*
bvanhe2@uic.edu[2]
davidhe@uic.edu[2]

ABSTRACT

The Fast Fourier Transform (FFT) is the workhorse of condition monitoring analysis. The FFTs' assumption of stationarity is often violated in rotating machinery. Even in a six second acquisition on a wind turbine, the shaft speed can change by 5%. For Shaft and Gear analysis, this is mitigated through the use of the time synchronous average. For general spectrum analysis, or bearing envelope analysis, there is no such mitigation: one hopes that the effect of variation in shaft speed is small. Presented is a time synchronous resampling algorithm which corrects for variation in shaft speed, preserving the assumption of stationarity. This allows for improved spectral analysis, such as used in bearing fault detection. This is demonstrated on a real world-bearing fault.

1. INTRODUCTION

It would be hard to imagine the condition monitoring (CM) of rotating equipment without the use of the Fast Fourier Transform (FFT). Everything from simple spectrums (such as Welch's method for power spectral density), to more advanced analysis (amplitude modulation and frequency modulation analysis (McFadden, 1985)) are dependent on the FFT.

In using the FFT, the CM practitioner must understand the base assumptions of continually differentiable (Gibbs Effect), and stationarity. In general, window functions (Hann, Hamming, etc.) are used to control or mitigate Gibbs effect, while the time synchronous average (TSA, McFadden 1987, Bechhoefer, 2009a), is used to mitigate the effect of non-stationarity in rotating machinery for shaft and gear analysis.

The issue of non-stationarity is not well addressed for the power spectral density (such as Welch's method), or for bearing analysis (envelope/heterodyne methods included). It is assumed that the smearing of energy due to changing shaft speed is small. For bearing analysis, the energy associated with a fault frequency is trended. A poor measure of that energy will result in variance in the trend, or just an inaccurate estimate of component damage.

Variance in shaft speed in rotating equipment is always present to varying degrees. The variation is due to:

- Limits in the control system bandwidth,
- Varying loads associated with the work the machine is producing,
- Or in the case of wind turbines, varying wind speed and torque ripple

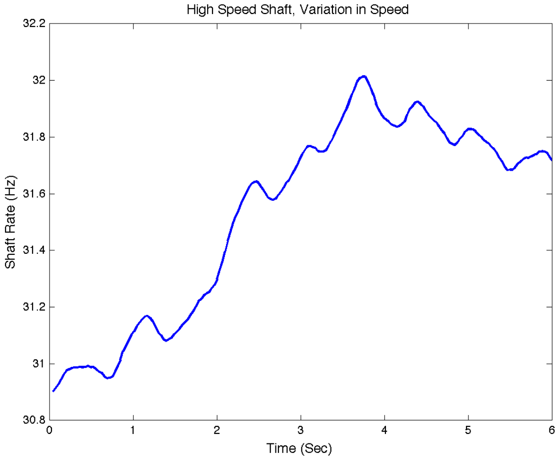


Figure 1. Variation in High Speed Shaft over 6 Second Acquisition

Wind turbines pose a particularly difficult environment. The wind, as noted, is time varying. Additionally, because the flow of wind is stalled in front of the tower, the lift on a blade as it passes in front the tower is reduced. This causes a 3/revolution torque ripple/change in shaft rate (Figure **1**).

Bechhoefer, E. el al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Finally, because of wind shear, the wind speed at the top of the rotor arc is greater than the bottom of the arc. Figure **1** shows the variance of a high-speed shaft on a wind turbine, over a 6 second acquisition. The instantaneous speed of the shaft is seen to range from 30.9 Hz to a maximum of 32.01 Hz, or a change in speed of 3.6%.

Consider the effect of this variation in speed on the spectral content of a bearing fault frequency, where the cage, ball, inner and outer race rates are: [0.42, 2.87, 9.46, 6.72]. The range of fault frequencies (Hz) during this acquisition are:

Table 1. Bearing Rates

Bearing\shaft	**Low: 30.9**	**High 32.01**
Cage (Hz)	12.98	13.44
Ball (Hz)	88.69	91.88
Inner Race (Hz)	292.3	302.8
Outer Race (Hz)	207.7	215.1

For the higher frequency bearing components, (inner/outer race), the frequency difference is significant: approximately 10 Hz. For spectral analysis, it poses a problem. Not only is there the issue of the spectral content smeared across a number of bins, but also which shaft rate does one use for analysis (the mean shaft rate over the acquisition)?

This issue of spectral spreading in the FFT is not academic. Consider the trend of an inner race fault on a high-speed shaft (Figure **2**). The variance in the condition indicator is proportional to the inner race energy. While vibration measurements are stochastic, not all of the variation in Figure **2** is due to measurement noise. We will show that some portion of the condition indicator (CI) is a function measurement error due to variance in shaft speed.

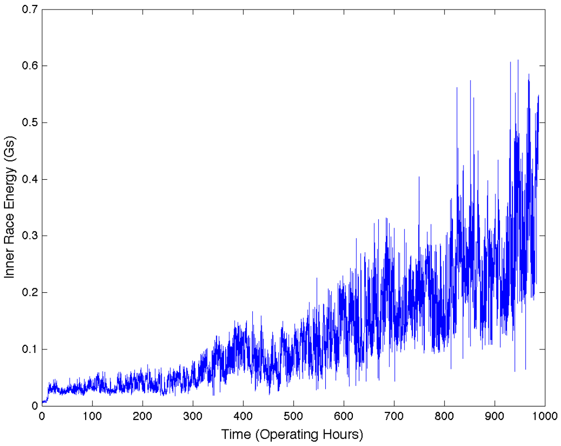


Figure 2. Trend of an inner race fault

This increased variance affects both the threshold setting process and alerting. Clearly if one does not have the luxury of sampling under steady state, a process is needed to mitigate the change in shaft RPM. We will show that resampling is one method that can be used to reduce the variance in the measured bearing energy.

1.1. Units of Measurement

In this paper, the units are in G's, where 1g is the earth standard gravitational acceleration. While the sensors output voltage, the accelerometer manufacture defines the scale value to convert for volts to G's. The ISO (ISO 10816) has developed standards for vibration limits for rotating industrial machinery, these standards are limited for equipment running between 10-200 Hz. Additionally, the limits are directed at imbalance, and not bearing faults. ISO vibration limits are inches/second, were the conversion from inches to G's is. G = 0.0162 * V * *f*. In general, 1 inch/sec is considered damaging vibration levels.

Many software packages output spectrum in power (G^2/Hz), but prior research (Bechhoefer 2008) revealed that the correlation between damage and energy (G's) was linear. For this reason, the units are in G's.

2. SYNCHRONOUS RESAMPLING

The model for vibration in a shaft in a gear box was given in (McFadden 1987) as:

$$x(t) = \Sigma_{i=1:K} X_i(1+ a_i(t))\cos(2\pi i\ fm(t)+ \Phi_i)+b(t) \quad (1)$$

where:

- X_i is the amplitude of the kth mesh harmonic
- fm(t) is the average mesh frequency
- $a_i(t)$ is the amplitude modulation function of the ith feature harmonic.
- $\phi_i(t)$ is the phase modulation function of the ith feature harmonic.
- Φ_i is the initial phase of harmonic k, and
- b(t) is additive background noise.

The mesh frequency is a function of the shaft rotational speed: *fm* = N*f*(t), where N is the number of teeth on the gear and *f(t)* is the shaft speed as a function of time. For bearings, *N* is the component rate, which is a non-integer value based on the bearing geometry. As noted, because of the finite bandwidth of the feedback control, or due to the environment, there is some wander in the shaft speed. This change in speed will result in smearing of amplitude energy in the frequency domain.

If a tachometer signal is present (such as a key phasor) and the ratio from the key phasor to the shaft under analysis, the vibration data can be resampled such that number of data points between one revolution and the next is the same. In the case of time synchronous averaging (TSA), the ensemble average of EQ(1) is calculated summing each revolution of resampled data, then dividing by the number of revolutions during the acquisition. .

Since the radix-2 FFT is most commonly used, the number of data points in one shaft revolution (r_n) are interpolated into m number of data points, such that:

- For all shaft revolutions n, m is larger than r, and
- $m = 2^{\text{ceiling (log2 }(r))}$ (again assuming Radix 2 DFT)

Since some other analysis process will be done on the resampled signal (envelop analysis, for example) – a radix-2 length is not necessary. However, for this example a Radix 2 length was used to calculate the resample length m. The algorithm resamples index r_i in m data points, the concatenates them into a new vector. Once all of the data is resampled, the envelope/spectrum is taken. Figure **3** compares the TSA algorithm to the Resampling algorithm.

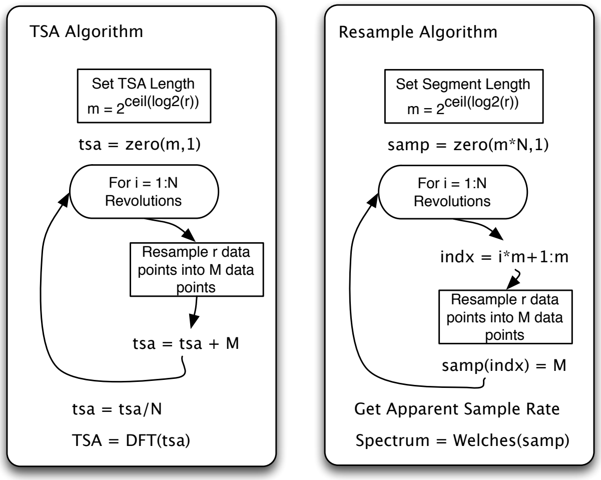


Figure 3 TSA and Resample Algorithm Flow

2.1. Example: Synchronous Resampling Algorithm

For example, say the sample rate was 1000 samples per second, and the lowest shaft rate was 10 Hz, for a .5 second acquisitions. The resample length, m, is 128. The number of data points between each key phasor is: 87, 92, 100, 95, 89, 37. For each shaft revolution, the data is resampled to length l.: Rev 1: 87->128, Rev 2: 92->128, Rev 3: 100->128, Rev 4: 95->128, Rev 5: 89->128. Note that for half of a second of data, there is 640 data points – the remaining 37 data were in the next, incomplete revolution, so the last 37 data points are dropped. The resample length is taken at the next largest power of 2 over the maximum length of all revolutions, again, assuming a radix 2 DFT.

Because of interpolation, the sample rate for each revolution is now changed. To accurately determine the frequency associated with a DFT bin, an apparent sample rate is needed. The apparent sample rate is the original sample rate * length of the resampled data / length of the original data: 1000 * 640/(463), or 1382.

2.2. TSA for Bearing Analysis

For shaft and gear analysis, existing TSA algorithms control for changes in shaft speed. For bearing, because they do not have integer number of shaft for a rate, the TSA is felt to be inappropriate for three reasons:

- Bearings are quasi-stationary – there is always some slippage such that even with correct geometry, the rates are not exact. This will make the bearing component non-synchronous with the TSA algorithm and in fact may separate the bearing signal out of the TSA.

- A bearing has rates for each component: cage, ball, inner and outer race. This would require the TSA to be run four separate times for each bearing, in order to capture the energy for each bearing component. While this may not be a problem for off line analysis, it may exceed the resources of an on-line analysis system. Considering that any given shaft is supported by 2 to 3 bearings, which would require 8 to 12 TSA analyses.

- In the evaluation of bearing health, it is important to be able to observe the relationship between the shaft, cage, ball, inner and outer race fault features. For example, an inner race fault that is modulated by shaft (e.g. side bands that are 1 shaft rate off of the inner race fault) is a more serious fault than an inner race fault, as it indicates wear and clearance issues on the shaft. The ability to view modulation between bearing components and shaft is a powerful diagnostics tool that is not available if using the TSA for each bearing components

That said, the structure of the TSA is the model for which this resample algorithm is based.

3. High Speed Shaft Bearing Fault

A commercial wind turbine with a 2 MW power output was installed with a condition monitoring system. Data was collected at 10-minute intervals. The data was sampled at 97656 sps for 6 seconds. Bearing envelop analysis was performed by band passing the signal between 9 to 11 KHz. Welches spectrum was used on the heterodyned signal with a DFT length of 4096, and with an overlap of 2048 points. Increased inner race energy on the high-speed shaft bearing indicated a fault (Figure **2**). An inspection of the bearing latter showed that the inner race was cracked. Using this data, the raw spectrum was compared to the resampled spectrum and the TSA (Figure **4**). The spectrum length was the same as the TSA length, so that the plots have similar bin widths.

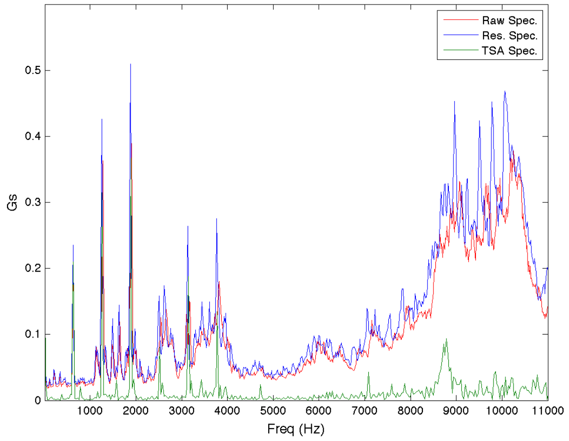


Figure 4. Raw, resampled, and TSA spectrum

The average shaft rate is 30.9 Hz. This shaft is driven by a 20-tooth pinion. The first 6 harmonics of the pinion are clearly visible at 620, 1240, 1860, 2480, 3100 and 3720 Hz. The TSA ran 188 revolutions, so that the noise floor of the TSA is approximately 1/sqrt(188), or 0.073x that of the raw or resampled spectrums (Bechhoefer, 2009).

In Figure **5**, a detailed view of the spectrum is given from 2200 to 4000 Hz, showing that the resampled spectrum has more spectral content than the raw spectrum or the TSA spectrum. In the 9000 to 10,500 Hz view, spectral peaks are visible 9220, 9510 9800 and 10,090 Hz.

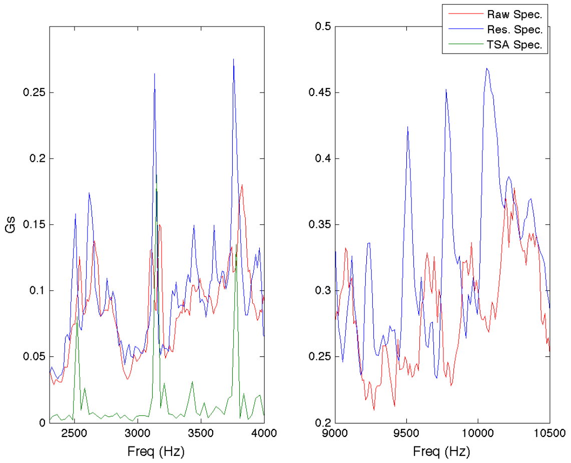


Figure 5. Detail of raw, resampled and TSA spectrum

The 290 Hz difference is close to the modulation rate of an inner race fault, which was 292 Hz. The high, broadband spectrum is indicative of bearing resonance. Because resonance is non-synchronous, the TSA does not capture this bearing resonance.

As an aside, plotting the TSA against the raw spectrum is a good way to identify bearing faults: Frequency content not present in the TSA which are present in the raw spectrum can only be gear mesh frequencies from other shafts in the gearbox, or a bearing fault.

The envelope of the raw and resampled data was taken with a window from 9KHz to 11KHz. This "window" covers the spectrum where bearing resonance is present. This is essential for successful bearing analysis using the envelope technique (Bechhoefer, 2010), see Figure **4**. The raw and resampled envelope spectrum is seen in Figure **6**.

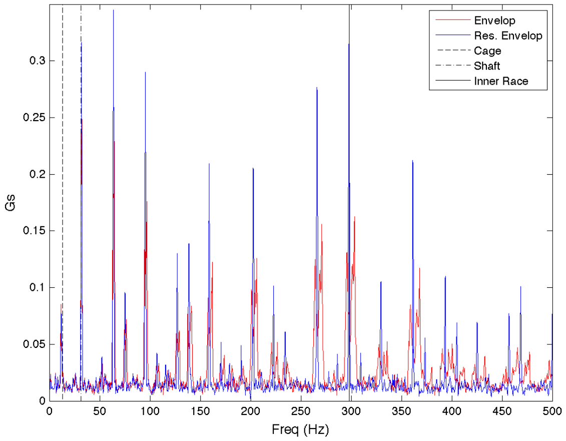


Figure 6. Envelope spectrum of raw and resampled data

Note that the cage (13 Hz), shaft (31 Hz) and inner race fault feature (292 Hz) are overlaid on the spectrum. Clearly, the resampled envelope spectrum fault features have greater energy. This is not a scale issue, as the noise floor for both spectrums are the same. The increased energy is because there is less spreading of energy into neighboring FFT bins. Figure **7** gives a detail view of the spectrum.

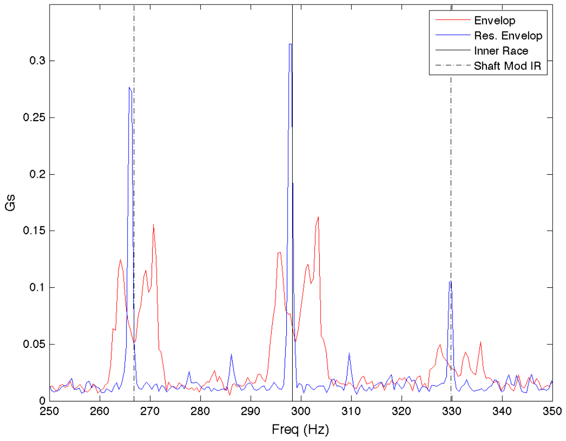


Figure 7. Detail view of inner race fault: envelope spectrum

This view highlights the improvement in resolution of the resampled data over the raw data. Note that from Table **1**, the fault frequency range for the inner race was 292 to 303 Hz, which is validated in Figure **7** in the raw envelope spectrum. Note that maximum value of the raw envelope spectrum was 0.17 Gs. As seen in the resampled spectrum, the true energy value is closer to .32 G's or a 47% error in the original measurement. This smearing of measurement data, results in additional noise in the CI measurement (Figure **2**). This noise in the measured CI is caused by the large variance in shaft speed.

Both inner race modulated by cage and inner race modulated by shaft are also clearly present in the resampled envelope spectrum. This type of information gives a maintainer additional diagnostics, which are missing in the raw spectrum.

3.1. Testing The Hypothesis on Shaft Speed as a Source of Variance

It is hypothesized that at least some of the variance in trend of the inner race energy was due to non-constant shaft speed. As previously noted, it was shown variation in shaft speed smears the measured energy associated with a fault in the spectrum. A test of this hypothesis could be done if one could reprocess the vibration data in Figure **2**. Unfortunately, raw data is collected only once per day (1 our of 144 acquisitions). This subset or raw data was reprocessed using and the measured inner race energy was calculated for the raw envelop spectrum, and the resampled envelop spectrum, over 50 days, and compared in Figure **8**.

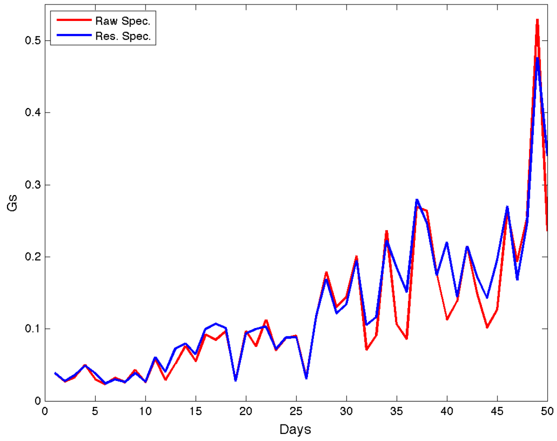


Figure 8. Raw and resampled inner race envelope energy over 50 days

It is easily observed that the resampled envelope energy is higher than the raw envelop. This would be expected in that, because there is less smearing of energy, there is more energy associated with the fault. To formally test this hypothesis: H_0 That resampling does not change the CI variance, vs. H_A That the resampling reduces the CI variance.

The sample variance was calculated from the de-trended data from day 25 to day 48. The sample variance for the raw envelop spectrum was 0.0047 (σ = 0.068), while for the resampled envelope, the variance was 0.002, (σ = 0.045), with an F statistic of 2.3 (approximately 57% reduction in variance). This is significant an alpha of 0.05 and 22 degrees of freedom, reject the null hypothesis that resample does not effect CI variance.

4. Conclusion

Condition monitoring of rotating machinery is complicated by the fact that machines under analysis do not always run at a constant rate. While the time synchronous average can be used to control variance in machine speed for shafts and gears – there is not such standard practice or algorithm to control variance in shaft speed for bearing or other non-synchronous analysis.

In general, it is assumed that the effect of spectral smearing due to variance in shaft speed is small. However, variation in shaft speed is commonly observed in the field. This problem is especially great for wind turbines, in which there is variation due to: changing wind speed, a 3/revolution torque ripple due to tower shadow, and a 1/revolution effect from wind shear. It is not surprising to see a 4% change in shaft speed in a 6 second acquisition.

In this paper, a resampling algorithm was developed in which raw data is synchronized by a key phasor to a shaft under analysis. The resampling process changes the effective sample rate and normalizes the data by removing the effect of changes in shaft speed. It allows both synchronous (shaft/gear) analysis and non-synchronous (bearing, bearing resonance) analysis.

This is demonstrated on a wind turbine high speed shaft bearing with an inner race fault. It is shown that by resampling, the frequency content of the envelop spectrum, which is spread over a frequency of 292 to 303 Hz (14 FFT bin), with raw envelope spectrum of 0.17 Gs. For the resampled spectrum, the true energy value is closer to .32 Gs. For this example, the raw energy spectrum had an error of 47% when compared to the resampled spectrum.

The hypothesis was tested that the resampled envelope energy for a fault would have lower variance. This was tested by reprocessing vibration data for a known fault with 50 samples. The reduction in variance was statistically significant at alpha of .05, or approximately a 55% reduction in variance.

This is a significant improvement in performance. This indicates that variation in speed accounts for a large variance in condition indicator values. Fielding this improved analysis algorithm will result in:

- Bearing faults will be easier to identify,

- That threshold setting will be simplified,
- That trend analysis will be improved and finally,
- That this will facilitate an improved prognostics capability.

The resampling algorithm used linear interpolation, but spline or cubic interpolation could be used.

REFERENCES

ISO 10816 Vibration Severity Standards.

Bechhoefer, E., He, D., (Bechhoefer 2008), Bearing Prognostics using HUMS Condition Indicators, American Helicopter Society 64th Annual fourm, Montreal.

McFadden, P., Smith, J., (McFadden 1985), A Signal Processing Technique for detecting local defects in a gear from a signal average of the vibration. Proc Instn Mech Engrs.

McFadden, P., (McFadden 1987) "A revised model for the extraction of periodic waveforms by time domain averaging", Mechanical Systems and Signal Processing 1 (1) 1987, pages 83-95

Bechhoefer, E., Kingsley, M. (Bechhoefer 2009a). "A Review of Time Synchronous Average Algorithms". Annual Conference of the Prognostics and Health Management Society

Christian, K., Mureithi, N, Lakis, A., Thomas, M., (Christian, 2007), "On the use of Time Synchronous Averaging, Independent Component Analysis and Support Vector Machines for Bearing Fault Diagnosis", First International Conference on Industrial Risk Engineering, Montreal, Dec 17-19 Montreal.

Bechhoefer, E., Kingsley, M., Menon, P., (Bechhoefer, 2009b), "Bearing Envelope Analysis Window Selection Using Spectral Kurtosis Techniques", IEEE PHM Conference, 2011.

BIOGRAPHIES

Eric Bechhoefer received his B.S. in Biology from the University of Michigan, his M.S. in Operations Research from the Naval Postgraduate School, and a Ph.D. in General Engineering from Kennedy Western University. His is a former Naval Aviator who has worked extensively on condition based maintenance, rotor track and balance, vibration analysis of rotating machinery and fault detection in electronic systems. Dr. Bechhoefer is a board member of the Prognostics Health Management Society, and a member of the IEEE Reliability Society. Dr. Bechhoefer is partner and President of Green Power Monitoring Systems, LLC, which is a developer of condition monitoring equipment for remote monitoring.

Brandon Van Hecke received his B.S. in Industrial Engineering from the University of Illinois at Chicago in 2010. He is a Ph.D. candidate in the Department of Mechanical and Industrial Engineering. His research interests include digital signal processing, machinery health monitoring and fault diagnostics based on the evaluation of vibration and acoustic emission signals, and condition based maintenance.

David He received his B.S. degree in metallurgical engineering from Shanghai University of Technology, China, MBA degree from The University of Northern Iowa, and Ph.D. degree in industrial engineering from The University of Iowa in 1994. Dr. He is a Professor and Director of the Intelligent Systems Modeling & Development Laboratory in the Department of Mechanical and Industrial Engineering at The University of Illinois-Chicago. Dr. He's research areas include: machinery health monitoring, diagnosis and prognosis, complex systems failure analysis, quality and reliability engineering, and manufacturing systems design, modeling, scheduling and planning.

Lubricants Health Monitoring

Giovanni Jacazio[1], Marco Libraro[2], Andrea Mornacchi[3], Massimo Sorli[4]

[1,2,3,4]*Politecnico di Torino –Department of Mechanical and Aerospace Engineering, Turin, 10129, Italy*

giovanni.jacazio@polito.it
marco.libraro@polito.it
andrea.mornacchi@polito.it
massimo.sorli@polito.it

ABSTRACT

Fault inception and growth in the components of a mechanical power drive are often heavily affected by the lubricant health. As a consequence, monitoring the lubricant health status and signaling a degradation of the lubricant properties could improve the reliability of the mechanical drive. A further merit of implementing a lubricant health monitoring system is the possibility to move from a time-based maintenance to a condition-based maintenance with the ensuing decrease of operating costs and reduction of the environmental impact. This would be of a particular advantage for those applications in which loss of operation generated by poor lubrication could cause a service disruption or lead to a safety critical condition. The proposed paper presents the initial part of an ongoing research activity on lubricants health monitoring. The work so far performed consisted of the following activities: perform a critical review of the studies addressing the lubricants condition; determine which are the significant features of the lubricants used in the majority of mechanical systems, that need to be extracted to assess the lubricant health; which occurences affect the lubricant health and how the oil health status can be assessed by different techniques , whose relative merits will be discussed; define the progression of the degradation of the lubricant health status; present the initial study of how different lubricant health monitoring techniques can be fused together to develop an efficient on-line lubricant health monitoring system. The paper thus presents the summary of the work performed to establish the fundamentals onto which a reliable PHM system for the lubricants of mechanical drives can be developed, that will be the final objective of the on-going research activity.

1. PROBLEM STATEMENT

Several research activities in prognostics and health management have been performed to evaluate the health status of mechanical components by analyzing the condition of the lubricant used in their operation. Optical and magnetic particle detectors have been proposed as sensors capable of providing useful information on the characteristics of the contamination particles entrained by the lubricant (Sjödin & Westin, 2013). Oil debris monitoring seems in fact to be one of the most effective approaches for PHM of the components of a mechanical drive. In fact, it seems logical that the gradual degradation of a component can lead to the accumulation of various types of particles in the lubricant. Therefore, monitoring such accumulation can provide an indication of the progression of the wear of the mechanical component and hence allow an estimate of its remaining useful life. An example of this technique is the oil debris monitoring sensor developed for the Pratt & Whitney F119, that relies upon sensing the disturbances to an electromagnetic field created by the passage of metallic particles. The sensor is based on an inductive measurement technique which enables detection, count and classification of wear metal particles by size and type.

In all these applications the lubricant is actually used as a diagnostic tool able to provide useful information on the health status of the mechanical drive, while no specific attention is paid on the health of the lubricant itself.

However, an ongoing research on the failure mechanisms for the mechanical components of the electromechanical actuators for aerospace applications showed that fault inception and growth are often heavily affected by the lubricant health. Most of the faults of the components of mechanical drives, such as gears and bearings are in fact caused by insufficient lubrication (Stachowiak & Batchelor, 2005). Poor lubrication leads to a direct contact between the surfaces of the mating components in relative motion with an ensuing progressing wear. The type and rate of the wear depends on several factors, such as component type, rubbing

G. Jacazio et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

speed and contact pressure, but the root cause is normally poor lubrication. Therefore, monitoring the lubricant health status, in particular the oil oxidation stability and not only the quantity of oil debris, and signaling a degradation of the lubricant properties could provide significant benefits and improve the overall reliability and availability of the actuator.

Of course, a simple way of preventing poor lubrication due to a decay of the lubricant properties is to replace the lubricant at small time intervals. Though this ensures good lubrication of the components throughout their life, it entails high operating costs. The costs are not only determined by those of the lubricant, but also by those associated to the maintenance time and to the lubricant disposal. It must be further emphasized that scheduled maintenance most of the time leads to the replacement of a lubricant that is still in good conditions and could effectively perform its function for a much longer time (Bowman & Stachowiak, New criteria to assess the remaining useful life of industrial turbine oils. Lubrication Engineering, 1996). Extending the time interval between two replacements of the lubricant is however a risky business unless a reliable information on the lubricant condition is available, since the operation of the mechanical components without proper lubrication leads to their premature failure.

A possibility for optimizing the combination of the two issues: reducing the frequency of lubricant replacement and ensuring a good lubrication of the mechanical components, is to carry out periodical laboratory tests of the lubricant during an initial operational phase of the actuator. By observing the evolution of the properties of the lubricant during the actuator operation an optimal time interval can be established between two consecutive replacements of the lubricant. Although this procedure is logical, it falls short of providing meaningful results if the actuator can operate in highly variable conditions, as it is the case of the actuators used in the aerospace or mobile machinery applications. In these cases the lubricant characteristics can evolve in a rapid and unpredicted way with the ensuing risk that the established replacement interval, that seemed to be acceptable during an initial test campaign, turns out to be inadequate. This risk is further enhanced if no other monitoring is performed of the operational characteristics of the actuator, such as the alert of possible hot spots that could cause a breakdown of the lubricant properties.

The development of a continuous and reliable method to evaluate the lubricant health status and its associated remaining useful life can thus provide a valuable contribution to the improvement of the availability of a mechanical actuator and to the reduction of its life cycle cost. From a PHM perspective this leads to a paradigm shift: the lubricant is not any longer simply a tool for assessing the health status and the remaining useful life (RUL) of the mechanical components, but it becomes itself an element subjected to continuous health monitoring and whose remaining useful life is determined by appropriate algorithms based on processing the collected data.

The development of an efficient and reliable PHM system for the lubricant goes through three logical steps:

- Step #1 is to clearly identify the physical quantities that provide the most significant indication of the lubricant health status
- Step #2 is to define the most appropriate sensors and measuring techniques that allow the extraction of the previously identified significant features
- Step #3 is the development of the algorithms capable of performing a reliable assessment of the lubricant health status and its RUL starting from the collected data

2. Lubricating actions in mechanical components

Relative motion between two solid bodies causes a series of complex interactions between the mating surfaces of the two bodies that are well described in tribology literature (Stachowiak & Batchelor, 2005). These interactions eventually lead to a wear of the surfaces of the two bodies, which depends on a large number of factors. When a lubricant, either liquid, solid, or gaseous is in contact with the surfaces of the two bodies in relative motion, the mutual interaction between the two surfaces is completely changed and can be even nil if a sufficiently deep layer of lubricant is created to physically separate the two surfaces. The end result is a reduced wear of the bodies surfaces with an ensuing increase of their life and a reduction of the energy dissipation.

Although several types of lubricants are used in mechanical components, the most common ones are oils and greases. Unless the system operational environment mandates special requirements, oil and grease are the preferred type of lubricant because they allow the development of a lubricant layer between the two solid surfaces with a relatively large thickness. This entails a decrease of the interactions between the surface irregularities of the two bodies; the greater the thickness of the lubricant layer, the lower the interactions between the two surfaces and the ensuing wear.

The lubricant layer thickness is strictly related to the physical properties of the lubricant. As an approximation, the layer thickness is proportional to the lubricant viscosity; however, increasing the viscosity brings about an increase of the energy dissipation in the lubricant layer with a consequent increase of the local temperature. This leads to potentially negative effects, such as the oxidation of the lubricant which in turn causes an increase of the viscosity. The selection of the optimal lubricant viscosity is thus the result of a careful tradeoff among several different needs.

The explanation of the lubrication mechanisms has been a long standing research subject that started at the end of the

19th century. In particular, most of the research studies addressed the physics of the formation of the lubricant layer as a function of the operating conditions. Except for the case of hydrostatic lubrication that uses an external power source to create the lubricant pressure necessary to create a separation between the two facing bodies, in all other cases the development of the lubricant layer is tied to the lubricant viscosity, to the relative motion between the two bodies and to the loads acting on them. Three main types of lubrication can be identified: hydrodynamic, elastohydrodynamic and boundary lubrication.

Hydrodynamic lubrication occurs in journal bearings in which two conformal bodies are in relative motion and is originated by the relative sliding velocity between the two bodies.

Elastohydrodynamic lubrication takes place in case of non-conformal contacts subjected to large loads, such as the case of gears teeth and rolling bearings. In this case, a very thin film of lubricant is created, that is anyhow sufficient to greatly reduce friction and wear. Three main factors play a role: hydrodynamics, elastic deformation of the solid bodies under load and the characteristic of the viscosity increase with pressure. This entails that only those lubricants showing a large increase of viscosity with pressure are suitable for elastohydrodynamic lubrication.

Boundary and extreme pressure lubrication is the prevailing type of lubrication in numerous mechanical applications, such as the metal cutting tools. This lubrication type is different from the previous two. An extremely thin film of lubricant is created on a very hard material, which leads to a reduction of the shear stresses with consequent low friction forces. The formation of the lubricant film mainly depends on the chemical properties of both the lubricant and the solid material.

The complex interactions among surfaces, lubricant, contaminats and environment conditions sometimes lead to a temporary reduction of the friction forces, even if severe wear occurs. However in the long term, whichever is the lubrication mechanisms, degradation of the lubricant properties generally leads to an increased wear of the solid surfaces and to greater friction forces. These increse the dissipation of mechanical energy and in consequence the lubricant temperature rises. Thereby the degradation of the lubricant properties is accelerated.

The lubricant positive action is not only limited to adhesion wear, consisting of the loss of material due to the mutual contact between the irregularities of the two solid bodies. Good lubrication also reduces abrasive, corrosive and oxidation wear. The physical properties of the lubricant must be such to prevent the development of chemical reactions on the surfaces of the two solid bodies. These chemical reactions can in fact entail a rapid wear of the bodies surfaces, especially when the reaction product consists of an oxide that can be easily entrained by the relative motion between the two bodies. Chemical reactions can be originated by different chemical agents. Of a particular interest is the case of the corrosive action of the lubricant itself. The proper selection of the lubricant is not only dependent on its viscosity characteristics and on the operating pressure and temperature, but also on the chemical reactivity of the lubricant with the bodies materials. The lubricant must be selected to be non-reactive with the materials of the lubricated surfaces; improper or anomalous operating conditions, or a prolonged use can however bring about chemical modifications of the lubricant which can then become a source of corrosion for the wetted materials.

3. Definition of the lubricant health status

Whichever is the lubricant type, whether its base stock is mineral or synthetic oil, and whether the lubricant is oil or grease, its health status is defined by a few measurable properties. With reference to the fundamentals of lubrication outlined in the previous section, the most significant features describing the lubricant health and its ability of performing its function are:

- Viscosity
- Corrosive behavior
- Contamination level
- Oxidation stability

Other characteristics of the lubricant, such as pour, flash and fire points are important for the initial selection of the most suitable lubricant for a given application, but are less critical for defining the lubricant condition during its life.

Viscosity is the most important parameter for determining the lubrication function since it directly affects the thickness of the lubricant layer and the energy dissipation along the sliding surfaces. The viscosity of a given lubricant depends on several factors, among which the amount of lubricant oxidation and the presence of water. Moreover, for a lubricant in optimal conditions, the viscosity decreases rapidly with increasing temperature while it increases with increasing pressure. Appropriate equations can describe the variation of the lubricant viscosity with temperature and pressure (Stachowiak & Batchelor, 2005).

The level of contamination affects both the lubricant behavior and its life. Water is a very common contaminant in the lubricant and reduces its viscosity, while promoting corrosion and oxidation of the bodies surfaces. Other typical contaminants are sulphur and chlorine which are present in different amount in the base stock and must be removed in the refinment process. Sulphur and chlorine make the lubricant highly corrosive, though they might improve its characteristics when they are present in a minimal amount. Debris coming from the wear of the sliding surfaces are a further common contaminant and their measurement can provide an indication of the condition of the bodies surfaces.

Although all the above mentioned features globally define the lubricant health status, the individual feature that has been recognized as mostly significant for monitoring the lubricant health and defining its remaining useful life is the degree of lubricant oxidation (Bowman & Stachowiak, New criteria to assess the remaining useful life of industrial turbine oils. Lubrication Engineering, 1996). The presence of oxygen causes the lubricant molecules to chemically react with oxygen created corrosive compounds which also normally increase the lubricant viscosity. This process takes place at any operating condition, but is enhanced by high temperatures and by the possible presence of catalysts. The compounds produced by the lubricant oxidation are highly corrosive limiting the lubricant life.

Resistance to oxidation can be improved by a prolonged refining process for removing aromatic fractions of the lubricant as well as molecules containing sulphur, oxygen and nitrogen. However, this is an expensive process which also has the drawback of decreasing the performance for the case of boundary lubrication. Therefore, the increase of the lubricant life is normally pursued by introducing special additives into the lubricant that counter the effect of free radicals and peroxides created by the oxidation process.

The conclusion of the analysis performed on the degradation of the characteristics of a lubricant was that though there are different possible degradation patterns, there is a compelling evidence that the most important root cause of lubricant degradation is the oxidation process of its molecules. Therefore, the amount of lubricant oxidation can be taken as the most significant feature for establishing the lubricant health and for determining its remaining useful life. The techniques for measuring the degree of lubricant oxidation, their reliability and their possible applicability to develop a lubricant PHM methodology are outlined in the following sections of this paper.

4. Techniques for Measuring the Degree of Lubricant Oxidation

The techniques for measuring the degree of lubricant oxidation can be divided in two groups: conventional and non-conventional techniques, which are briefly described in the following paragraphs.

4.1. Conventional techniques

The best known conventional techniques are those defined by ASTM specifications, namely: RBOT (ASTM D2272), TOST (ASTM D943) and TAN (ASTM D974).

The RBOT (ASTM D2272 - 11, 2009) method consists of putting a given amount of lubricant in a sealed volume containing oxygen and copper, and measuring the time necessary for the oxidation process to cause a defined decrease of the pressure inside the container.

The TOST method (ASTM D943, 2010) measures the time necessary for 300 ml of lubricant to react with oxygen in presence of a catalyst based on iron and copper. The test is concluded when the lubricant acidity reaches the level of 2 mg of KOH/g.

The TAN (Total Acid Number) (ASTM D974 - 12, 2011) method enables to determine the quantity (mg) of potassium hydroxide (KOH) necessary to neutralize 1 g of lubricant. The measure is performed taking advantage of a pH indicator. The greater the concentration of acid corrosive compounds in the lubricant, the greater the quantity of potassium hydroxide needed for neutralizing the lubricant.

All the above described methods are off-line type methods and are hardly usable for a lubrication PHM system in which the characteristic feature extraction must be performed on a regular basis with an on-line device and without commands issued by an operator. Moreover, the test times of these methods are very long: from 40 hours of the RBOT method to even 1500 hours for the TOST method. Therefore, a lubricant PHM system must rely on innovative non-conventional methods that are presented in the following paragraphs.

4.2. Sealed capsule DSC and Pressurized DSC

Differential scanning calorimetry (DSC) is a new technique for analyzing the thermal properties of a fluid. The DSC enables the determination of the characteristics of a lubricant sample by measuring the differences between the thermal fluxes through the test sample and a reference sample (typically oxygen) when the two samples are heated or cooled according to a specified procedure. The temperatures differences between the test and the reference samples are proportional to the heat fluxes and are originated by endothermal or exothermal reactions occurring in the test sample.

If the fluid under test is a lubricant, it is then possible to estimate its oxidation degree by measuring the induction time, i.e. the time necessary to obtain an impending oxidizing reaction in the lubricant (Sharma & Stipanovic, 2003). If the test sample has a limited oxidation degree, there is a relatively high concentration of antioxidant agents, hence there will be a large induction time before an oxidizing reaction starts. On the contrary, if the oxidation degree of the test sample is high, the induction time will be reduced.

The standard DSC technique is however not suitable for the objective of developing a lubricant PHM system. The high volatility of most lubricants causes an appreciable evaporation during the test leading to a loss of the lubricant mass that greatly disturbs the measuring process (Bowman & Stachowiak, Determining the oxidation stability of lubricating oils using sealed capsule differential scanning calorimetry (SCDSC), 1996). Modified techniques were thus recently developed starting from the basic DSC with the specific purpose of measuring the oxidation degree of a lubricant. These techniques are known as High Pressure DSC (PDSC) and Sealed Capsule DSC (SCDSC) .

Both PDSC and SCDSC are based on the same operating principle of the standard DSC, albeit with some differences. The PDSC operates in a very high pressure environment in which the lubricant evaporation is extremely low, while in the SCDSC the test and the reference samples are placed in sealed capsule, and no evaporation thus occurs. In both these techniques the greater oxygen concentration also speeds up the lubricant oxidation, thereby reducing the measurement time. The PDSC technique was standardized by the ASTM D6186 specification (ASTM D6186 - 08, 2003), but has so far been used only for research and development since the correlation between the results provided by these tests and the actual behavior of the corresponding lubricant in service has yet to be proven.

4.2.1. SCDSC

Among the different instruments developed to measure the lubricant induction time (OIT), a scansion differential calorimeter with sealed capsules presents several merits, both economically and for the quality of the measure. In these calorimeters, the capsules are hermetically sealed in an environment with purified oxygen. An appropriate selection of the test parameters (Bowman & Stachowiak, Determining the oxidation stability of lubricating oils using sealed capsule differential scanning calorimetry (SCDSC), 1996) enables to perform the test in a maximum time of 35 minutes, which is much lower than required by the conventional techniques.

This technique for measuring the oxidation degree of a lubricant was tested on lubricants with very different composition. In particular, tests were conducted on oil for transformers, for turbines (Bowman & Stachowiak, 1998), for ground vehicles and also for vegetable oils (Fox, Simpson, & Stachowiak, 2001). It is important to point out that not all tests led to clear and satisfactory results. For instance, for the case of lubricants for ground vehicles when the OITs are plotted versus the oxidation times, the corresponding diagram is not monotonous decreasing, therefore the measurement of the OIT alone is not sufficient to define the oxidation degree. On the contrary, tests performed on turbine oils are very accurate and promising (Figure 1).

It is interesting to correlate the results obtained from the SCDSC tests with the kinematic viscosity and composition of the same lubricant (Figure 2).

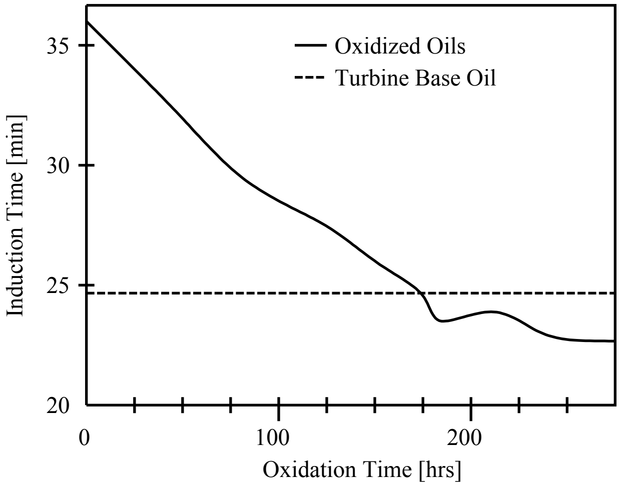


Figure 1. Induction time vs. oxidation time measured for a turbine oil by SCDSC technique (Bowman & Stachowiak, Determining the oxidation stability of lubricating oils using sealed capsule differential scanning calorimetry (SCDSC), 1996)

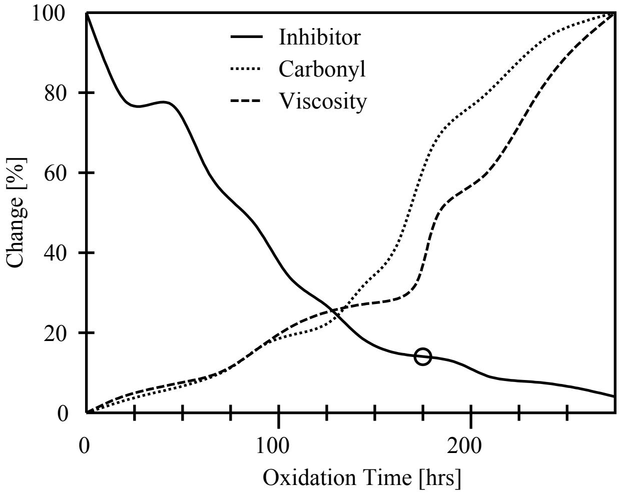


Figure 2. Percentage variations of a turbine oil inhibitor, carbonyl and viscosity vs. oxidation time for a turbine oil (Bowman & Stachowiak, Determining the oxidation stability of lubricating oils using sealed capsule differential scanning calorimetry (SCDSC), 1996)

The percentage change of the quantities shown on the y-axis of Figure 2 refers to the variations with respect to a new oil; as for the viscosity, the initial value of the kinematic viscosity was 68 cS. The chemical composition can be determined with the Fourier transform infrared spectroscopy (FTIR), that enables to monitor the concentration of the antioxidants and of the oxidation products.

As an example, for the case of turbine lubricants a larg increase of viscosity is noticed after 175 hours of oxidation induced at 140 °C in a laboratory. The viscosity increase is due to the polymerization of the lubricant molecules which on its turn is generated by the lubricant oxidation. The sharp viscosity increase occurs when the antioxidant concentration is only 15% of its initial value, and this is the condition for which the lubricant remaining useful life is considered to be zero (Bowman & Stachowiak, New criteria to assess the remaining useful life of industrial turbine oils. Lubrication Engineering, 1996).

Measuring the antioxidant concentration is by itself not always sufficient to determine the oxidation degree of a lubricant. Cases were observed in which the oxidation products remain small also for small amount of antioxidants, and cases were also observed with presence of high amount of oxides and antioxidants at the same time.

The oxidation induced in a laboratory keeping the lubricant at a constant high temperature allows for measuring the time necessary to bring the lubricant life to the end. This time is a function of the oil temperature and of its contamination level. It allows for identifying on the diagram induction time vs. oxidation time the induction time t_i for which the lubricant remaining useful life is reduced to zero. This time is a characteristics of the lubricant and a map of t_i versus RUL can be created (Figure 3).

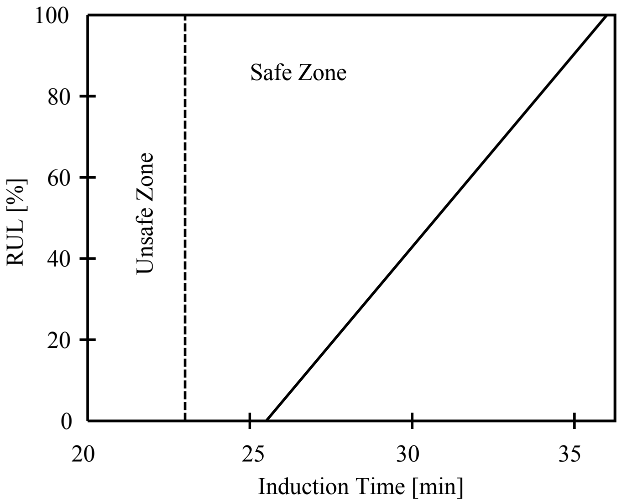


Figure 3. RUL vs. induction time map for a turbine oil analyzed by the SCDSC technique (Bowman & Stachowiak, 1998)

For the case of a specific turbine oil the map obtained from a linear approximation has a correlation coefficient of 99% with experimental data from lubricant oxidized in a laboratory and from lubricants spilled from industrial plants. Of course, such very high correlation coefficient is not obtained for all lubricants that have been tested.

It is interesting to notice that for some types of lubricants there is a fairly good correlation for the evaluation of the oxidation degree obtained from the SCDSC technique and from standard techniques (RBOT, TAN); a correlation coefficient up to 0.94 is obtained between SCDSC and RBOT techniques. It is thus apparent that SCDSC is a technique to be positively considered in the implementation of a lubricant PHM system since it greatly reduces the measurement time with respect to the conventional techniques.

A research was performed to assess the reliability of SCDSC in case water and metal particles are present in the lubricant. There could be a risk of wrong estimate of the oxidation degree of a lubricant taken from industrial applications for which the contamination degree is not known. Measurements of the induction time for lubricants contaminated with water, solid copper, solid iron and soluble iron ions (Bowman & Stachowiak, Application of sealed capsule differential scanning calorimetry part I: Predicting the remaining useful life of industry-used turbine oils, 1998) showed that only iron ions appreciably affect the induction time, but they are released only in case of very high corrosion and if sludges are present in the lubricant.

4.2.2. PDSC

In Pressure Differential Scanning Calorimetry (PDSC) the oil volatility is decreased by the high pressure making sealed capsules unnecessary. The reference and the sample are in the same environment and oxidation is promoted by oxygen flow rate.

The OIT can be measured in two different conditions: isothermal method and temperature ramp method. For some specific combinations of oil additives the temperature ramp method results fit very well oxidation measures obtained by D 943 method, while for other combinations the results may present some differences (Sharma & Stipanovic, 2003). The ASTM D6186 method is an isothermal method, but the results do not correlate well with those obtained by D943 method. This is the reason why the ASTM D6186 method is not used to evaluate the oxidation stability of used industrial oils. Extensive studies were made in order to improve the repeatability and accuracy of the results of isothermal method. For this purpose, multiple tests were performed by varying one parameter at a time (Sharma & Stipanovic, 2003). In addition, researches proved that soluble metal catalysts improve the results and shorten the OIT measuring time. In conclusion, the optimized PDSC isothermal method showed some improvements against the ASTM D6186: shorter OIT, better results than ASTM method and good correlation with D943 for a set of oil with the same additive package.

In general, thermal methods based on the DSC show a fairly good correlation between the thermal properties of the lubricants and their oxidation stability, so in the future these methods could be used in order to evaluate the oxidation of used industrial oils. Moreover, the time required to measure the oxidation stability (10 – 40 min) is significantly shorter than the standard methods. Nevertheless, it is hard to believe that they could be integrated in online prognostics and health management systems. They could, however, be worthy instruments for offline health monitoring programs.

4.3. Electrochemical impedance spectroscopy

The Electrochemical Impedance Spectroscopy (EIS) is widely applied to determine some properties of materials. It allows for measuring the impedance of a fluid sample by applying a sinusoidal voltage signal *V(t)* with small amplitude V_0 and with a frequency *f* while measuring the current response *I(t)*. The impedance *Z* is the ratio of *V(t)* and *I(t)*, hence it depends on the frequency *f* which usually varies in a wide range during a test (from μHz to GHz) (Lvovich, 2012).

For an ideal impedance consisting of a parallel combination of an ideal resistor and an ideal capacitor the Nyquist plot of the impedance is a semicircle. But for real lubricants the results are different and they depend on the test conditions and the properties of the lubricant. The analysis of the experimental results leads to the development of an equivalent circuit model of the oil which is characterized by a similar Nyquist plot. In this way any variation of the experimental data caused by the oxidation process may be related to one or more variations of the equivalent model parameters (resistances and capacitors).

Assuming that the test conditions are constant, the Nyquist plots of a not oxidized oil and of an oxidized one show some differences (Figure 4) because of their different chemical composition. By analyzing them it is possible to estimate the oxidation stability of the oxidized oil (Lvovich & Smiechowski, 2006).

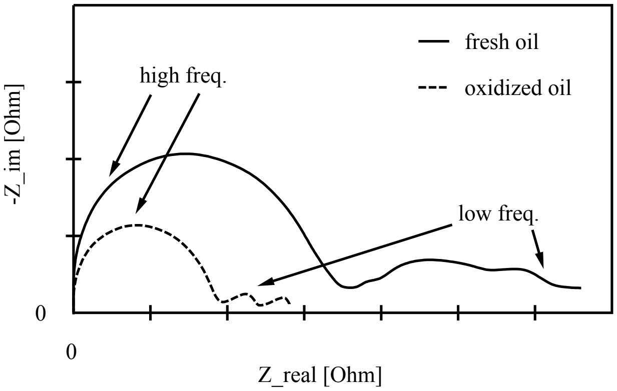


Figure 4. EIS measures concerning a fresh and an oxidized oil in Nyquist plot form (Lvovich & Smiechowski, 2008)

Recent researches coped with lots of lubricants conditions, focusing on the contamination and oxidation influences on EIS spectrum. In the low–frequencies range, where some ambiguous results were collected in the past, improvements were gained by the application of the Non-linear Electrochemical Impedance Spectroscopy NLEIS (Lvovich & Smiechowski, 2008). However, the development of equivalent circuit models able to describe physical and chemical processes concerning the interaction among electrodes, lubricant and other agents is very complex and requires extensive validation. To overcome this issue, some researchers tried to correlate EIS experimental data with laboratory tribology results (viscosity, TAN, soot measurement, etc) by applying Symbolic Regression and taking notice of the chemical interactions (Byington, Mackos, Argenna, Palladino, Reimann, & Schmitigal, 2012). They implemented an online EIS equipment on several trucks in order to collect lubricant data continuously and in the meanwhile laboratory tests were conducted on oil samples taken from the trucks. The researchers were able to find out numerical models providing good correlation between the features generated by the oil condition monitor and the laboratory tests results by applying symbolic regression. Furthermore, since the models predicted well the laboratory tests results, they argued that the EIS oil condition monitor could be useful for assessing the health and estimating the RUL of the oils in mechanical equipments.

The EIS has surely a great potential in lubricants analysis, prognostics and health management, but there are some critical issues affecting this method: long measurement sessions to gather the complete impedance spectra and high instrumentation cost because of the requirement of high quality electronic equipment, even if cheaper EIS instrumentation was developed (Carullo, Ferraris, Parvis, Vallan, Angelini, & Spinelli, 2000). As a consequence, its implementation cannot be satisfatory for several mechanical applications.

4.4. Viscosity and Dielectric constant

The researchers in the field of lubricant prognostics and health management tried to correlate measures of some physical properties of oils with their contamination (water, metallic particles and soot) and oxidation status. Most of all, oil viscosity and dielectric constant were investigated because of their sensitivity to both the contamination and the oxidation (Zhu, Yoon, He, Qu, & Bechhoefer, 2013). In particular, the dielectric constant measurement of a lubricant was performed by comparing the capacitance of a thin lubricant layer with the capacitance of an air layer of the same dimensions (Raadnui & Kleesuwan, 2005).

Online oil monitoring systems using both a kinematic viscometer and a dielectric constant sensor for prognostics and health management purposes were hence developed. Some interesting results were gained addressing the water-contaminated lubricants. Indeed, the technical literature offers several numerical models able to correlate well viscosity and dielectric constant variations to water contamination level. It is therefore possible to define the contamination level by measuring these oil properties. Some

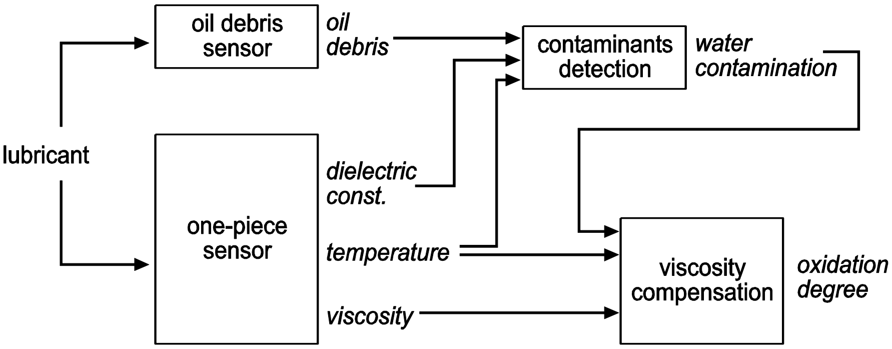


Figure 5. Structure of the proposed lubricant PHM system

tests were carried out for increasing water contamination from 0% to 5% while the measurements of viscosity and dielectric constant were performed. The particle filter technique eventually showed a good correlation between the oil RUL and its viscosity and dielectric constant measures. However, the effectiveness of the method was not proven for different kinds of contamination or in case of oxidation.

Kinematic viscometers and dielectric constant sensors are affordable and commercially available sensors and comparing with other instruments they could be fairly easily integrated in online oil monitoring systems. In addition, integrated sensors were recently developed for vehicle and aerospace applications measuring density, viscosity, dielectric constant and temperature of the oil at the same.

In the authors opinion, the estimation of the lubricant health taking advantage of cheap sensors could give a valuable contribution for the development of an effective and reliable lubricant prognostics and health management system. However, more studies need to be conducted in order to identify the actual causes of lubricant degradation.

5. Development of an Efficient Lubricant health monitoring System Based on Data Fusion

It was outlined at the beginning of this paper that the main objectives of the initial phase of the research were to identify the characteristic feature for lubricant deterioration, to perform a critical review of all existing techniques used to measure different properties of a lubricant and to eventually define the structure of an effective and reliable lubricant PHM system. In particular, a comprehensive review of the existing techniques that can contribute to estimate the health status of a lubricant was summarized in section 4. It was shown that all the techniques providing a direct measurement of the lubricant oxidation degree do not suit well for a PHM system since they require long measurement times and cannot be used as on-line measuring devices. The technique based on viscometer and dielectric constant sensor has the advantage of being implementable in an on-line device, but it does not provide a direct measurement of the lubricant oxidation degree, which was recognized as the characteristic feature defining the lubricant health status. Actually, the lubricant viscosity is affected by its degree of oxidation, but it can also be affected by other factors, such the amount of contaminants. It was shown that viscosity is heavily affected by the oil oxidation and its value can rise up to 140% of the initial value for significantly oxidized oils, but viscosity variations can also be generated by other factors, such as the presence of water and metal particles.

The basic idea for the development of an efficient lubricant PHM system is then to combine the information provided by the measurements of viscosity and dielectric strength with that obtained from a particle detection sensor. Data fusion and processing by means of suitable algorithms will thus allow the necessary feature extraction and the assessment of the lubricant health status.

A further advantage of data fusion from multiple sensors is the possibility to compare data provided by different sensors to detect the cause of the lubricant degradation. This is of a great merit for an oil monitoring system oriented to real industrial applications. In fact, engineers usually are aware of the main issues concerning the degradation of a specific mechanical equipment working in a well-known environment, hence they choose additive packages to make the lubricant life longer. However, an issue is to actually understand which is the deteriorating agent being responsible for the lubricant replacement. Furthermore, detecting the origin of deterioration allows for evaluating correctly the available data. As an example, the dielectric constant of a lubricant grows for different types of contamination: water, metal particles and dust (Raadnui & Kleesuwan, 2005). Obviously it does not represent an issue for laboratory-contaminated samples, while it is an important issue if the sample has just been spilled from a mechanical drive. In fact, some contamination agents affect lubricant deterioration and some mechanical equipments wear more than others.

The authors thus positively came to the conclusion that a reliable lubricant PHM system can be developed by fusing the information obtainable from two sensors:

- One-piece sensor able to measure viscosity, dielectric constant and temperature of the lubricant at the same time
- Magnetic and/or optical particle detection sensor

These sensors provide necessary and sufficient data for calculating the lubricant oxidation level, hence its health condition. In fact, as it was previously described, the variation of the lubricant viscosity is strictly related to its oxidation level, but the viscosity measure requires compensation for oil temperature variation and water contamination. However, temperature is also measured by the one-piece sensor and the water amount can be determined by fusing the information on the value of the dielectric constant measured by the the one-piece sensor and the data obtained from the particle detection sensor which allows the PHM system for determining if the dielectric constant variation is a consequence of only water contamination, only particle contamination or both. At the same time his sensor provides information about the health of the mechanical components.

The described oil health monitoring system has then the following main features:

- It is an on-line system, hence suitable for being used in an operating equipment such as an actuator, or a mechanical drive
- Sensors are cheap and commercially available. In addition they could be easily integrated in an online monitoring system
- Data fusion can lead not only to the evaluation of the oxidation degree, but also to the identification of the amount of water and particles contamination, thereby allowing to better define the lubricant degrading agent

It is important to emphasize that the proposed PHM system is not intended as a possible replacement to standard measurement techniques, but as a system to be applied in operating mechanical equipment for a continuous assessment of the lubricant health status. Its purpose is to perform a robust numerical correlation between multiple lubricant physical properties that are easily measurable, and its oxidation degree, thereby allowing the implementation of a lubricant PHM system suitable for industrial applications. Having defined the lubricant PHM philosophy and the associated sensors suite, the next step of the research will be focused on the development of the algorithms needed to perform data fusion and RUL estimate, as well as run simulations to assess the PHM system robustness. Following that, the final step will be the PHM system validation.

Validation can be performed by testing laboratory-prepared samples characterized by different contaminating agents amount and different oxidation levels. The first step will concern the PHM system ability to detect and measure the contaminants and whether it is able to correctly estimate the water contamination level by simultaneously measuring the oil debris and the oil dielectric constant.

Furthermore, numerical models performing the viscosity and the dielectric constant compensation that were already successfully validated for non-oxidized oils, need to be validated also for oxidized and contaminated oils.

Last, after the PHM system proves its capability of calculating correctly the compensated-viscosity, it will be important to find out if the measurement of the dielectric constant could be used to further improve the estimate of the lubricant oxidation degree.

6. Conclusions and further work

The initial phase of a research activity being performed by the authors on health management and prognostics of lubricants was described, highlighting the most important causes of lubricant degradation and how it is possible to measure them by conventional and non-conventional techniques.

Furthermore, an original and innovative PHM system for lubricants is proposed. It employs two commercially available sensors (a magnetic oil debris sensor and a multi-purpose sensor) which can be easily integrated in an online lubricant monitoring system. It is the authors' opinion that this system enables to reliably detect type and amount of contaminating agents and the oxidation degree of the lubricant, thereby providing all necessary information to develop an industrially applicable PHM system for a lubricant.

The next step of the research activity will address the process of data fusion enabling to achieve robust and reliable estimation of the oil contaminants and of the oxidation degree. It will then be possible to move to the final goal of the research: the development and the validation of effective algorithms for lubricants RUL evaluation.

References

Adhvaryu, A., Erhan, S. Z., Liu, Z. S., & Perez, J. M. (2000). Oxidation kinetic studies of oils derived from unmodified and genetically modified vegetables using pressurized differential scanning calorimetry and nuclear magnetic resonance spectroscopy. *Thermochimica Acta , 364*, pp. 87-97.

ASTM D2272 - 11. (2009). *Standard Test Method for Oxidation Stability of Steam Turbine Oils by Rotating Pressure Vessel.*

ASTM D6186 - 08. (2003). *Standard Test Method for Oxidation Induction Time of Lubricating Oils by Pressure Differential Scanning Calorimetry (PDSC).*

ASTM D943. (2010). *Standard Test Method for Oxidation Characteristics of Inhibited Mineral Oils.*

ASTM D974 - 12. (2011). *Standard Test Method for Acid and Base Number by Color-Indicator Titration.*

Bowman, W. F., & Stachowiak, G. W. (1998). Application of sealed capsule differential scanning calorimetry part I: Predicting the remaining useful life of industry-used turbine oils. *Lubrication Engineering , 54* (1), pp. 19-24.

Bowman, W. F., & Stachowiak, G. W. (1996). Determining the oxidation stability of lubricating oils using sealed capsule differential scanning calorimetry (SCDSC). *Tribology International , 29* (1), pp. 27-34.

Bowman, W. F., & Stachowiak, G. W. (1996). New criteria to assess the remaining useful life of industrial turbine oils. Lubrication Engineering. *Lubrication Engineering , 52* (10), pp. 745-750.

Byington, C., Mackos, N., Argenna, G., Palladino, A., Reimann, J., & Schmitigal, J. (2012). Application of symbolic regression to electrochemical impedance spectroscopy data for lubricating oil health evaluation. *Annual conference of the prognostics and health management society 2012.* September 23-27, Minneapolis, MN.

Carullo, A., Ferraris, F., Parvis, M., Vallan, A., Angelini, E., & Spinelli, P. (2000). Low-cost electrochemical impedance spectroscopy system for corrosion monitoring of metallic antiquities and works of art. *IEEE Transactions on Instrumentation and Measurement , 49* (2), pp. 371-375.

Dupuis, R. (2010). Application of oil debris monitoring for wind trubine gearbox prognostics and health management. *Annual Conference of the prognostics and health management society, 2010.* October 13-16 , Portland, OR.

Fox, N. J., Simpson, A. K., & Stachowiak, G. W. (2001). Sealed capsule differential scanning calorimetry: An effective method for screening the oxidation stability of vegetable oil formulations. *Lubrication Engineering , 57* (10), pp. 14-20.

Kauffman, R. E., & Rhine, W. E. (1988). Development of a remaining useful life of a lubricant evaluation technique. Part I: differential scanning calorimetric techniques. *Lubrication Engineering , 44*, pp. 154 - 161.

Koza, J. R. (1992). *Genetic programming: on the programming f of computers by means of natural selection.* MIT Press.

Lvovich, V. F. (2012). *Impedance spectroscopy, Applications to electrochemical and dielectric phenomena.* Wiley.

Lvovich, V. F., & Smiechowski, M. F. (2008). Non-linear impedance analysis of industrial lubricants. *Electrochimica Acta , 53*, pp. 7375-7385.

Perez, J. M. (2000). Oxidative properties of lubricants using thermal analysis. *Thermochimica Acta , 357-358*, pp. 47-56.

Raadnui, S., & Kleesuwan, S. (2005). Low-cost condition monitoring sensor for used oil analysis. *Wear , 259*, pp. 1502-1506.

Sharma, B. K., & Stipanovic, A. J. (2003). Development of a new oxidation stability test method for lubricating oils using high-pressure differential scanning calorimetry. *Thermochimica Acta , 402* (1-2), pp. 1-18.

Sjödin, E., & Westin, P. O. (2013). Comparison of optical and magnetic particle detection systems for detection of pitting damage in low speed hydraulic motors. *The 13th Scandinavian International Conference on Fluid Power, SICFP2013.* June 3-5, Linköping, Sweden.

Stachowiak, G. W., & Batchelor, A. W. (2005). *Engineering Tribology.* Butterworth Heinemann.

Zeman, A., Sprengel, A., Niedermeier, D., & Späth, M. (1995). Biodegradable lubricants: studies on thermo-oxidationof metal-working and hydraulic fluids by differential scanning calorimetry (DSC). *Thermochimica Acta , 268*, pp. 9-15.

Zhu, J., Yoon, J. M., He, D., Qu, Y., & Bechhoefer, E. (2013). Lubrication oil condition monitoring and remaining useful life prediction with particle filtering. *International Journal of Prognostics and Health Management , 4* (20), p. 27.

BIOGRAPHIES

G. Jacazio is professor of applied mechanics and of mechanical control systems. His main research activity is in the area of aerospace control and actuation systems and of prognostics and health management. He is a member of the SAE A-6 Committee on Aerospace Actuation Control and Fluid Power Systems, and a member of the international society of prognostics and health management.

A. Mornacchi is a mechanical engineer and a PhD student in mechanical engineering. His primary interest is in the areas of modelling and simulation of hydraulic and electromechanical servos. He is a member of the servosystems and mechatronics group of the Department of Mechanical and Aerospace Engineering.

M. Libraro is a mechanical engineer. His main research interests is on the prognostics and health management of the electromechanical actuators for aerospace applications. He is a member of the servosystems and mechatronics group of the Department of Mechanical and Aerospace Engineering.

M. Sorli is professor of applied mechanics and of mechatronics. His research interests are in the areas of mechatronics, mechanical and fluid servosystems, spatial moving simulators, smart systems for automotive and aerospace applications. He is member of the TC Mechatronics of IFToMM, ASME and IEEE.

Wear Prognostic on Turbofan Engines

Jérôme Lacaille[1], Aurélie Gouby[1] and Olivier Piol[1]

[1]*Snecma, 77550 Moissy-Cramayel, France*

jerome.lacaille@snecma.fr
aurelie.gouby@snecma.fr
olivier.piol@snecma.fr

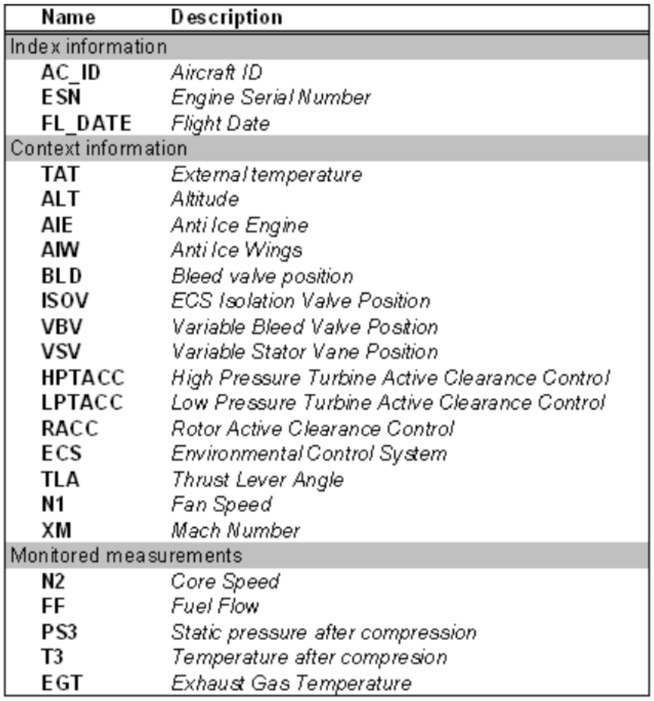
Abstract

One of the most evident characteristic of wear for a turbofan engine is the exhaust gas temperature (EGT). It seems clear that this temperature increases when some carbon deposits on the turbine, when the compressor efficiency diminishes so the fuel flow should increase to produce the same amount of thrust, or even when some unbalance opens the spaces between the turbine and the casing. In any cases, an increase of the EGT should be analyzed because it is a wear symptom of the engine. It is mostly concluded by a water wash in the best case or a shop visit inspection and repair in the worst case. The engine manufacturer defines a schedule plan with its customer based on consumption of the EGT margin. This margin is the amount of available increase of the exhaust temperature before an inspection. Contractually, the engine is restored with a minimum EGT margin after each repair. Thus it is up to the manufacturer to understand how this margin is used to plan shop visits and to the company to estimate the current state of its engine. However the EGT measurement is subject to a lot of noise and the company regularly washes their engines to increase randomly the margin and their capabilities. In this article we present a simple, automatic and embeddable algorithmic method to transform the successive EGT measurements in a delay indicator computed after each flight giving the amount of available use time. One challenge is to take care of the random wash or repair executed by the user. Finally this indicator may be transmitted automatically with the other data broadcasted by the aircraft computer (ACMS/ACARS) and it is used by the manufacturer to prepare his shop logistic.

1. Introduction

Snecma is an engine manufacturer and produces turbofan for most of the short and medium range applications. For medium range specific fleets, aircrafts is doing around 4 flights per day whereas most of short range aircrafts are doing around 10 flights per day. During each flight, measurements are recorded on each engine. They are capitalized on the aircraft computer (ACMS) with other navigation observations such as the altitude, mach speed, etc. During each takeoff and a stable cruise phase a snapshot of measurement is broadcasted to the ground for monitoring.

Table 1. Short list of some snapshot measurements.



Name	Description
Index information	
AC_ID	*Aircraft ID*
ESN	*Engine Serial Number*
FL_DATE	*Flight Date*
Context information	
TAT	*External temperature*
ALT	*Altitude*
AIE	*Anti Ice Engine*
AIW	*Anti Ice Wings*
BLD	*Bleed valve position*
ISOV	*ECS Isolation Valve Position*
VBV	*Variable Bleed Valve Position*
VSV	*Variable Stator Vane Position*
HPTACC	*High Pressure Turbine Active Clearance Control*
LPTACC	*Low Pressure Turbine Active Clearance Control*
RACC	*Rotor Active Clearance Control*
ECS	*Environmental Control System*
TLA	*Thrust Lever Angle*
N1	*Fan Speed*
XM	*Mach Number*
Monitored measurements	
N2	*Core Speed*
FF	*Fuel Flow*
PS3	*Static pressure after compression*
T3	*Temperature after compresion*
EGT	*Exhaust Gas Temperature*

The ground operator follows each engine flight after flight. In fact there are three phases in the ground monitoring process as pictured on Figure 1 next page. The first one is a fast answer mainly based on FADEC error detections (the engine control computer); the second one is the trend monitoring we are interested in this article and which deals

Jérôme Lacaille et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

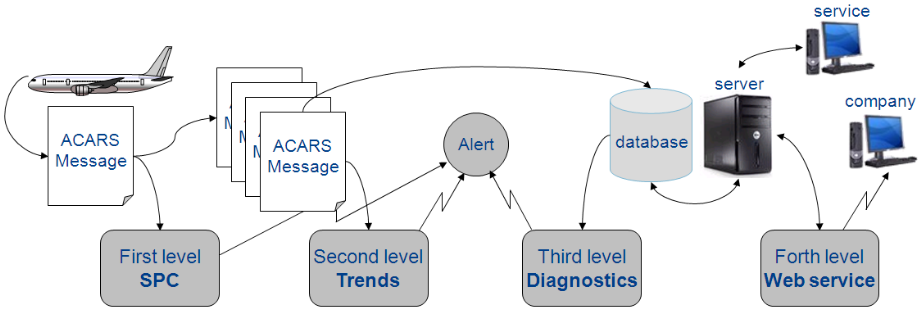


Figure 1. Three different diagnostic levels and a web service.

with successive flights of the same engine; and the last one is fleet monitoring (Lacaille & Come, 2011a) that compares engines to establish a prognostic on its internal state.

Trend monitoring algorithms look at successive snapshots of observations to help the fleet manager analyzing the wear trend of the engine. Some algorithms detect sudden changes of the engine behavior (Lacaille & Come, 2011b; Lacaille & Nya Djiki, 2009; Lacaille, 2009a) and classify the type of change (Bellas, Bouveyron, Cottrell, & Lacaille, 2012, 2013; Come, Cottrell, Verleysen, & Lacaille, 2010; Cottrell et al., 2009; Flandrois, Lacaille, Masse, & Ausloos, 2009; Lacaille, 2009b).

The EGT margin algorithm just looks at the exhaust temperature to predict a given drop and anticipate the need for a potential water wash or shop visit if the margin is really small.

2. Measurements

The EGT margin is the difference between a maximum admissible value for the specific engine application and the observed temperature measured just before the exhaust nozzle. This value is given in °C and decreases progressively to zero. The maximum value corresponds to the certified maximal admissible temperature for an engine type. The value subtracted to this maximum threshold is an estimation of this maximal value for the current engine when measured at sea level with an external temperature equivalent to standard value plus 15°C during the most stressful moment of the takeoff. However, even if acquired with lot of care this measurement still depends on actual external conditions, engine thrust, aircraft speed. A normalization procedure is applied to suppress these last dependencies. This normalization is an analytical certified computation and a mathematical analysis (Lacaille, 2010) confirms the precision of this result.

For example, on Figure 2 the first plot shows the original measurement of an exhaust gas temperature during 300 successive flights. These measurements highly depend on external conditions like the flight mission: altitude, speed gross weight, localization (sea, land or desert) instant of acquisition and external conditions: weather, wind…

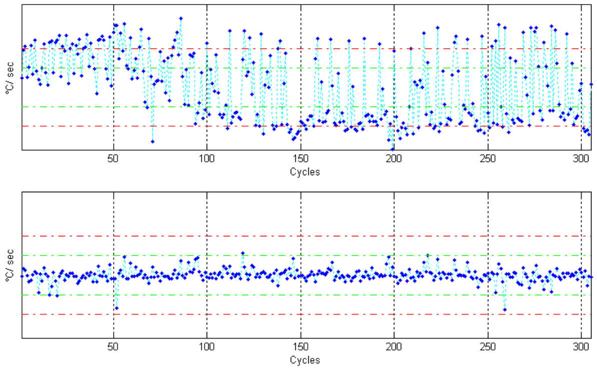


Figure 2. Normalization of the EGT measurement.

The two graphs have the same scale, they are centered on the EGT mean value; the top one is the original acquired measurements and the bottom one is the normalized data. Green dashed lined corresponds to 3σ bounds and red dashed lines to 6σ.

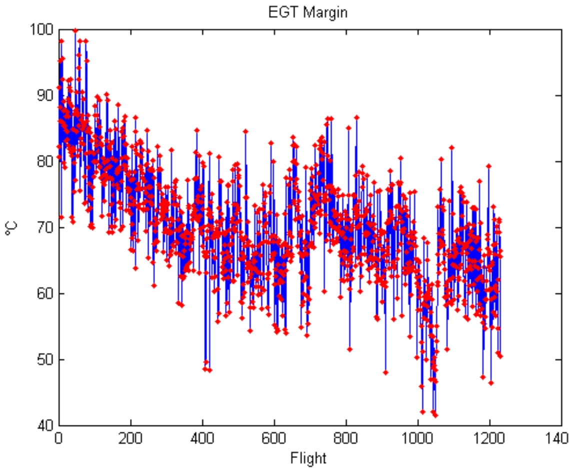


Figure 3. Plot of an exhaust gas temperature margin for an example engine.

A margin computation example is presented on Figure 3. It is based on normalized data. One observes that instead of continuously decreasing as we may have anticipated; the signal is subject to random oscillations. To deal with this behavior we tried two approaches. The first one directly captures this phenomenon with a dynamic model of the temporal evolution. This algorithm is detailed in the next section. The second approach assumes that if the margin grows, it is because the airline decides an intervention on the engine: a water-wash for example. Then it may not be so important to deal with increasing margin and we just try to analyze the downward trends. This second solution is much simpler and may be efficient for the airline but not for the MRO to improve the shop logistic.

The next plot (Figure 4) presents a weekly smoothed version of the preceding signal. This signal clearly presents the oscillations of the margin over time. It seems that the increasing steps appear regularly but with different effects.

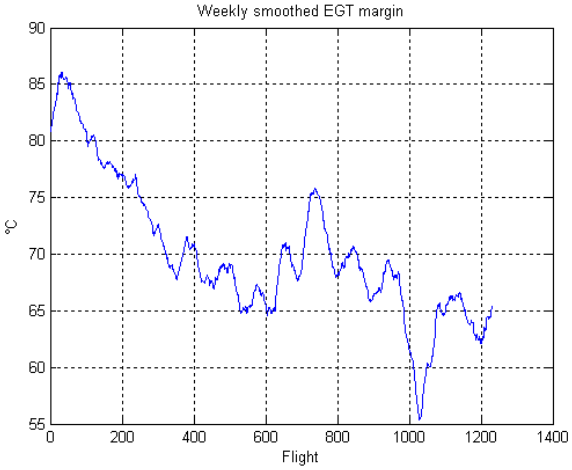


Figure 4. Moving average of the EGT margin measurement over a week.

The goal of the study is to estimate the probability to cross a minimum threshold before a given horizon h. In general this horizon corresponds to the necessary notice before an engine maintenance operation. This is the real need for maintenance operators and it is schematized on Figure 5 by a computation of the probability of detection (POD) after time $t+h$. We will also produce lower bound of the time left before crossing this threshold.

It is sometime easier to think about an estimate of the remaining useful life (RUL) and eventually to produce an indicator that corresponds to the probability that this RUL is less than a given delay. On Figure 5 we also introduce the probability of failure (POF) which is the probability that the remaining useful life is less than h.

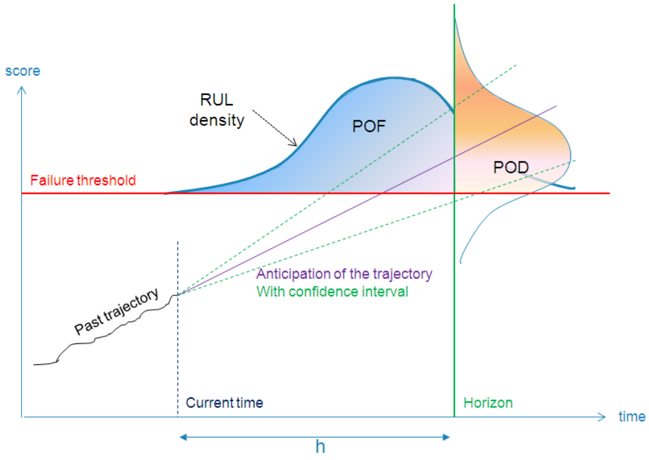


Figure 5. We introduce two output indicators: the probability of failure (POF) directly linked to the remaining useful life (RUL) and the probability of detection (POD) used in the maintenance logistic.

3. Anticipation Algorithm

A nice method to anticipate a continuous process is to model its behavior with an autoregressive model and then filter the signal with a Bayesian update of the state. The standard way to use dynamic filter for anticipation purpose is to throw particles with a sampling scheme (Liu, 2001). However the particle filters or other equivalent Bayesian derived algorithms (Kalman, extended or unscented filters) need an input about the dynamic model to follow (An, Choi, & Kim, 2012), (Saxena, Goebel, Field, & Filter, 2012). In our case, the evolution of the EGT margin is unknown and probably depends on the company process. We need for example to find if it is possible to anticipate the availability of airport technicians for water washes, and it probably depends on the airline politics and its financial stress. Anyway we may assume some regularity and try to find a good autoregressive model: one that predicts this behavior depending on the past observations.

The mathematics to retrieve a good autoregressive (ARMA) model from observations is given in (Lacaille, 1998) with possible adaptation to non linear neural models. But most of today's algorithmic toolboxes are able to fit autoregressive models.

Hence it is possible to retrieve a hidden state (x_t) from observations (y_t) where

$$\begin{cases} x_t = Fx_{t-1} + e_t \\ y_t = Hx_{t-1} \end{cases} \tag{1}$$

with (e_t) a process noise, F the state transition matrix and H the observation matrix that define the dynamic system. The main problem is to find the good rank of the system (dimension of the transition matrix F), but a good guess is given using some information criterion like AIC (Akaike information criterion) which was specifically built for this purpose). This model estimation is done after each new observation and for each different engine. The observation set taken into account for learning extends from six month to one year of observations (~1000 flights) in our study.

The second step, once the dynamic captured on the past observations, is to use a sampling algorithm to simulate particles (probable trajectories) and infer a probability to cross a threshold before a maximum time allowed by the company rules and the availability of a technical team to realize the maintenance operations.

Figure 6 presents the preceding curve where only the first 75% of the first flights are used for learning the dynamic model. Then a sampling algorithms simulate this model, weights are given to each trajectory according to the filter relevance computed on the first "observed" 75% points. The blue-to-green curves at the end of the graph shows the results of the simulations (100 particles in this case). In fact the dynamic is not sufficiently regular to establish a real prediction. We don't capture any regularity in time or in size, hence the probability density of the particles after three months (~300 flights) is almost a Gaussian noise.

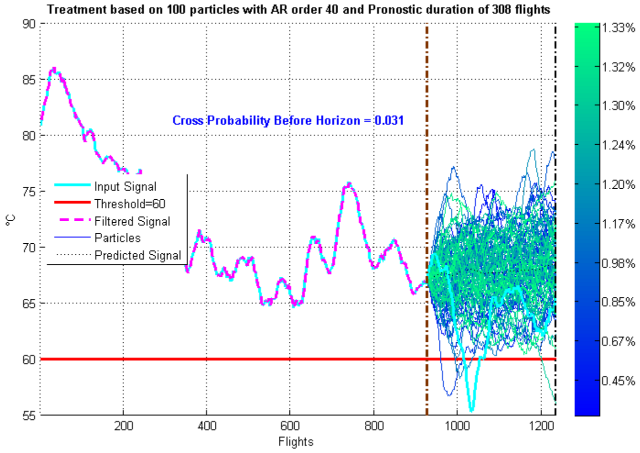


Figure 6. Estimation of the future of the margin at 75% of the available data. The blue-green color map shows the weight of the different particles. Green trajectories represent particles with higher weights.

However, three month anticipation is definitely too optimistic. Figure 7 is a zoom of the preceding graph near the anticipation point. One week corresponds roughly to 25 points and most of the repair procedures may be realized in two days (less than 10 flights) so even if this tool is not very efficient we should not be too hard with ourselves.

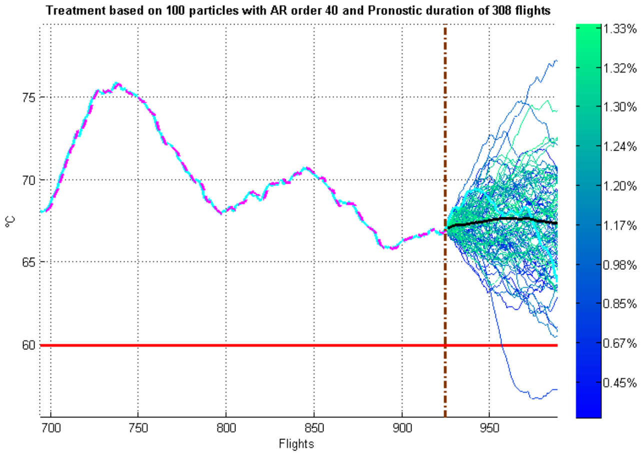


Figure 7. Zoom of the preceding estimation. The black curve presents the weighted mean of all trajectories.

4. Skipping the airline interventions

In fact the underlying problem is much simpler if we don't bother with the random increases of the margin. We understand that those phases are completely unnatural because an engine cannot repairs by itself. Margin increases are the result of company operations and the widths of those phases are only computation artifacts due to smoothing and variation of the acquisition context.

We suppose that the statistic model of the step process $dX_t = X_{t+1} - X_t$ behind our observations is an independent process decomposable into two independent parts:

$$dX_t = U_t + Z_t G_t \tag{2}$$

- a decreasing part $U_t \sim N(-u, \sigma_u)$ for example with a Gaussian distribution with a constant negative trend $-u$;
- and a step function representing the airline maintenance operation build from a product of a binomial distribution and a positive gap $Z_t G_t$
  - where $Z_t \sim B(p)$ with a small probability p to fire an maintenance operation;
  - and a $G_t \sim N(+g, \sigma_g)$ with also for example an Gaussian behavior and a gap of mean size g which is supposed positive and greater than normal trend u.

We are only interested in the decreasing part modeled by U_t, hence the best way is to find a set of instants t where $Z_t = 0$ The probability to have a maintenance operation is $p = P(Z_t = 1)$ is supposed small, so we will detect big gaps and neglect measurement around those instants.

Figure 8 presents the result of such a detector. The graph shows points localized around wide separated instants as we imagined. The smoothing process and previous normalization procedure may induce some artificial thickness but as the number of other (decreasing) points is great enough for a model, so we may just ignore all detected instants without much loss and no risk to pervert the estimation of u and σ_u by mixing other distributions.

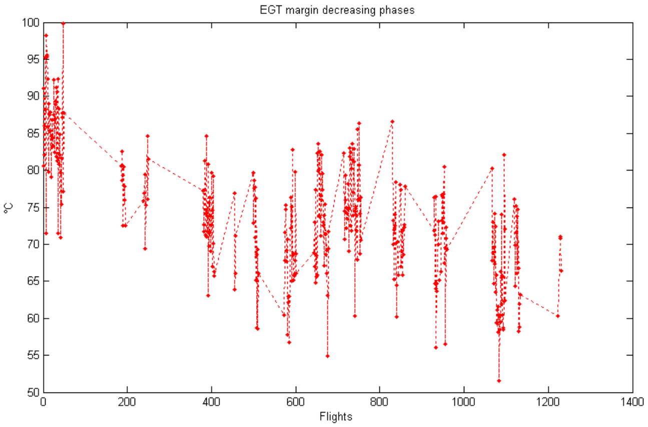


Figure 8. Search of the decreasing phases of the EGT margin. The observations where the EGT margin decreases are concentrated on small periods. The duration of those intervals is only a computation artifact due to the trend estimation.

Once all maintenance intervals suppressed from our dataset we concatenate the decreasing phases by just adding the necessary bias at each phase to ensure to obtain a continuous curve (Figure 9).

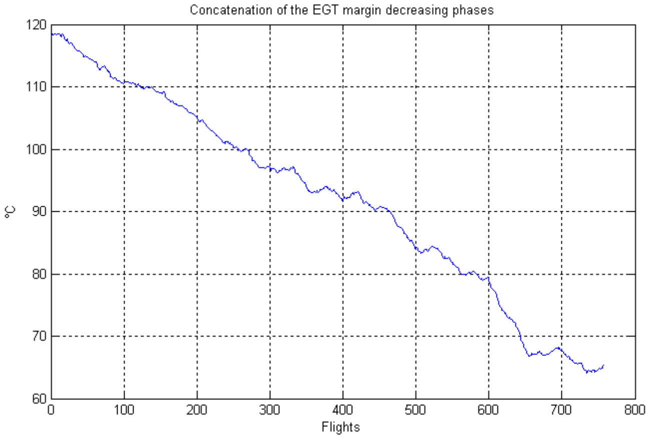


Figure 9. Concatenation of decreasing EGT margin phases.

Each small curve is added after the preceding, with corresponding bias to ensure continuity. In fact we begin with the last measurements (the last interval of data), then we concatenate to the left the preceding measurements' interval and so on. This is done by progressively adding from right to left the values of the variations $dx_t^- = x_{t-1} - x_t$. The last values of the last packet correspond to the real observations (higher values have no meaning, just the general trend is important.)

With no surprise the resulting curve is almost linear. The prediction shown on Figure 10 is a lot simpler. This time no real need to learn the dynamic of the signal: only the main trend is enough. However the algorithm used is still a particle filter because on younger engines like the one plotted on Figure 11 the behavior is not strictly linear but has a slow decrease of its trend.

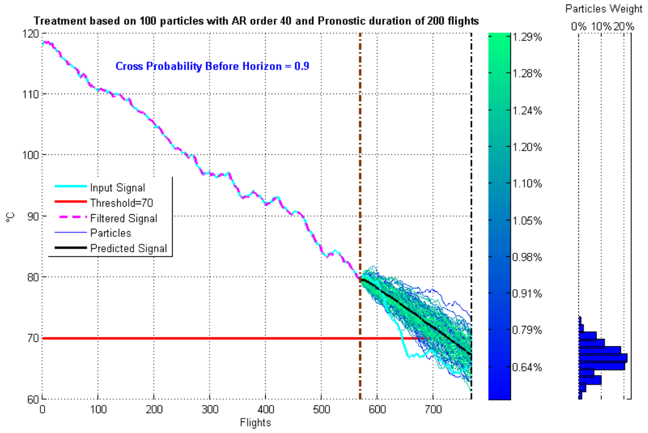


Figure 10. Prediction of the linear trend with a particle filter. The dynamic filter is just used here for presentation so we can observe the variance of the trend coefficient.

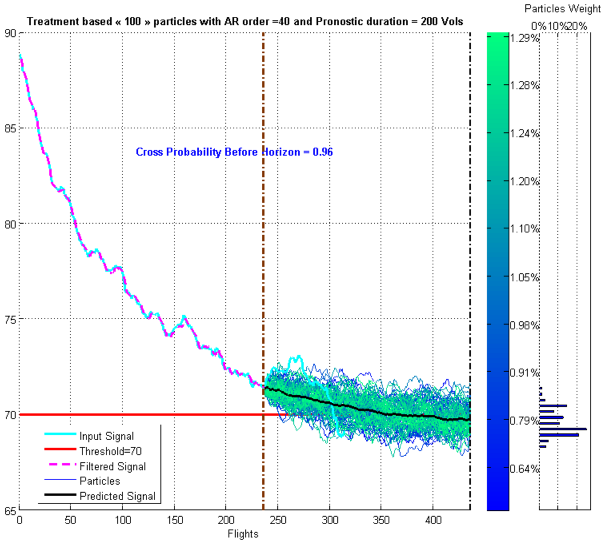


Figure 11. A younger engine with a slow decrease of its margin trend.

Eventually the final algorithm should not use a dynamic filter if a simple regression is sufficient to estimate the trend which value follows a Student law easy to estimate (Besse, 2003).

5. Validation results

There are two required outputs of the algorithm. Those outputs are validated on an experiment set. A validation experiment is a selection of an engine at a given time with EGT margin computations for the 500 past flights for calibration and 200 next flights for confirmation. We used a set of $N = 150$ such experiments built on 50 different engines and 3 observation times per engine. We took wide intervals between each selected time instant in the engine's lives to get rid of local dependencies.

5.1. Estimation of the RUL for 10% of relative decrease of the margin

The requirement was to give an estimation of the RUL corresponding to a decrease of 10% of the last measure of the margin. This estimation should be given with a maximal lower bound ΔT_0 set at 95% of the distribution of the RUL estimation.

$$P\left(RUL_t^{10\%} \geq \Delta T_0\right) = 0.95 \tag{3}$$

where the $RUL_t^{10\%}$ is the delay after current time t before the EGT margin y_t cross a threshold that corresponds to a gap of 10% of its current value.

Figure 12 shows a distribution of the RUL on our set of experiments.

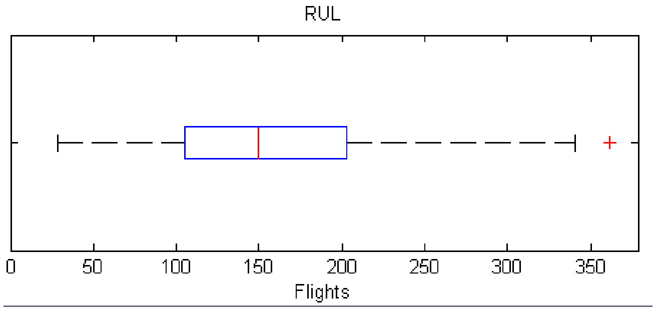


Figure 12. Distribution of the RUL computed on our experiment set.

This first output may be used as an alert indicator. Suppose we asked the client company to repair its engine at time $t + \Delta T_0$. Then the main mistake is to miss the threshold and wait too long. Our quality indicator is the proportion of misdetections ρ_{miss} on a set of N experiments. We will also look at the distributions of the delay error (or misdetection error) because it can modulate our result.

$$\rho_{miss} = \frac{1}{N}\sum_{i=1}^{N} 1_{\{RUL<\Delta T_0\}} \tag{4}$$

On a set of 150 experiments we obtain

$$\rho_{miss} = 22.1\% \pm 6.5\% \tag{5}$$

The 6.5% value after the proportion corresponds to a symmetric confidence interval computed by cross validation on the experiment set.

The next graph (Figure 13) shows a box plot of the misdetection errors.

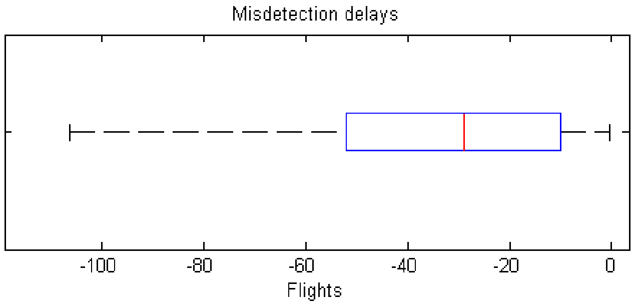


Figure 13. The delays of misdetections.

We observe that the values for those misdetections are around 30 flights and less than 50 with a 95% probability. This is less than two weeks of error for less than 25% of the estimations.

5.2. Probability to cross the 10% margin before one month (100 flights)

$$P_{100}^{10\%} = P(y_t - y_{t+100} \geq y_t * 0.1) \tag{6}$$

This second output is the probability that we cross a threshold before one month. The main goal of this algorithm is to space unnecessary interventions on the engine. The risk there is to prepare the shop when it was not really needed. Our quality indicator is a rate of unnecessary alerts ρ_{ua}, which should be as small as possible. Figure 14 presents a plot of the probability computation according to the real observed RUL.

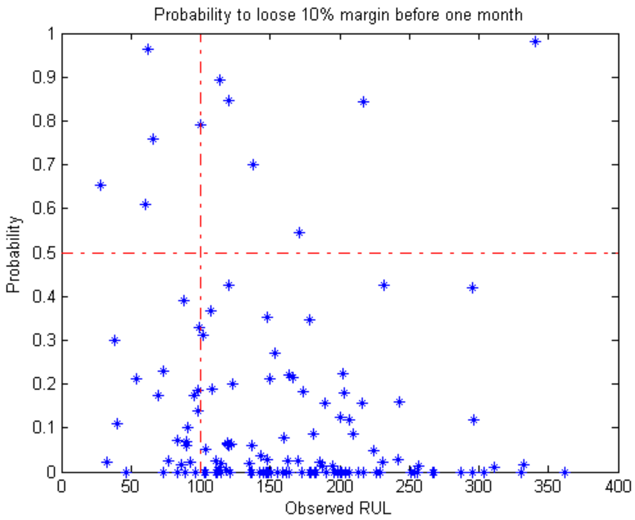


Figure 14. The probability to cross the threshold after a horizon of one month according to the real observed RUL.

The problem appears for observations on the upper right corner of this plot: when the probability is high and the RUL far away (after the month horizon). Arbitrary setting a probability threshold of 50% one may compute the rate of unnecessary alerts as

$$\rho_{ua} = \frac{1}{N}\sum_{i=1}^{N} 1_{\{(P>0.5)\&\,(RUL>100)\}} \qquad (7)$$

This computation gives

$$\rho_{ua} = 6.5\% \pm 3.7\% \qquad (8)$$

The 50% threshold seems to be a logical choice for decision purpose when one observes a probability of degradation and want to decide if an alert should be emitted.

6. Conclusion

We built a really simple algorithm able to predict with good efficiency the evolution of the EGT margin. The risks associated with both output uses (time or probability) are well mastered and not too big. In the first case the prediction error is of at most two weeks for a really small number of cases when the RUL estimation for 10% of margin decrease is between one to two months. On the other hand, the probability of threshold crossing before one month generates less than 10% early interventions.

The next step is to build a general decision rule based on both outputs which will help us to better master the risks.

Nomenclature

ACARS	Aircraft Communications Addressing and Reporting System
ACMS	Aircraft Condition Monitoring System
AIC	Akaike Information Criterion
ARMA	Auto Regressive and Moving Average
LASSO	Least Absolute Shrinkage and Selection Operator
FADEC	Full Authority Digital Engine Control
MRO	Maintenance and Repair Overhaul
OSA-CBM	Open Systems Architecture for Condition-based Maintenance
PFA	Probability of False Alarm
POD	Probability Of Detection
POF	Probability Of Failure
RUL	Remaining Useful Life
SAMANTA	Snecma Algorithm Maturation And Test Application

REFERENCES

An, D., Choi, J., & Kim, N. H. (2012). A Comparison Study of Methods for Parameter Estimation in the Physics-based Prognostics. *PHM*. Minneapolis (MO).

Bellas, A., Bouveyron, C., Cottrell, M., & Lacaille, J. (2012). Robust Clustering of High-Dimensional Data. *ESANN* (pp. 25–27). Bruges (Bx).

Bellas, A., Bouveyron, C., Cottrell, M., & Lacaille, J. (2013). Model-based Clustering of High-dimensional Data Streams with Online Mixture of Probabilistic PCA. *ADAC*, 1–20.

Besse, P. (2003). *Pratique de la modélisation statistique*.

Come, E., Cottrell, M., Verleysen, M., & Lacaille, J. (2010). Self Organizing Star (SOS) for health monitoring. *ESANN*. Brugges (Bx).

Cottrell, M., Gaubert, P., Eloy, C., François, D., Hallaux, G., Lacaille, J., & Verleysen, M. (2009). Fault prediction in aircraft engines using Self- Organizing Maps. *WSOM*. Miami (FL).

Flandrois, X., Lacaille, J., Masse, J.-R., & Ausloos, A. (2009). Expertise Transfer and Automatic Failure Classification for the Engine Start Capability System. *AIAA InfoTech*.

Lacaille, J. (1998). Synchronization of multivariate sensors with an autoadaptive neural method. *Intelligent & Robotic Systems*, *21*(2), 155–165.

Lacaille, J. (2009a). An Automatic Sensor Fault Detection and Correction Algorithm. In American Institute of Aeronautics and Astronautics (AIAA) (Ed.), *Aviation Technology, Integration, and Operations Conference (ATIO)*. Hilton Head (SC).

Lacaille, J. (2009b). Standardized failure signature for a turbofan engine. *IEEE Aerospace conference* (p. 11/0505). Big Sky (MT).

Lacaille, J. (2010). Standardization of Data used for Monitoring an Aircraft Engine. US patent 2010076468A1

Lacaille, J., & Come, E. (2011a). Visual Mining and Statistics for a Turbofan Engine Fleet. *IEEE Aerospace Conference* (p. 11/0405). Big Sky (MT)

Lacaille, J., & Come, E. (2011b). Sudden change detection in turbofan engine behavior. *CM & MFPT*. Cardiff, UK: British Institute of Non-Destructive Testing.

Lacaille, J., & Nya Djiki, R. (2009). Model Based Actuator Control Loop Fault. *European Conference on Turbomachinery Fluid Dynamics and Thermodynamics*. Gratz, Austria.

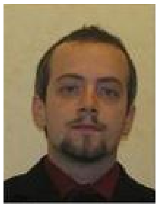
Liu, J. S. (2001). *Monte carlo Strategies in Scientific Computing*. *Book* (p. 245). Springer.

Saxena, A., Goebel, K., Field, M., & Filter, E. K. (2012). Uncertainty Representation and Interpretation in Model-based Prognostics Algorithms based on Kalman Filter Estimation. *PHM*. Minneapolis (MO).

BIOGRAPHY

Jérôme Lacaille is a Safran emeritus expert which mission for Snecma is to help in the development of mathematic algorithms used for the engine health monitoring. Jérôme has a PhD in Mathematics on “Neural Computation” and a HDR (habilitation à diriger des recherches) for “Algorithms Industrialization” from the Ecole Normale Supérieure (France). Jérôme has held several positions including scientific consultant and professor. He has also co-founded the Miriad Technologies Company, entered the semiconductor business taking in charge the direction of the Innovation Department for Si Automation (Montpellier - France) and PDF Solutions (San Jose - CA). He developed specific mathematic algorithms that where integrated in industrial process. Over the course of his work, Jérôme has published several papers on integrating data analysis into industry infrastructure, including neural methodologies and stochastic modeling.

Aurélie Gouby is a Snecma engineer with a double skill in mathematics and informatics. She’s working directly with the PHM team helping system engineers implementing algorithmic solution in the engine.



Olivier Piol is a Snecma engineer working on performance analysis for civil engines. He’s working directly with the PHM team on gas path monitoring.

Electronic Returnless Fuel System Fault Diagnosis and Isolation: A Data-Driven Approach

Bharath Pattipati[1], Krishna Pattipati[1], Youssef A. Ghoneim[2], Mark Howell[2], and Mutasim Salman[2]

[1]*Department of Electrical and Computer Engineering*
University of Connecticut
371, Fairfield Road, U-4157, Storrs, CT 06269, U.S.A
{Bharath, Krishna}@engr.uconn.edu

[2]*GM Global R&D Center*
30500 Mound Road
Warren, Michigan 48090, U.S.A
{youssef.ghoneim, mark.howell, mutasim.a.salman} @gm.com

ABSTRACT

The Electronic Return-less Fuel System (ERFS) manages the delivery of fuel from the fuel tank to the engine. The pressure in the fuel line is electronically controlled by the fuel system control module by speeding up or slowing down the fuel pump. This allows the system to efficiently control the amount of fuel provided to the engine when compared to vehicles equipped with a standard fuel system wherein the fuel pump continuously runs at full speed. A failure in the fuel system that impacts the ability to deliver fuel to the engine will have an immediate effect on system performance. Consequently, improved reliability and availability, and reduction in the number of walk-home situations require efficient fault detection, isolation and prognosis of the ERFS system. This paper develops and implements data-driven fault detection, isolation and severity estimation algorithms for the ERFS. The HIL Fuel System Rig and a Chevrolet Silverado truck were used to collect and analyze the fuel system behavior under different fault conditions. Several data-driven classifiers, such as support vector machines, K-nearest Neighbor, Discriminant analysis, Bayes classifier, Partial- least squares, Quadratic and Linear classifiers, were implemented on a limited set of data for both training and testing. Regression techniques, such as Partial least squares regression and Principle component regression, are used to estimate the severity of faults. The resulting solution approach has the potential to be applicable to a wide variety of systems, ranging from automobiles to aerospace systems.

1. INTRODUCTION

Electronic Return-less Fuel Systems (ERFS) are fast replacing the traditional mechanical fuel delivery systems to transport fuel from the vehicle's fuel tank to the fuel rails and fuel injectors. In the ERFS system, the Fuel System Control Module (FSCM) regulates the pressure on the fuel lines to a desired pressure command from the Engine Control Module (ECM) based on the required engine speed by varying the pulse-width-modulation (PWM) control of the fuel pump. A fuel filter and a pressure regulator may be positioned on the respective intake and outlet sides of the fuel pump. Filtered fuel is thus delivered to a fuel rail, where it is ultimately injected into the engine cylinders. An ERFS includes a sealed fuel tank and lacks a dedicated fuel return line. The regulation of the fuel rate to the injectors improves the fuel economy and eliminates liquid recirculation to the fuel tank. The fuel economy is improved by reducing the electrical load on the alternator and by reducing the rail pressure under most operating conditions. With return-less systems, there is no return line and no circulation of fuel back to the fuel tank from the engine. Consequently, there is no heating of the fuel in the tank and no increase in fuel vapor pressure from driving the vehicle. This reduces the risk of excessive pressure build up inside the fuel tank, vapor leaks, and potential improvements in air/fuel ratio control, and vehicle's emission performance.

Diagnostic and prognostic methods have mainly evolved upon three major paradigms, viz., model-based (Chiang, Russel, & Braatz, 2001), data-driven, and knowledge (experience)-based approaches. The model-based approach uses a mathematical representation of the system and thus incorporates a physical understanding of the system into the monitoring scheme. A major advantage of the physics-based

Bharath Pattipati *et. al*. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

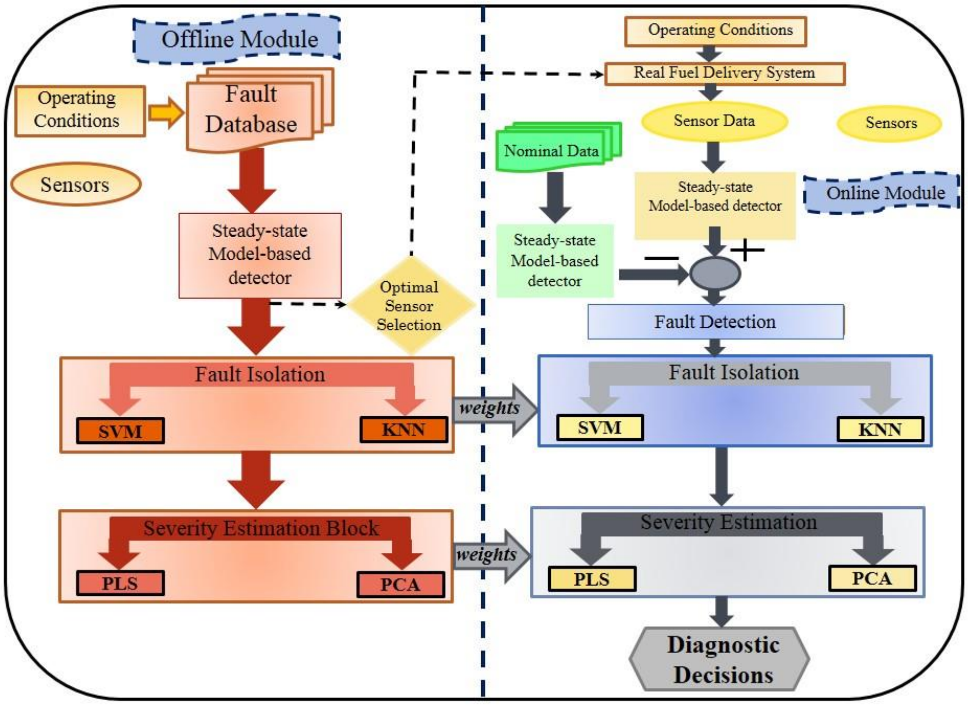


Figure 1. Framework for real-time fault detection and diagnosis of fuel systems

model is that the model bears certain behavioral resemblance to the actual system, which can be very useful in the design of a diagnostic procedure. However, models developed from first principles are seldom used for fault diagnosis in automotive industry mainly because of their complexity. In addition, automotive system dynamics are often nonlinear,

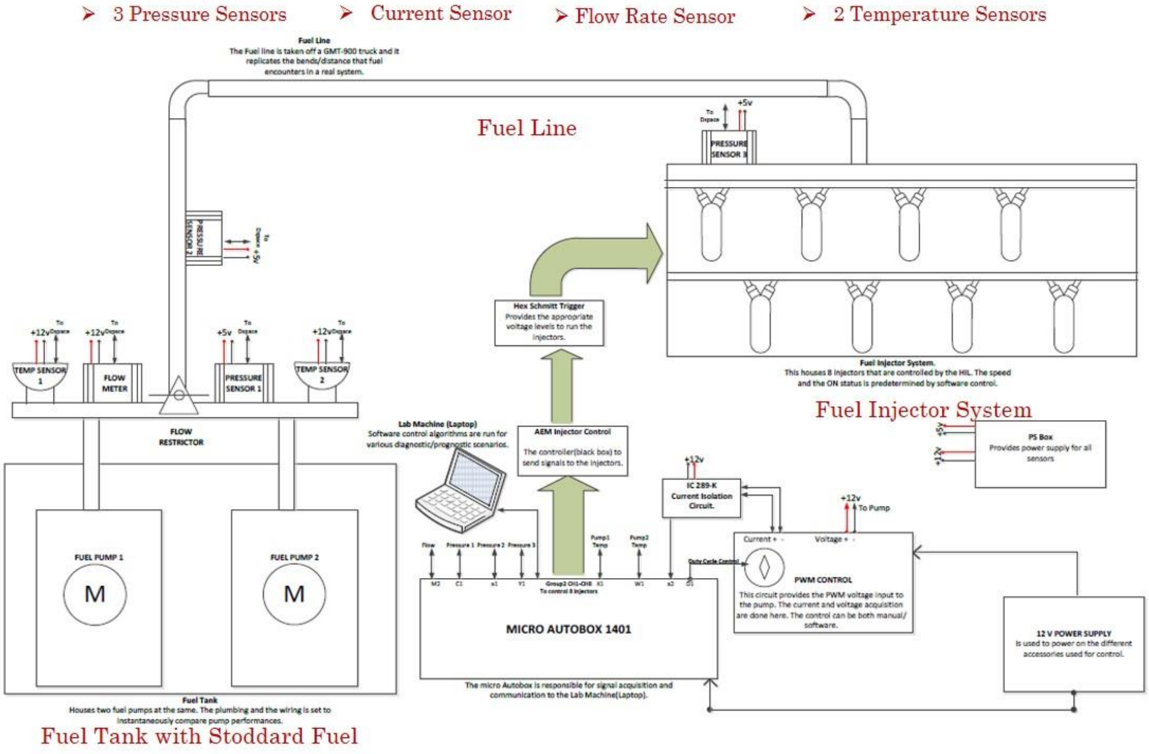

➢ 3 Pressure Sensors
➢ Current Sensor
➢ Flow Rate Sensor
➢ 2 Temperature Sensors
Fuel Line
The Fuel line is taken off a GMT-900 truck and it replicates the bends/distance that fuel encounters in a real system.
Fuel Line
+5v
To Dspace
PRESSURE SENSOR 3
PRESSURE SENSOR 2
To Dspace +5v
+12v To Dspace
TEMP SENSOR 1
+12v To Dspace
FLOW METER
+5v To Dspace
PRESSURE SENSOR 1
+12v To Dspace
TEMP SENSOR 2
FLOW RESTRICTOR
Hex Schmitt Trigger
Provides the appropriate voltage levels to run the injectors.
Fuel Injector System.
This houses 8 injectors that are controlled by the HIL. The speed and the ON status is predetermined by software control.
Fuel Injector System
PS Box
Provides power supply for all sensors
+5v
+12v
Lab Machine (Laptop)
Software control algorithms are run for various diagnostic/prognostic scenarios.
AEM Injector Control
The controller(black box) to send signals to the injectors.
+12v
IC 289-K
Current Isolation Circuit.
+12v
To Pump
Flow
Pressure 1
Pressure 2
Pressure 3
Pump1 Temp
Pump2 Temp
Duty Cycle Control
M2
C1
a1
Y1
Group2 CH1-CH8
To control 8 injectors
X1
W1
a2
D1
MICRO AUTOBOX 1401
The micro Autobox is responsible for signal acquisition and communication to the Lab Machine(Laptop).
Current + -
Voltage + -
PWM CONTROL
This circuit provides the PWM voltage input to the pump. The current and voltage acquisition are done here. The control can be both manual/ software.
12 V POWER SUPPLY
is used to power on the different accessories used for control.
FUEL PUMP 1
M
FUEL PUMP 2
M
Fuel Tank
Houses two fuel pumps at the same. The plumbing and the wiring is set to instantaneously compare pump performances.
Fuel Tank with Stoddard Fuel

Figure 2. GMT 900 based Hardware-in-the-loop rig

which renders the design of fault diagnosis procedures difficult. However, with the advances in computing and an improved understanding of automotive systems, the design of model-based diagnosis schemes is expected to be integrated into the concurrent engineering design process. Model-based methods use statistical estimation techniques based on consistency checks (often termed residuals, “deltas”) generated using observers (e.g., Kalman filters, reduced-order unknown input observers, interacting multiple models, particle filters) and parity relations (dynamic relations among measured variables) to track the component degradations.

A data-driven approach to fault diagnosis and prognosis is preferred when system models are not available (e.g., when subsystem vendors do not share models for competitive reasons), but instead system monitoring data is available (Namburu, Azam, Luo, Choi, & Pattipati, 2007). Here, failure prognosis involves forecasting of system degradation and time-to-failure based on “state awareness” gleaned from monitored data. Neural network and statistical classification methods are illustrative of this approach. The fault scenarios must span the universe of system faults for data-driven approaches to be effective. Mathematical models may be derived (estimated or “identified”) from data as well. Data-driven models include static models and dynamic models. Static models include linear and polynomial models, and look-up tables. Dynamic models include dynamic linear and nonlinear system models.

Knowledge-based systems are based on the methods and techniques of artificial intelligence. The core components of these systems are the knowledge base and the inference mechanisms. Examples of knowledge-based systems are: rule-based systems, case-based reasoning systems, and graphical models (Luo, Tu, Pattipati, Qiao, & Chigusa, 2005). Examples of graphical models include: signed directed graphs, multi-signal flow graphs, Petri nets, and Bayesian networks (Luo et. al., 2006).

Conventional diagnostic techniques for a vehicle fuel system typically rely on knowledge of a prior failure condition. For example, when servicing a vehicle, the maintenance technician may determine that the fuel pump requires repair or replacement by direct testing and/or review of a recorded diagnostic trouble (error) code. This reactive diagnosis may not occur until vehicle performance has already been compromised. A proactive approach which tracks degradations in a fuel system is more advantageous than a reactive approach, particularly when used with emerging vehicle designs utilizing an ERFS.

In this paper, the fault detection and isolation problem of EFRS is characterized and some basic definitions are given. The main idea of fault diagnosis is to determine if there is any fault or abnormal behavior is present in the system, and to localize (isolate) the fault. In order to detect and localize the fault, a diagnosis system is needed. The diagnosis systems exploits the known signals, i.e. input signals such as control signals, and measured output signals from the system under diagnosis, to infer the fault.

The problem of fault diagnosis can be divided into several sub-problems. Here, we focus on three:

• *Fault Detection*: To determine if a fault is present in the system and usually the time when the fault has occurred.

• *Fault Isolation*: Determination of the location of the fault, i.e. which component or components have failed.

• *Fault Severity* (*Estimation*): Determination of the size and possibly time-varying behavior of a fault.

The three sub-problems are closely nested, and many algorithms cover several of them.

The focus of this paper is to develop data-driven fault isolation, and severity estimation algorithms based on neural network and statistical pattern recognition techniques exemplified by Support Vector Machines (SVM) (Vapnik, 1995), (Ge, Du, Zhang, & Xu, 2004), (Smola, Bartlett, Scholkopf, & Schuurmans, 2000), *k*-Nearest Neighbor (KNN), Principal Component Analysis (PCA) (Jackson, 1991), Partial Least Squares (PLS) (Bro, 1996), Gaussian Mixture Models (GMM), Discriminant Analysis, and so on (Bishop, 2006), (Duda, Hart, & Stork, 2001), and validate them based on fault injection in the HIL bench and the Chevrolet Silverado truck. We also estimate the severity of the isolated fault by PLS and principal component regression. The techniques chosen in the paper are based on popularity, range of complexity, robustness, data structure, and to assess the difficulty of the classification and regression problem.

The paper is organized as follows. Section 2 presents the overall framework for real-time fault detection and diagnosis of fuel delivery systems. Section 3 presents the neural network and statistical pattern recognition techniques. Section 4 presents the results of these classification and regression techniques for fault isolation and severity estimation on real data collected from the Chevrolet Silverado truck and the HIL rig. In section 5, we present the implementation of these data-driven techniques, embedded software in Simulink®, which can be used for real-time fault isolation and severity estimation. Finally, section 6 concludes the paper with summary and future research directions.

2. FRAMEWORK FOR REAL-TIME FAULT DETECTION, ISOLATION & SEVERITY ESTIMATION OF FUEL DELIVERY SYSTEMS

The Fault Detection and Diagnosis (FDD) process consists of an *offline* training phase and an *online* testing phase. Figure 1 depicts the block diagram of a real-time FDD scheme for the fuel delivery system.

During the steady-state detection, a *model-based detector* based on residuals, parity equations, regression, and parameter estimation techniques is implemented on the ECU, and detects the fault and estimates the state of health (SOH) during real-time operation of the vehicle. This model-based algorithm will be presented in a future paper. The nominal residuals for system operation are obtained during the offline phase via HIL rig experiments, and testing and validation is performed on the Chevrolet Silverado truck data collected at Milford Proving Grounds, and the faults detected in real-time conditions based on these nominal conditions.

In the *offline* phase, steady-state sensor data from different fault classes is used to train two fault classifiers, the Support Vector Machines (SVM), and the *k*-Nearest Neighbor (KNN). Partial Least Squares (PLS), and Principal Component Regression (PCR) estimators were also trained to assess the fault severities after fault isolation. The trained classifiers and their corresponding parameters and/or weights are exported to the online module for real-time FDD. An optimal sensor selection block is used to select the significant sensor suite for maximum diagnosability.

The *online* FDD phase consists of three steps: fault detection, fault isolation or classification, and fault identification or severity estimation. In the fault detection step, the steady-state model based detector analyses the residuals generated from the steady-state measurements of faulty and nominal systems. Upon detection of a fault, trained classifiers (SVM and KNN) are used for the online categorization of faults. In the next step, the PLS and PCA estimators corresponding to the isolated fault are used to determine its severity.

3. FAULT UNIVERSE

The fuel pump is an electronically controlled closed-loop system that maintains a desired fuel system pressure (~ 400 KPa for GMC 900 truck) and provides fuel flow on-demand to the engine under all operating conditions. The five critical fuel pump faults considered in this paper are listed in Tables 1 and 2. The faults in Table I correspond to those in GMT 900 truck and the faults in Table II are for the HIL Rig.

Altogether, fault injection experiments were performed with a commonly occurring motor/fuel pump fault, 2 sensor faults (pressure and current sensors), a pump module fault, and a fuel line fault. The fuel pressure and current sensors are located anywhere between the fuel pump and fuel rail, and the pressure and current sensor bias faults are often difficult to isolate, especially between each other, as current bias shows up as pressure bias and vice-versa. As the fuel pump degrades with age, the motor winding resistance increases and consequently, the pump PWM increases to supply the same desired pressure. A positive and negative pressure sensor bias results in the pump drawing less and more current respectively to compensate for the sensor errors. The Filter plugged fault is a result of the pump filter being blocked or clogged, and the effect of a leakage in the fuel line is represented by the fuel leakage fault.

The 2 faults in Table 1 were conducted at 2 different severity levels using a pressure and resistance box, respectively. However, since the HIL Rig allows for more flexibility, the winding fault was conducted at 10 different severity levels, and the pressure sensor bias fault, current sensor bias fault, filter plugged fault and fuel leakage faults were conducted at 4 severity levels as summarized in Table 3. The severity levels experiments of the winding fault, pressure and current sensor bias faults were conducted by adding resistances (resistance box), adding (positive) and subtracting (negative) pressure (pressure box), and injecting current, to the pump resistance, pressure, and current correspondingly. The filter plugged and fuel leakage fault experiments were conducted by restricting the fuel flow using valves (*flow restrictor* in Figure 2). The severity levels were chosen to represent the degradation of a fuel pump from low (operating normally) to high (end-of-life).

Table 1. Fault universe for GMT 900 truck

Fault	**Fault Type**	**Component**
F1.1	Pressure Sensor Bias (Slew 9.3 on pressure box)	Fuel Line
F1.2	Pressure Sensor Bias (Slew 9 on pressure box)	Fuel Line
F2.1	Winding Fault (1 Ω resistance added)	Pump/ Motor
F2.2	Winding Fault (0.66 Ω resistance added)	Pump/ Motor

Table 2. Fault universe for HIL rig

Fault	**Fault Type**	**Component**
F1	Winding/Commutator Fault	Pump/Motor
F2	Pressure Sensor Bias Fault	Fuel Line
F3	Current Sensor Bias Fault	Fuel Line
F4	Filter Plugged	Pump Module Fault
F5	Fuel Leakage Fault	Fuel Line

Table 3. Severity levels of each fault for HIL rig

Winding Fault	Pressure Sensor Bias Fault	Current Sensor Bias Fault	Filter Plugged	Fuel Leakage Fault

0.3158 Ω 0.4 Ω 0.5 Ω 0.66 Ω 0.75 Ω 1 Ω 1.2 Ω 1.5 Ω 2 Ω 3 Ω	50 KPa 100 KPa -50 KPa -100 KPa	1 A 1.5 A 2 A 3 A	25% closed 50% closed 80% closed 100% closed	25% closed 70% closed 80% closed 100% closed

4. Fuel Delivery System Hardware-in-the-loop Rig

A HIL system was designed as a means for validating the diagnostic algorithms, analyze the fuel system behavior under different operating conditions, and compare the physics-based system models to the actual system. The HIL rig was controlled by a lab machine and its performance parameters were linked to a user-interface (display screen) via CAN, to warn customers of likely vehicle failure/breakdown. A schematic of the GMT 900 based HIL rig is shown in Figure 2.

The fuel tank assembly houses two pumps, one for reference (for e.g. healthy pump) and the other (e.g. faulty pump) for applying different faults and subsequently, comparing the two pumps simultaneously under various diagnostic scenarios. Each pump has its own shut-off valve, when the other pump is in operation. The entire system has a control valve that enables fuel circulation in the loop, which comes in handy to study pump dynamics. Each pump is fitted with 2 thermocouples which act as temperature sensors for monitoring the temperature and providing warning in case of overheating.

Fault simulations were run on HIL Rig using a drive profile obtained from the GMT 900 test vehicle at *Milford Proving Grounds*. A Simulink®-dSpace model of the fuel system was used to extract the sensor and parameter identifier (PID) data (current, voltage, pressure, flow, and PWM) from the HIL Rig as shown in Figure 3. The desired engine speed and pressure profiles for the Milford Proving Ground (MPG) drive cycle are presented in Figures 4 and 5, respectively.

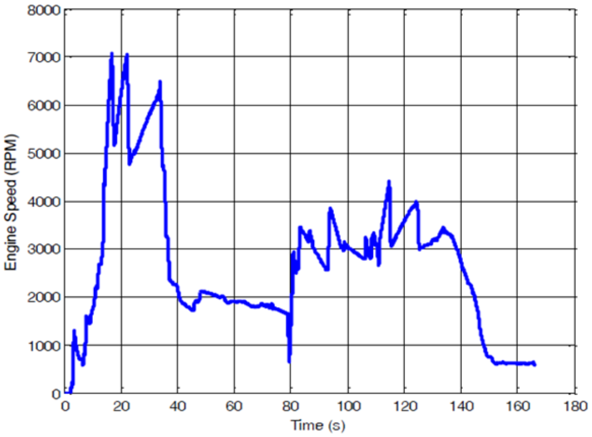


Figure 4. Desired engine speed for MPG drive cycle

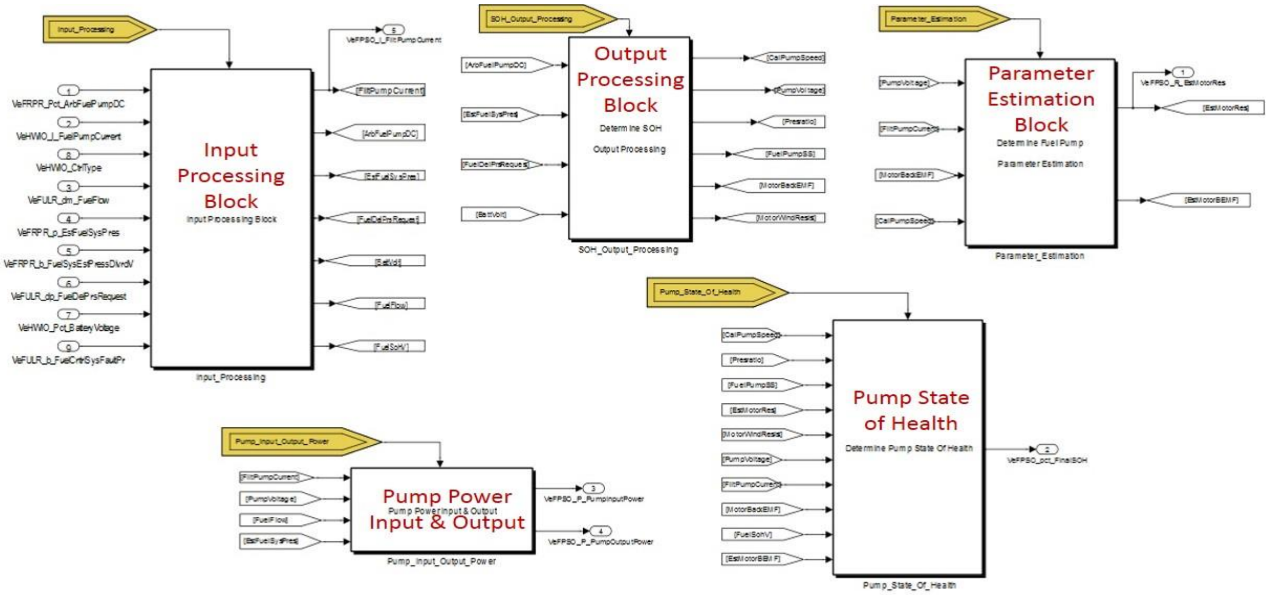


Figure 3. Simulink®-dSpace ERFS model

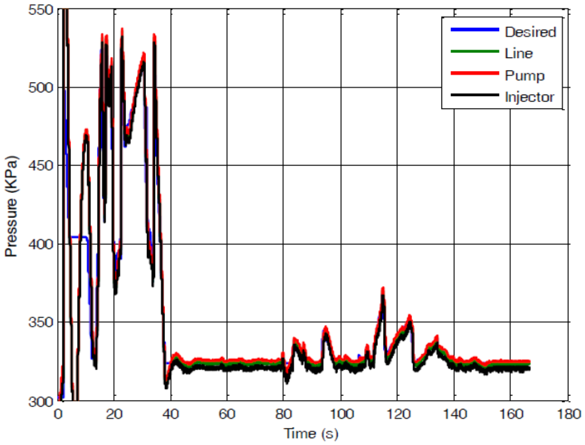


Figure 5. Desired pressure for MPG drive cycle

5. Fault Isolation & Severity Estimation of Fuel Delivery System

5.1 Fault Isolation on the GMT 900 Truck Data

Parameter identifier (PID) data was collected from a GMT 900 (Silverado) truck under *idle* and *normal* conditions by driving on Mound Road, Warren, Michigan.

The PIDs collected are listed below.

1. Current
2. Pressure
3. Flow
4. PWM
5. Current variance
6. Desired Pressure
7. Engine Speed
8. Vehicle Speed
9. Pump Pressure
10. Pump Voltage
11. Pump Efficiency
12. Status

The *features* used for fault isolation are presented in Table 4 below.

Table 4. Features for fault isolation

Power Out (Pressure x Flow x Pump Efficiency)	Power In (Voltage x Current x PWM)	PWM	Current	Flow	Pump Pressure

The fault universe, listed in Table 1, was used to define the *fault classes* for the classification algorithms as follows:

Class 1: No Fault

Class 2: Pressure sensor bias (*Slew9.3 on the pressure box*)

Class 3: Pressure sensor bias (*Slew9 on the pressure box*)

Class 4: Winding Fault (*1 ohm resistance added*)

Class 5: Winding Fault (*0.66 ohm resistance added*)

The classification results under the two driving conditions of the truck are presented in Tables 5 and 6.

Table 5. Classification/fault isolation accuracy (5x2 cross-validation) under *idle* conditions

Rank	Classifier	Accuracy
1	SVM	100%
2	QDA	100%
3	Fisher Discriminant Analysis 1. Linear 2. Quadratic	99.9813% 100%
4	GMM	99.6747%
5	PLS	92.3520%

Table 6. Classification/fault isolation accuracy (5x2 cross-validation) under *normal driving* conditions

Rank	Classifier	Accuracy
1	QDA	99.84%
2	Fisher Discriminant Analysis 1. Linear 2. Quadratic	98.9711% 99.8356%
3	SVM	98.2556%
4	GMM	95.1846%
5	PLS	81.2649%

The classification accuracies can be further improved using preprocessing techniques such as auto-scaling, mean-centering, PCA and PLS.

The classification task under idle conditions is much easier than the normal driving conditions. As seen in Table 5, both SVM and discriminant analysis (linear discriminant analysis and quadratic discriminant analysis) perform well while classifying faults under idling conditions (or steady-state operating conditions) of the truck. However, the classification is reasonably good even under normal operating conditions. SVM consistently performs well with *no false alarms* under both operating conditions of the truck and hence, was selected as one of the techniques for fault isolation in the data-driven software.

5.2 Fault Isolation and Severity Estimation on the HIL Rig Data

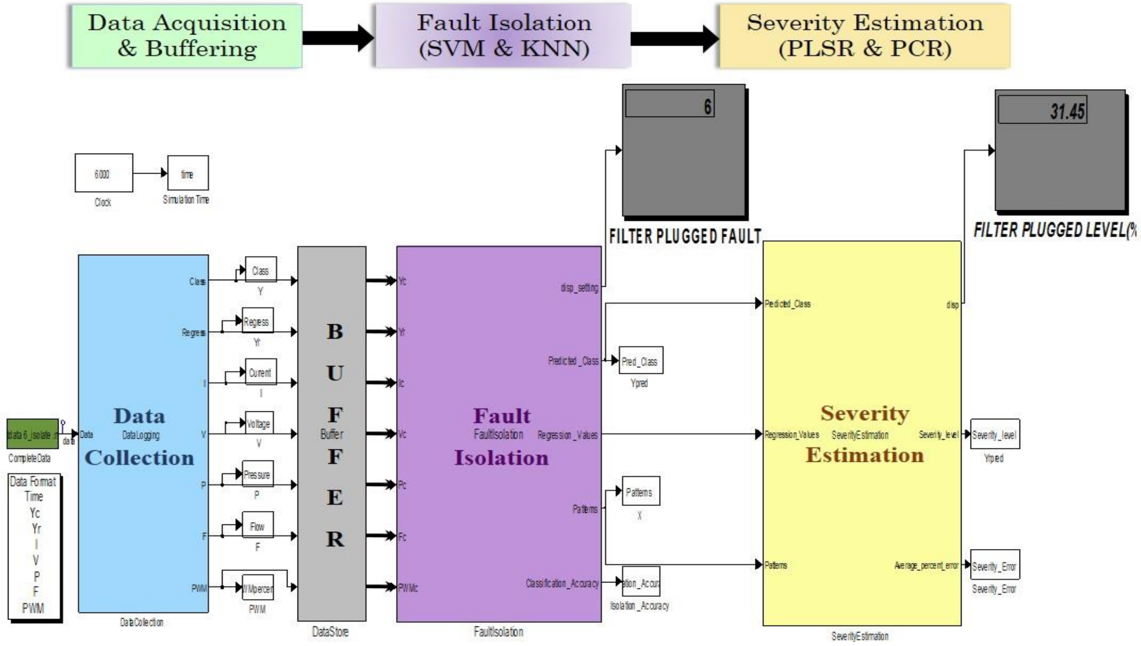


Figure 6. Data-driven fault isolation and severity estimation software

The PIDs listed below were directly used as features for the data-driven fault isolation and severity estimation.

1. Current
2. Voltage
3. Pressure
4. Flow
5. PWM

The fault classes used for isolation are as follows.

Class 1: No Fault

Class 2: Current Bias Fault

Class 3: Pressure Bias Fault

Class 4: Winding Resistance Fault

Class 5: Fuel Leak

Class 6: Filter Plugged

Table 7 presents the fault Isolation results for the HIL Rig. SVM, and KNN showed the highest accuracy of correct classification rate (> 99%). On the other hand, the Bayes and PLS classifiers showed the lowest accuracy.

Table 7. Classification/fault isolation accuracy (5x2 cross-validation)

Rank	Classifier	Correct Classification Rate (%)*	Overall False Alarm (%)
1	SVM	99.7028%	0.2972%
2	*k*-Nearest Neighbor 1. *k=1* 2. *k=2* 3. *k=3*	1. 99.5218% 2. 99.5218% 3. 99.4565%	1. 0.4782% 2. 0.4782% 3. 0.5435%
3	Discriminant Analysis 1. Linear 2. Diag Linear	1. 85.2393% 2. 81.3819%	1. 14.761% 2. 18.618%
4	Bayes Classifier with GMM Model	82.0410%	17.959%
5	PLS	81.2871%	18.713%

After a fault is detected and isolated, the severity estimation of the fault is needed in some cases. We used partial least squares regression (PLSR) and principal component regression (PCR) to estimate the severity of the isolated fault.

Simulations were run on the HIL Rig to collect data for each severity level. The Milford Proving Ground (MPG) drive cycle was run for each severity level of each failure model and PIDs were collected using the Simulink®-dSpace model of the ERFS system. Table 3 presented the different severity levels for each fault class.

The average percent error for each severity level is computed as follows:

$$\left|\frac{\text{Actual severity level} - \text{Average estimated severity level}}{\text{Actual severity level}}\right| \times 100 \qquad (1)$$

Tables 8 and 9 show the average percent error for each severity level for both PLSR and PCR.

Table 8. Average errors for each severity level for PLSR

Winding Fault	Pressure Bias	Current Bias	Filter Plugged	Fuel Leak
8.9049%				
47.17%				
5.441%				
7.5665%	0.9262%	8.9693%	2.50146%	0.2821%
5.224%	4.2832%	3.3192%	8.25643%	0.4683%
27.978%	2.04771%	0.7428%	9.3655%	0.0952%
6.645%	0.12258%	3.5819%	11.4154%	0.1155%
16.962%				
6.55%				
15.5783%				

Table 9: Average errors for each severity level for PCR

Winding Fault	Pressure Bias	Current Bias	Filter Plugged	Fuel Leak
9.4538%				
47.4694%				
5.8208%				
7.6480%	0.8987%	8.8356%	1.7446%	0.3537%
5.43909%	4.117%	3.2657%	8.7698%	0.4622%
28.0482%	2.338%	0.5541%	9.3293%	0.1133%
6.56998%	0.4139%	3.3568%	11.6222%	0.1119%
16.864%				
6.573%				
15.5545%				

The R^2 results are presented in Table 10. The fit accuracy doesn't provide as good an insight into the problem of severity estimation as the average percentage errors due to the fact that it looks for strictly the same value as the truth and provides a comparison between the true (Y) and estimated values ($\hat{Y}$).

$$R^2(\%) = \left(1 - \frac{\left\|Y - \hat{Y}\right\|_2^2}{\left\|Y - mean(Y)\right\|_2^2}\right) \times 100 \qquad (2)$$

Table 10. R^2 fit results for different regression methods

Faults	Regression Techniques	
	PLSR	PCR
Current Bias	92.6863%	92.6285%
Pressure Bias	98.1168%	98.1251%
Winding Fault	91.9874%	91.99%
Fuel Leak	99.9035%	99.9033%
Filter Plugged	89.4918%	89.6038%

The overall data-driven fault isolation and severity estimation software based on Figure 1 was implemented in Simulink®/MATLAB® environment using Embedded MATLAB® functions as shown in Figure 6. The Data Acquisition & Buffering Block simulates real-time data storage of the sensor and PID data (current, voltage, pressure, and flow). Once a preset number of samples (e.g. 1000) are stored in the database, the fault isolation block consisting of SVM and KNN is triggered. As soon as the fault is isolated, the severity estimation block consisting of regression techniques (PLS and PCA) are triggered, and the severity level of the fault is estimated. The parameters for SVM, KNN, PLS and PCA are obtained in the training phase offline.

6. Conclusions & Future Work

In this research, a data-driven fault detection and isolation (FDI) approach for automotive ERFS is presented based on data collected from a HIL fuel system rig and a GMT 900 truck. In the Silverado truck, three fault classes (No fault, pressure bias, and resistance faults) were introduced for classification under *idle* and *normal* driving conditions. Both SVM and QDA perform with accuracies greater than 98% while classifying faults under *idle* and *normal* driving condition conditions.

In the HIL rig, six fault classes (No fault, current bias, pressure bias, motor resistance, fuel leak, and fuel filter blocked faults) were introduced under a drive profile obtained from a GMT 900 test vehicle. SVM, and KNN showed the highest accuracy of correct classification rate (> 99%). On the other hand the quadratic classifier and the linear classifier showed the lowest accuracy.

Severity estimation levels for each fault using PLSR and PCR were performed using the data from the HIL with different severity levels. The results showed that fuel leak and pressure bias fault severity estimates have the highest accuracy, while the filter plugged fault severity estimate has the lowest accuracy.

The future work will involve the following steps:

1. Extensive real-time vehicle testing to validate the robustness of the data-driven fault isolation and severity estimation approach for the ERFS.

2. Develop remaining useful life (RUL) prediction strategies for the ERFS.

3. Condition-based Maintenance (CBM) of fuel system comprising of early fault diagnosis, isolation, and RUL based on system state awareness to optimally plan and execute preventive maintenance decisions for individual and fleet of vehicles.

ACKNOWLEDGEMENT

The work reported in this paper was supported by National science foundation (NSF) under grants ECCS-0931956 (NSF CPS) and ECCS-1001445 (NSF GOALI). We thank GM R&D and NSF for their support of this work. Any opinions expressed in this paper are solely those of the authors and do not represent those of the sponsor.

The authors would like to express their gratitude to Mehmet E. Inan for his help in data collection and analysis.

The authors thank the anonymous reviewers for valuable suggestions that enhanced the paper.

REFERENCES

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. New York, USA: Springer.

Bro, R. (1996). Multiway Calibration. Multilinear PLS. *Journal of Chemometrics*, vol.10, issue 1, pp. 47-61. doi: 10.1002/(SICI)1099-128X(199601)10:1<47::AID-CEM400>3.0.CO;2-C

Chiang, L. H., Russel, E., & Braatz, R. (2001). *Fault Detection and Diagnosis in Industrial Systems*. London, U.K.: Springer-Verlag.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification*. 2nd edition, New York, USA: John Wiley and Sons.

Ge, M., Du, R., Zhang, G., & Xu, Y. (2004). Fault diagnosis using support vector machine with an application in sheet metal stamping operations. *Mechanical Systems and Signal Processing*, vol. 18, pp.143-159. doi: 10.1016/S0888-3270(03)00071-2

Jackson, J. E. (1991). *A User's Guide to Principal Components*. New York, USA: John Wiley & Sons.

Luo, J., Tu, H., Pattipati, K., Qiao, L., & Chigusa, S. (2005). Graphical models for diagnosis knowledge representation and inference. *IEEE Autotestcon*, pp. 483-489. doi: 10.1109/MIM.2006.1664042

Luo, J., Tu, H., Pattipati, K., Qiao, L., & Chigusa, S. (2006). Diagnosis knowledge representation and inference. *IEEE Instrumentation and Measurement Magazine*, vol. 9, issue 4, pp. 45–52. doi: 10.1109/MIM.2006.1664042

Namburu, S. M., Azam, M. S., Luo, J., Choi, K., & Pattipati, K. R. (2007). Data-driven modeling, fault diagnosis, and optimal sensor selection for HVAC chillers. *IEEE transactions on automation science and engineering*, vol. 4, no. 3. doi: 10.1109/TASE.2006.888053

Smola, A. J., Bartlett, P. L., Scholkopf, B., & Schuurmans, D. (2000). *Advances in Large Margin Classifiers.* Cambridge, Massachusetts: The MIT Press.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York, USA: Springer.

Accommodating Repair Actions into Gas Turbine Prognostics

Zakwan Skaf[1], Martha A Zaidan[1], Robert F Harrison[1], and Andrew R Mills[1]

[1] *Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, S1 3JD. UK*
z.skaf@sheffield.ac.uk, m.a.zaidan@sheffield.ac.uk, r.f.harrison@sheffield.ac.uk, a.r.mills@sheffield.ac.uk

ABSTRACT

Elements of gas turbine degradation, such as compressor fouling, are recoverable through maintenance actions like compressor washing. These actions increase the usable engine life and optimise the performance of the gas turbine. However, these maintenance actions are performed by a separate organization to those undertaking fleet management operations, leading to significant uncertainty in the maintenance state of the asset. The uncertainty surrounding maintenance actions impacts prognostic efficacy. In this paper, we adopt Bayesian on-line change point detection to detect the compressor washing events. Then, the event detection information is used as an input to a prognostic algorithm, advising an update to the estimation of remaining useful life. To illustrate the capability of the approach, we demonstrated our on-line Bayesian change detection algorithms on synthetic and real aircraft engine service data, in order to identify the compressor washing events for a gas turbine and thus provide demonstrably improved prognosis.

1. INTRODUCTION

Gas turbine engines are subject to operational degradation which, over time, will reduce their performance. For effective fleet management, the ability to predict this degradation through prognostics is seen as a vital part of modern health monitoring. Prognostics enables forward predictions of the time to failure, thus offering a route to increase time in-service and reduced disruption for improved asset management. For accurate prognosis, knowledge of maintenance actions which affect the rate and state of degradation is of prime importance but is often difficult to obtain and incorporate.

Maintenance actions like compressor washing increase the usable engine life and the performance of the gas turbine. These actions are performed at geographically dispersed locations by organisations independent to those performing fleet management, which lead to uncertainty in the maintenance state of the asset. Organisational barriers do not permit the feedback of whether an advised maintenance action is taken or if maintenance is performed independent from fleet management advice. The uncertainty surrounding maintenance actions impact the ability to accurately trend and extrapolate the health degradation of a unit.

The solution proposed in this paper accurately detects maintenance events directly from the measured service data through a change detection algorithm. The event detection information is subsequently used as an input to a prognostic algorithm (Zaidan et al., 2013), advising the prognostic algorithm to update the estimation of remaining useful life.

2. LITERATURE REVIEW

Prognosis seeks to estimate the future health state of an asset and this problem has been addressed through a number of approaches, such as particle filters (Schwabacher & Goebel, 2007) and hidden Markov models (Tobon-Mejia et al., 2011), which can capture uncertainty in the projection of health state. Our work in Zaidan et al. (2013), provides a deterministic and efficient calculation which are then extended in this paper to accommodate maintenance events.

The problem of change point detection or detecting abrupt changes in time-series data has attracted a lot of research in the statistics and data mining communities over the last three decades (Basseville & Nikiforov, 1993; Brodsky & Darkhovsky, 1993; Gustafsson, 2000; Kawahara & Sugiyama, 2012). Change-point detection has been widely used in a range of real-world problems such as signal segmentation of a data stream (Tobon-Mejia et al., 2011), fraud detection in mobile networks (Bolton & Hand, 2002), climate change detection (Reeves et al., 2007), motion detection in vision systems (Ke et al., 2007), stock market prices (Chen & Gupta, 1997), nuclear engineering (Fearnhead & Clifford, 2003), and the aerospace domain (Fujimaki, 2005). These methods bring to bear a selection of models, statistical techniques and threshold selection policies to identify change events in data. It is proposed in this work that a unified approach can be provided with Bayesian change detection providing a rigorous means

Zakwan Skaf et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

to utilise a statistic data model and incorporate expectations about an impending change as a prior belief.

Generally, change-point detection methods can be classified into two categories depending on the time of detection: retrospective detection (batch processing), and on-line detection (sequential processing). A retrospective change point detection method waits until the end of a fixed period of time, and then uses all the data throughout the period of time to locate the change points. For example, if we are going to detect maintenance events with annual updating, we have to wait until the end of the year to collect all the engine service data before doing any analysis to locate temporarily the maintenance actions. Although retrospective change point detection requires longer reaction periods, it tends to give more robust and accurate detection (Kawahara & Sugiyama, 2012). On the other hand, on-line change point detection must detect change points as soon as possible, this inevitably means acting upon less information. To provide the most timely input into the prognostic algorithms, on-line methods are the focus of this paper.

Many of the previous Bayesian approaches to change point detection have been retrospective (Barry & Hartigan, 1993; Xuan & Murphy, 2007), and have demonstrated strong performance for off-line datasets but are not suitable for making instant decisions. A Bayesian on-line change point detection algorithm was recently introduced by Adams & MacKay (2007), and in an alternative formulation by Fearnhead & Liu (2007). While computational cost can be made to be approximately linear in Fearnhead & Liu (2007) by applying resampling strategies, a preferred recursive formulation by Adams & MacKay (2007) provides a closed form solution that is linear and introduces no approximation errors. This closed-form Bayesian algorithm, estimates the time since the last change point, which is called the run-length. Adams and MacKay used an underlying predictive model of the time series that is updated at each sample point, to estimate the probability of a new sample extending or resetting to zero the run-length. Recently, the algorithm has been implemented to automatic speech recognition systems to work in real-world environments (Chowdhury et al., 2012).

In this paper, we propose the integration of the on-line change point detection algorithm (Adams & MacKay, 2007), with a Bayesian-inference prognostic approach (Zaidan et al., 2013). The prognostic algorithm is updated as new data is received and outputs predictive probability distributions for the expected future health. The predictive distributions can be used in the detection algorithm to incorporate step change discovery into prognostic methodologies. The Bayesian methods employed effectively addresses the problems of novelty threshold selection; the incorporation of prior knowledge; and change detection with uncertain, noisy, and missing data.

To illustrate the capability of the approach, on-line Bayesian change detection algorithm will be implemented on real aircraft engine service data, in order to identify the compressor washing events of a population for gas turbines.

3. Mathematical Model

3.1. Prognostic Model

The true degradation is unknown and we choose to model a related health index of the asset which may be estimated from noisy data collected from the system. The health index estimate may in many systems be described as a probability distribution for a parametrically linear model which is projected forward in time to give an anticipated future health index, shown below in Equation (1):

$$\mathbf{x_{1:t}} = \phi(\mathbf{t})^T \mathbf{w} + \varepsilon \tag{1}$$

where ε is a random error term that follows a normal distribution $\varepsilon \sim \mathcal{N}\ (0, \sigma_n^2)$. In general, ϕ is a polynomial basis function, $\mathbf{w}$ is a vector of weights and $\mathbf{x_{1:t}}$ is the set of degradation measurements. Here we choose $\phi(\mathbf{t}) = (\mathbf{1}, \mathbf{t})^T$ as an affine function form with $\mathbf{t}$ denoting time, but of course variables other than time may be included such as variables related to usage.

Data measured from the environment is modelled as a normal distribution with mean $\phi(\mathbf{t})^T\mathbf{w}$ and variance σ^2. The likelihood distribution, $p(\mathbf{x_{1:t}}|\mathbf{t}, \mathbf{w}, \sigma^2) \sim \mathcal{N}(\phi(\mathbf{t})^T\mathbf{w}, \sigma^2 I)$, is used with a prior distribution to calculate the parameters of the posterior distribution. It is necessary to select an appropriate prior distribution of our data in order to obtain an analytically tractable posterior distribution, which is desirable for real-time, deterministic computation. We assume the prior distribution is a normal-inverse gamma (NIG) distribution and written as $p(\mathbf{w}, \sigma^2) \sim \text{NIG}(\mathbf{w}, V, a, b)$. The parameters for the prior distribution $(\mathbf{w}, V, a, b)$ can be built from an in-service database by way of ordinary least squares (OLS) estimation.

The posterior distribution for the model parameters, $p(\mathbf{w}, \sigma^2|\mathbf{x_{1:t}}, \mathbf{t}) \propto p(\mathbf{x_{1:t}}|\mathbf{t}, \mathbf{w}, \sigma^2)p(\mathbf{w}|\sigma^2)p(\sigma^2)$, are calculated based on a parametrised NIG distribution $(\text{NIG}(\mathbf{w}^*, V^*, a^*, b^*))$ as detailed in Zaidan et al. (2013).

The predictive distribution $(\pi_t = p(\mathbf{x_{t+1}}|\mathbf{x_t}))$, used to extrapolate for prognosis, can be used to evaluate the belief that a new data point belongs to the learnt mode by evaluating a predictive student-t distribution populated from the posterior updated model parameters. This distribution is constructed as $\text{St}(\phi(\mathbf{t}_*)^T\mathbf{w}^*, b^*(1+\phi(\mathbf{t}_*)^T V^* \phi(\mathbf{t}_*)), a^*)$, and may be used to calculate π_t for data point x_{t+1}.

3.2. Change-point Detection Technique

The detection of step change in engine performance data, is performed to identify the compressor washing events of a gas turbine using a Bayesian on-line change point detection approach. This approach is based on Bayes' theorem which allows us to make some inference for event E from observed data x. In other words, we can calculate the posterior probability $P\left(E|x\right)$ of E given x by using the Bayes' theorem:

$$p\left(E|x\right) \propto p\left(x|E\right) p\left(E\right) = \text{likelihood} \times \text{prior} \tag{2}$$

In this work, our goal is to partition the engine data into segments, which each show a natural decline in performance, separated by change events, where performance is recovered through a maintenance action. The delineations between segments are called the change points.

To determine these change points, we use the run-length method suggested by Adams & MacKay (2007), which is based on the Bayes' theorem. The data are independent and identically distributed (i.i.d) between change points, and the parameters are independent across the change points. The positions of change-points are not specified in advance but instead must be inferred from the data. The change point has occurred if the run-length, r_t, drops to zero; otherwise, the run-length increases by one ($r_t = r_{t-1} + 1$).

In this method, the predictions of the next data point should consider all possible run-lengths and weigh them by the probability of the run-length given the data. By finding the most probable run-length to be 0, i.e. an end to the current data segment, we find a change point. Notationally, we write x_t as the data at time t and $\mathbf{x_{1:t}}$ for the set of data $\{x_1, x_2, ..., x_t\}$, in addition, $\mathbf{x_t^{(r)}}$ is the set of most recent data corresponding to run-length r_t at time t.

The objective, for each time step t, is to estimate the run-length distribution $p\left(r_t|\mathbf{x_{1:t}}\right)$ over the collected data. By applying a confidence threshold to the run-length distribution, we can determine that the change point has occurred and then setting $r_t = 0$; or otherwise, conclude that it has not occurred and increment run-length as $r_t = r_{t-1} + 1$. The probability distribution for the run-length $p\left(r_t|\mathbf{x_{1:t}}\right)$ at time t can be estimated sequentially to predict the change point.

The run-length distribution $p\left(r_t|\mathbf{x_{1:t}}\right)$ can be computed as

$$p\left(r_t|\mathbf{x_{1:t}}\right) = \frac{p\left(r_t, \mathbf{x_{1:t}}\right)}{p\left(\mathbf{x_{1:t}}\right)} \tag{3}$$

with the probability of evidence calculable by marginalisation, $p\left(x_{1:t}\right) = \sum_{r_t} p\left(r_t, \mathbf{x_{1:t}}\right)$.

The recursion relation for $p\left(r_t, \mathbf{x_{1:t}}\right)$ can then be derived by writing as the marginal over r_{t-1}, and noting $\mathbf{x_{1:t}} = \{x_t, \mathbf{x_{1:t-1}}\}$:

$$p\left(r_t, \mathbf{x_{1:t}}\right) = \sum_{r_{t-1}} p\left(r_t, r_{t-1}, x_t, \mathbf{x_{1:t-1}}\right) \tag{4}$$

$$= \sum_{r_{t-1}} p\left(r_t, x_t|r_{t-1}, \mathbf{x_{1:t-1}}\right) p\left(r_{t-1}, \mathbf{x_{1:t-1}}\right) \tag{5}$$

$$= \sum_{r_{t-1}} p\left(r_t|r_{t-1}\right) p\left(x_t|r_{t-1}, \mathbf{x_t^{(r)}}\right) p\left(r_{t-1}, \mathbf{x_{1:t-1}}\right) \tag{6}$$

By exposing the previous time-step joint probability $p\left(r_{t-1}, \mathbf{x_{1:t-1}}\right)$, a sequential estimate is possible.

The prior belief of change, $p\left(r_t|r_{t-1}\right)$, only needs to consider two possible states – the run-length increases or resets to zero. By this binary condition, the method is made tractable. Consequently, the joint distribution of $p\left(r_t, \mathbf{x_{1:t}}\right)$ is computed for only these two cases: as a growth function when $r_t = r_{t-1} + 1$; or a change point function when $r_t = 0$.

The expression $p\left(x_t|r_{t-1}, \mathbf{x_t^{(r)}}\right)$ is the predictive distribution given the only the previous data points to build models. This is calculated by fitting probabilistic models to all possible r_{t-1} run-lengths of the data ($\mathbf{x_t^{(r)}}$) using the model shown in Equation (1), and assessing the probability of the data point at x_t given the predictive distribution for that model. The calculated predictive distributions, which we label as $\pi^{(\mathbf{r})}$, for the normally distributed data is calculated directly from the student-t distribution, as outlined in Section 3.1. The detection of change point enables the model parameters to be reset to some initial conditions.

Assuming that the prior probability of a change-point is given by the pre-specified hazard rate (H) (which, for simplicity, we assume to be independent of r_t), then

$$p\left(r_t|r_{t-1}\right) = \begin{cases} 1-H & if \quad r_t = \quad r_{t-1}+1 \\ H & if \quad r_t = \quad 0 \\ 0 & otherwise \end{cases} \tag{7}$$

In this work we tuned the value of H empirically (to 0.02), however the data could be used to select the value either by a priori learning from fleet data or using on-line techniques such as shown in Wilson et al. (2010), removing need for heuristic tuning.

The proposed on-line change point detection algorithm applied to prognostics is summarised as follows:

1. Learn priors for degradation model parameters (using OLS).
2. Initialise the run-length distribution $p\left(r_{t-1}\right) = 1$.
3. Observe new datum x_t.
4. Evaluate predictive probability using student-t distributions for all run-lengths $\pi_{\mathbf{t}}^{(\mathbf{r})}$
5. Evaluate the hazard function $H\left(r_t\right)$ (constant in this example)
6. Evaluate the growth probabilities

$p\left(r_t, x_{1:t}\right) = p\left(r_{t-1}, \mathbf{x_{1:t-1}}\right) \pi_{\mathbf{t}}^{(\mathbf{r})} (1 - H)$

7. Calculate the change point probabilities
$p\left(r_t = 0, x_{1:t}\right) = \sum_{r_t} p\left(r_{t-1}, \mathbf{x_{1:t-1}}\right) \pi_{\mathbf{t}}^{(\mathbf{r})} H$

8. Calculate the evidence
$p\left(x_{1:t}\right) = \sum_{r_{t-1}} p\left(r_t, \mathbf{x_{1:t}}\right)$

9. Determine the run-length distribution
$p\left(r_t | \mathbf{x_{1:t}}\right) = \frac{p(r_t, \mathbf{x_{1:t}})}{p(\mathbf{x_{1:t}})}$

10. Apply a threshold to the run-length distribution to determine if a change point has been detected. Reset the run-length as $r_t = 0$, and goto step 2; or, increment $r_t = r_{t-1} + 1$

11. Update the degradation model parameters distribution through the steps outlined in Section 3.1 to calculate the predictive $p(\mathbf{x_{t+1}} | r_{t-1}, \mathbf{x_t^{(r)}})$ for all possible run-lengths

12. Estimate the remaining useful life (RUL) by projecting forward the degradation model parameters using the prognostic model (Zaidan et al., 2013)

13. Return to Step 3.

4. Case Study

Gas turbine engines become fouled with airborne contaminants such as unburned fuel, oil, solids and pollen which encrust compressor components. Proper operation and maintenance can be used to minimize the fouling type losses. For example, compressor washing can be used as effective method to maintain the compressor efficiency and prevent significant fouling to occur. The washing of gas turbine compressors maximize the power output, and fuel efficiency, as well as increase the life time of the compressors components (Kurz & Brun, 2001; Malinge & Courtenay, 2007; GE, 2008).

As engine degradation happens, the engine efficiency will decrease. Consequently, the fuel consumption will increase to generate the required thrust. As a result the temperature of the engine will increase, and therefore the global health of the engines can be derived from the core flow temperature measured at the turbine exit (Marinai et al., 2003). The temperature is named either Exhaust Gas Temperature (EGT) or Turbine Gas Temperature (TGT). An estimate of the difference between the certified TGT maximum (redline) and a projection of TGT to full-rated take-off at reference conditions is named TGT margin (Malinge & Courtenay, 2007). The TGT margin is usually used to monitor the overall performance of the engine through Engine Health Monitoring (EHM) to detect the shifts performance for each engine, indicating the need for inspection/maintenance, and to forecast the remaining useful life of the engines.

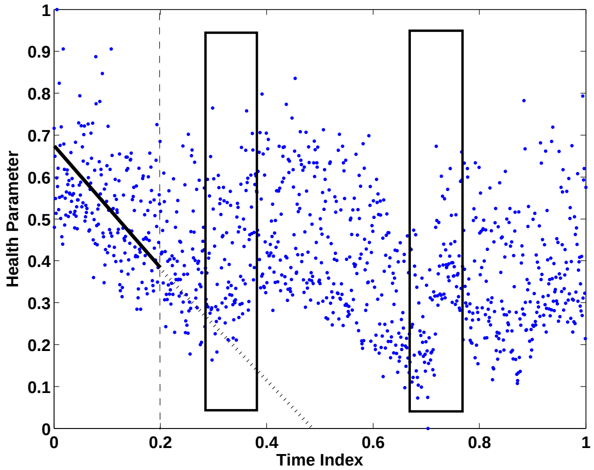


Figure 1. Example data of a single engine's measured value of TGT margin over a number of maintenance events, the time region of which are shown with black rectangles. The solid black line shows a regression fit of the linear model to the data received by time index 0.2, and the dotted line the projection of future values.

In general, the estimation of RUL is at the centre of system prognostics and health management. RUL gives operators estimation for decision making by quantifying how much time is left until functionality of engine is lost. RUL can be defined as the difference between present time and the time when the prediction of TGT margin crosses the zero TGT margin. Figure 1 shows an example of the TGT margin signal with compressor washing events. The measured value of TGT margin is presented as blue dots and suspected maintenance action (detected visually by step changes in the data) are highlighted to occur at some time within the rectangular region. The model presented in Equation 1 is fitted to the data received up until time index 0.2 (black line), from which point a projection is made (the techniques of prognosis are not the focus of this paper but those projections also incorporate uncertainty bounds not shown). It is clear that inclusion of the effects of compressor washing is needed for accurate estimation of RUL.

In the following section, we experimentally investigate the performance of the proposed algorithm using synthetic and real-world datasets.

4.1. Case study 1: Synthetic Dataset

In this first case study, the on-line Bayesian change point detection algorithm will be tested on synthetic data. Technique verification with synthetic data is an important method to evaluate the performance prior to testing to real-world dataset because the ground truth is available, in contrast to the real data which is subject to uncertainty in the true event times and the measured TGT margin. The synthetic data is generated to have similar noise and shape characteristics to the real data in Figure 1, and provides the true health index and change point times corrupted by the artificial noise.

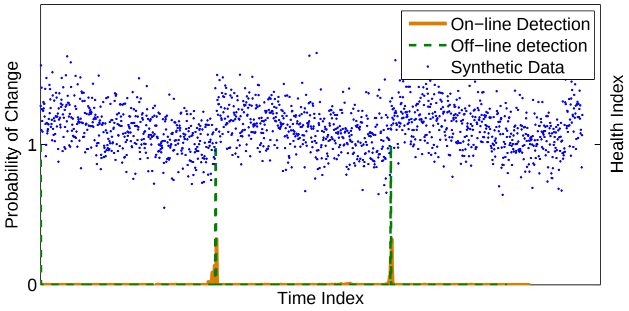


Figure 2. Synthetic Data: The graph represents the degradation signal (blue dots) (segmented by two change events) and the probabilities of zero run-length at each time instant for the on-line technique validated against an off-line technique.

The graph in Figure 2 shows an artificial time-series signal containing two change points, which can be visually identified at intervals spaced 30% along the time index axis. Superimposed on this figure are the probabilities of zero run-length at each time point calculated from two different change detection approaches. The on-line method described in Section 3.2 is validated against visual inspection and a widely cited off-line (retrospective) technique (Ruggieri, 2013). The on-line technique can clearly be thresholded to provide a change detection indication, however the probabilities of zero-length are lower in magnitude and time resolution compared to the off-line technique, though sufficient for our application this implies lower robustness.

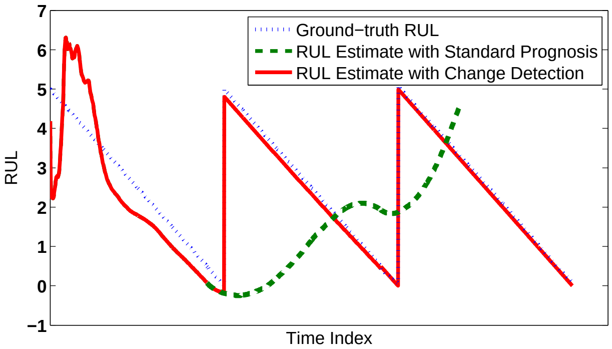


Figure 3. RUL estimation for synthetic data: The true RUL (dotted blue line), the estimated RUL (red line) with change detection, and the estimated RUL without change point detection (dashed green line) are shown.

A possible enhancement to decision making is to exploit the property of this on-line algorithm to calculate the probability of all run-lengths at each time step, not only zero-length. Because the confidence in a change detection increases as n more data points are collected, the most probable run-length after a true change time will occur at run-length n data points after the true change. A simple strategy is thus to observe the probability mass at low run-lengths (empirically it was found 0-5 points were sufficient, see Table **??**) and compare this to the probability of the run length increasing, this change in probability mass around an event is an effective measure of detection robustness. Enhancements can also be made computationally by strategies such as not carrying very low run-length probabilities to the next algorithm iteration, this is discussed in Turner et al. (2009).

Figure 3 shows the mean of the estimated RUL from the prognostic algorithm with (red line) and without (green dashed line) change detection. This estimate is made at every observation time over the asset life and compared against the known true RUL. It is clear how the estimated RUL increased after detecting a change point event through a reset of the prognostic algorithm, whereas with no change detection the estimate of RUL diverges to infinity. Due to the periodic nature of synthetic signal, setting the model priors to the posterior estimate before the change events means that the initial model learning shown at the start of the training period is avoided, this is only appropriate for perfect repair.

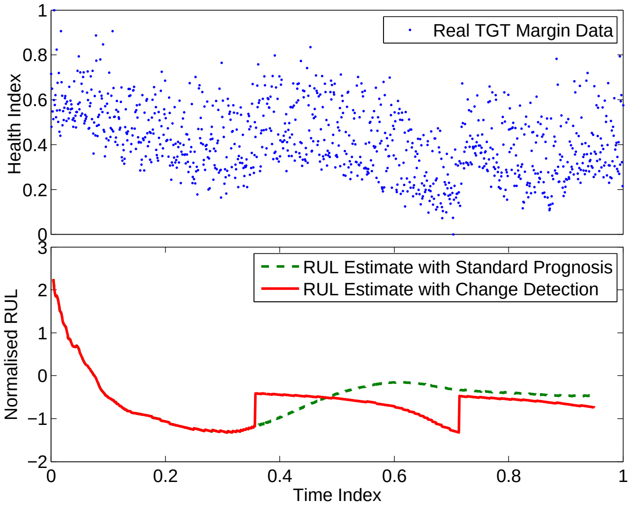


Figure 4. The TGT margin data is shown in the upper plot, with the mean RUL estimate for the prognostic algorithm with and without change detection in the lower plot

4.2. Case study 2: Real-World Dataset

Having validated the effectiveness of change detection and prognostic strategy, we apply the proposed method to real-world datasets, the results are shown in Figure 4. The x-axis is the time index which is the normalised number of flights, and the y-axis embodies the health index which is the normalised TGT margin, this is shown by the blue dots in the upper plot. The lower plot shows the mean RUL estimate for the prognostic algorithm with and without change detection. The step changes in RUL can be shown to visually coincide with significant events in the data at t=0.37 and 0.74, indicative of a compressor wash.

As with many prognostic applications with real data, the lack of ground true degradation for the real data makes an estimate of performance problematic. In addition, there is right censoring (asset removal) of the data before the crossing of the functional failure TGT margin threshold of zero degrees. To obtain an approximation of ground truth, the linear model was first trained on all available data and the zero margin crossing of this model used to generate the approximation to true RUL. The lack of ground truth also extends to the unavailability of true cause for the shift in TGT margin and the exact time.

Notwithstanding the difficulties in obtaining ground truth, some confidence can be developed by testing the change detection performance on synthetic data. Visual inspection of 50 sets of engine data were used to estimate the log ratio of step change magnitude (the signal of interest) to noise (SNR) for various suspected events. This ratio was found to be greater than one for the events in the data. By generating synthetic data with a set of noise characteristics and applying the change detection, Table **??** was created. As SNR decreases the change in probability mass for run-length around the change event decreased, the time accuracy can be measured by the sample interval width. For the highest SNR example, the probability mass shifted by 85 percent from growth to run-length reset over 2 samples (0.85 detection probability within 1 data point), whereas for at SNR of -0.2 only 5 percent change in probability occurred over ±3 samples. These quantified results, and observations from the real data, motivated the application to the full set of real engine data. In these tests, observed suspected changes were detected within a 5 observation interval at greater than 90% probability mass change, but the accuracy is difficult to quantify with precision since the ground truth information is not available.

Table 1. Change Detection Performance

	Probability Mass Change in sample interval		
SNR	±1	±3	±5
5.1	0.85	1.00	1.00
2.8	0.80	1.00	1.00
1.2	0.40	0.95	1.00
0.5	0.20	0.95	1.00
-0.2	0.00	0.05	0.40
-0.9	0.00	0.01	0.05

Despite this unavoidable limitation of the data, the application to the service data validates the principle of the approach, with its performance verified with synthetic data.

5. Conclusions and Future Prospects

Compressor washing increases the usable engine life and optimises the performance of the gas turbine. However, there are uncertainties about the timing and true effect of maintenance actions. These uncertainties surrounding maintenance actions impact prognostic efficacy because there is no information when the prognostic algorithm should be adjusted to accommodate performance changes arising from maintenance action. A Bayesian change point detection method was developed, to be illustrative of the possible prognostics fusion approach, in this paper to detect these maintenance events from the data.

The proposed method of on-line change point detection algorithm was implemented on an example of real aircraft engine service data, in order to identify the compressor washing events of a gas turbine and thus demonstrate the possibility of improved prognosis. Using synthetic data, the robustness of the approach was evaluated for both the detection and impact on underlying ground-truth prognosis. The event detection information was used as an input to a prognostic algorithm, advising the prognostic algorithm to update the estimation of remaining useful life.

In future research work, the following areas could be considered: the hazard rate for maintenance events could be learnt in advance from fleet datasets based on time and degradation, and exploiting this prior parameter to improve sensitivity; and issues regarding the computational efficiency of the change detection approach should be studied. In addition, while the change detection is effective at locating in time the change, there is further analysis needed to incorporate how to handle this event. The performance recovery is not perfect after each event and events later in the life of the turbine recover less performance, the data is to be mined to determine a distribution for expected recovery and this can be used to intelligently inform the reset of the prognostic estimation. The algorithms arising from this work are planned for integration into fleet management software to allow access to a vast array of data and thus facilitate robust testing.

Acknowledgements

This project is co-funded by the Technology Strategy Board, the Welsh Assembly Government and Scottish Enterprise.

References

Adams, R. P., & MacKay, D. J. (2007). Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.

Barry, D., & Hartigan, J. A. (1993). A bayesian analysis for change point problems. *Journal of the American Statistical*

Association, *88*(421), 309–319.

Basseville, M., & Nikiforov, I. V. (1993). *Detection of abrupt changes: theory and application*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, *17*, 2002.

Brodsky, B. E., & Darkhovsky, B. S. (1993). *Nonparametric Methods in Change-point Problems*. Dordrecht: Kluwer.

Chen, J., & Gupta, A. K. (1997). Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical Association*, *92*(438), 739 – 747.

Chowdhury, M. F., Selouani, S., & O'Shaughnessy, D. (2012). Bayesian on-line spectral change point detection: a soft computing approach for on-line asr. *Int. J. Speech Technol.*, *15*(1), 5–23.

Fearnhead, P., & Clifford, P. (2003). On-Line Inference for Hidden Markov Models via Particle Filters. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, *65*(4), 887–899.

Fearnhead, P., & Liu, Z. (2007). On-line inference for multiple change points problems. *Journal of the Royal Statistical Society B*, *69*, 589–605.

Fujimaki, R. (2005). An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceedings pakdd-2005: Ninth pacific-asia conference on knowledge discovery and data mining.* ACM Press.

GE. (2008). Flight Operations Newsletter. *GE Flight Operations Support*, *3*(1), 1-12.

Gustafsson, F. (2000). *Adaptive Filtering and Change Detection*. Wiley.

Kawahara, Y., & Sugiyama, M. (2012). Sequential change-point detection based on direct density-ratio estimation. *Stat. Anal. Data Min.*, *5*(2), 114–127.

Ke, Y., Sukthankar, R., & Hebert, M. (2007). Event detection in crowded videos. In *Ieee international conference on computer vision.*

Kurz, R., & Brun, K. (2001). Degradation in gas turbine systems. *Journal of Engineering for Gas Turbines and Power(Transactions of the ASME)*, *123*(1), 70–77.

Malinge, Y., & Courtenay, C. (2007). Safety first the airbus safety magazine.

Marinai, L., Singh, R., Curnock, B., & Probert, D. (2003). Detection and prediction of the performance deterioration of a turbofan engine..

Reeves, J., Chen, J., Wang, X. L., Lund, R., & QiQi, L. (2007). A review and comparison of changepoint detection techniques for climate data. *Journal of Applied Meteorology and Climatology*, *46*(6), 900–915.

Ruggieri, E. (2013). A Bayesian approach to detecting change points in climatic records. *Int. J. Climatol.*, *33*(2), 520–528.

Schwabacher, M., & Goebel, K. (2007). A survey of artificial intelligence for prognostics. In *Proceedings of aaai fall symposium.*

Tobon-Mejia, D., Medjaher, K., Zerhouni, N., & Tripot, G. (2011). Hidden markov models for failure diagnostic and prognostic. In *Prognostics and system health management conference (phm-shenzhen), 2011* (p. 1 -8).

Turner, R., Saatci, Y., & Rasmussen, C. E. (2009). Adaptive sequential Bayesian change point detection.

Wilson, R. C., Nassar, M. R., & Gold, J. I. (2010). Bayesian online learning of the hazard rate in change-point problems. *Neural computation*, *22*(9), 2452–2476.

Xuan, X., & Murphy, K. (2007). Modeling changing dependency structure in multivariate time series. In *Proceedings international conference in machine learning.*

Zaidan, M., Mills, A. R., & Harrison, R. (2013). Bayesian Framework for Aerospace Gas Turbine Engine Prognostics. In *Proceedings aerospace conference, 2013 ieee.*

BIOGRAPHIES

Zakwan Skaf received the B.S. degree from the faculty of Mechanical Engineering in Aleppo University, Aleppo, Syria, in 2001, and the MSc and Ph.D. degrees from the University of Manchester, Control Systems Centre, Manchester, U.K., in 2006 and 2011, respectively. He joined the Rolls-Royce University Technology Centre (UTC) at Sheffield University in 2012. His research interests are fault detection and diagnosis, fault tolerant control, non-linear systems, stochastic distribution control, and minimum entropy control.

Martha A Zaidan has joined the Rolls-Royce University Technology Centre (UTC) since 2010 as a recipient of Dorothy Hodgkin Postgraduate Award (DHPA) for his PhD research. His Research is focused on prognostic health monitoring using advanced data-driven modelling for aerospace gas turbine engine. He also holds MSc (Eng) degree (Distinction) in Control Systems from same department (2008-2009). During his Master degree and prior to joining the UTC, he conducted research in control engineering for applications in bioengineering and twin rotor dynamic. He earned BEng degree (Distinction) in electrical engineering from Trisakti University (2003-2007) in Indonesia. His main research interests are health monitoring technologies, intelligent control systems, reliability engineering and machine learning.

Robert F. Harrison, CEng received the B.Sc. and Ph.D. degrees in sound and vibration engineering in 1979 and 1983, respectively from the University of Southampton, England. He is currently Professor of Computational Data Modelling in the Department of Automatic Control and Systems Engineering at the University of Sheffield, England. His main research interests are in the development and application of machine learning and data modelling solutions for problems of practical interest in fields ranging from clinical and bio-medicine, communications, and transport. He is author of more than 70 journal articles and 150 conference papers in these and other areas.

Andrew R Mills, CEng is a Research Fellow at the University of Sheffield, from where he graduated with a MEng in Control System Engineering. He has worked in the defence industry on aerospace and automotive applications before his current role as Programme Manager and leading Health Management Technologies research within the University Technology Centre supported by Rolls-Royce. Research interests are in a broad range of aspects related to gas turbine system health management and controls.

System Level RUL Estimation for Multiple-Component Systems

João Paulo Pordeus Gomes[1], Leonardo Ramos Rodrigues[1], Roberto Kawakami Harrop Galvão[2] and Takashi Yoneyama[2]

[1] *EMBRAER S.A., São José dos Campos, São Paulo, 12227-901, Brazil*

joao.pordeus@embraer.com.br
leonardo.ramos@embraer.com.br

[2] *ITA – Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo, 12228-900, Brazil*

kawakami@ita.br
takashi@ita.br

ABSTRACT

Aircraft are highly valuable assets and large budgets are spent in predictive maintenance programs in order to maximize fleet availability. The application of PHM (Prognostics and Health Monitoring) technologies can be a powerful decision support tool to help maintenance planners. The estimated RUL (Remaining Useful Life) for each monitored component, obtained from a PHM system, can be used to plan in advance for the repair of components before a failure occurs. However, when system architecture is not taken into account, the use of PHM information may lead the operator to replace a component that would not immediately affect the availability of the system under consideration. In this paper, a methodology that combines fault tree information and individual components RUL estimations into a system level RUL (S-RUL) estimation is applied in a real life case study. The results showed that the methodology could have been successfully used in order to anticipate the failure of an aircraft ECS (Environmental Control System) and prevent an AOG (Aircraft on Ground) event from happening.

1. INTRODUCTION

PHM has been recognized by the members of the aeronautical sector such as aircraft operators, MRO (Maintenance, Repair and Overhaul) service providers and aircraft manufacturers as a technology that may lead to important competitive advantages such as reduction in operational cost and increase in fleet reliability (Rodrigues & Yoneyama, 2012).

The main goal of a PHM system is to estimate the remaining useful life (RUL) and the health state of components and systems. It comprises a set of techniques which use analysis of measurements to assess health state and predict impending failures of monitored equipments. Many works proposed PHM solutions for a high diversity of aeronautical components such as valves (Moreira & Nascimento Jr, 2012), pumps (Gomes, Leão, Vianna, Galvão & Yoneyama, 2012), engines (Babbar, Ortiz, Syrmos & Arita, 2009) and electronic components (Sandborn, 2005).

Methods for decision support using RUL estimations can be found in literature. Sandborn & Wilkinson (2007) and Rodrigues, Gomes, Bizarria, Galvão & Yoneyama (2010) presented examples of decision support methods using PHM information to improve maintenance planning. However, these works focused on the maintenance of one component, without considering that it is part of a system.

Modern aircraft are a good example of a complex system. They comprise multiple subsystems, each of them composed by multiple components. For safety analysis purposes, aircraft system architecture is often represented by a fault tree. When multiple components in a system are monitored by a PHM method, multiple RUL distributions are available for the decision maker. Although this seems to be positive, dealing with this amount of information could turn maintenance planning into a challenging task.

A possible solution for this problem is to calculate a system level RUL distribution based on the RUL distribution of each component. In this new framework, the decision maker does not have to deal with a set of component level RUL estimations. Instead, the S-RUL provides information related to the time when the whole system will stop working (i.e. when the combined failures of individual components will lead to a system failure).

In this work, a methodology to calculate the S-RUL distribution using component level RULs and system architecture information embedded in fault tree representation is applied. This methodology was proposed by Ferri et al. (2013). A case study is presented to illustrate

Gomes, J. P. P. et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the application of the methodology. In this case study, a subsystem of an aircraft ECS (Environmental Control System) is considered.

2. Fault Tree Representation

Fault Tree Analysis (FTA) is a failure analysis technique that, due to its ease of use and effectiveness in discovering and representing the interaction of component failures in a system, was widely adopted since the seventies in industries such as nuclear power generation, aviation and automotive (SAE, 1996).

During the FTA process, graphical diagrams called "fault trees" are produced in order to investigate what are the possible causes for a specific system failure, called the "top event". Fault trees represent sequences of events that may lead to the undesired top event under consideration. These sequences usually start from faults originated in system components, which combine with other component faults in order to cause failures that will propagate through the system.

The basic elements of a fault tree are the top event, the intermediate events and the basic events. Intermediate events represent failures propagated through the system and can be represented as a logical combination of basic events and other intermediate events. Basic events in a fault tree usually represent component faults. It is possible to attribute a probability of occurrence to each of the basic events in a given operating scenario. If the probabilities of all the basic events are known, it is possible to calculate the probability of the top event to occur using fault tree topology. Figure 1 shows an example of a simple fault tree.

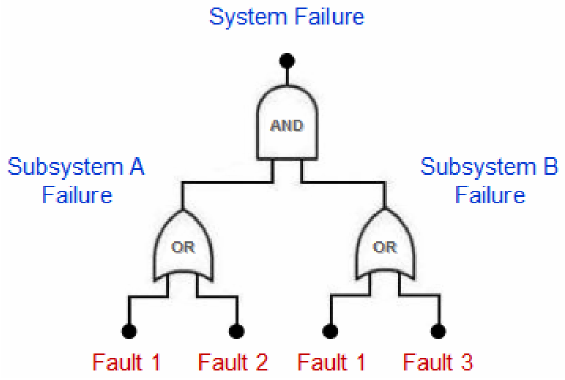


Figure 1. Fault tree example

Assuming that all basic events are independent, a convenient form of calculating the top event probability is by transforming the fault tree into its cut sets form. A cut set is a combination of basic events which, if they all occur simultaneously, will cause the occurrence of the top event.

In the cut sets form, each cut set is represented by an AND logical gate containing in its inputs all basic events forming the cut set under consideration. An OR logical gate is then used, and the output of each AND logical gate is connected to one of its inputs. Figure 2 shows the same fault tree as in Figure 1 transformed to its cut sets form representation. In this example, one cut set is composed by only one basic event (Fault 1). In such a situation, the AND logical gate can be omitted for this cut set and the basic event can be directly connected to the OR logical gate.

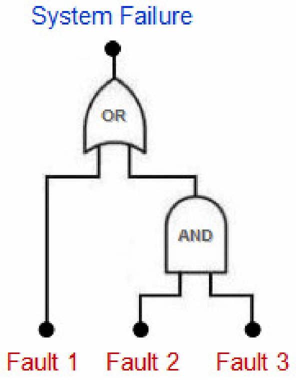


Figure 2. Fault tree in the cut sets form

Each input of the top OR gate is by itself a sufficient cause for the top event. The probability of the top event can then be obtained by calculating the union probability of all cut sets. On the other hand, if the basic events are mutually independent, the probability of each cut set can be obtained by calculating the joint probability of the basic events that compose the cut set. If all the basic events are mutually independent, the joint probability of a cut set is just the product of all its basic events.

3. System Level RUL

The System Level RUL (S-RUL) is calculated using the system architecture represented by the system fault tree and the RUL distributions for each component obtained from a PHM system. The procedure to calculate the S-RUL is summarized in Figure 3 (Ferri et al., 2013).

In step 1, the fault tree that represents the system under study is obtained. This information is commonly available for aircraft systems since fault trees are widely used in safety analysis. In step 2, system minimum cut sets representation is obtained based on the system fault tree. In step 3, the RUL estimation of each component is obtained from the PHM system. These estimations are commonly given as probability density functions. In step 4, the probability of each component to fail before instant k is calculated using the RUL predictions for each component.

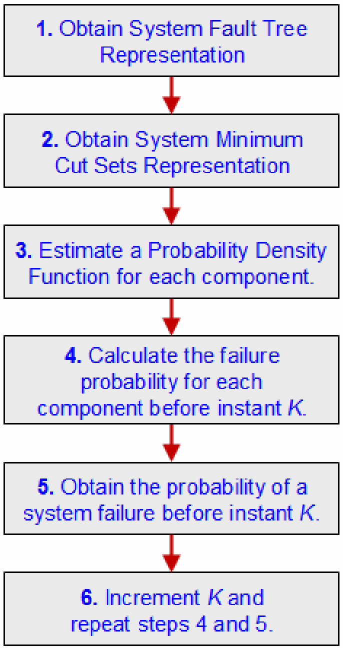

1. Obtain System Fault Tree Representation
2. Obtain System Minimum Cut Sets Representation
3. Estimate a Probability Density Function for each component.
4. Calculate the failure probability for each component before instant *K*.
5. Obtain the probability of a system failure before instant *K*.
6. Increment *K* and repeat steps 4 and 5.

Figure 3. SRUL calculation procedure

Using the minimal cut sets representation, the probability of each cut set to occur before instant *k* can be calculated by Eq. (1):

$$P(c_i)=\prod_{j=1}^{n}P(e_j) \qquad (1)$$

where $P(c_i)$ is the probability of the *i-th* cut set, $P(e_j)$ is the probability of the e_j basic event and n is the number of basic events in the *i-th* cut set. After calculating the probability of each cut set, the probability of the top event to occur before instant *k* can be calculated using Eq. (2):

$$P_T=1-\prod_{i=1}^{m}\left(1-P(c_i)\right) \qquad (2)$$

where P_T is the probability of the top event and m is the number of cut sets. It represents the probability of at least one cut set to occur, which is numerically equal to one minus the probability of no cut set to occur.

Steps 5 and 6 are repeated for subsequent instants. This procedure will result in a CDF (Cumulative Distribution Function) representing the probability of a system failure to occur over time.

4. Case Study

The system under study is a subsystem of an environmental control system (ECS) in an aircraft. This subsystem comprises two monitored components, a pressure control valve and a temperature control valve. A schematic view is presented in Figure 4.

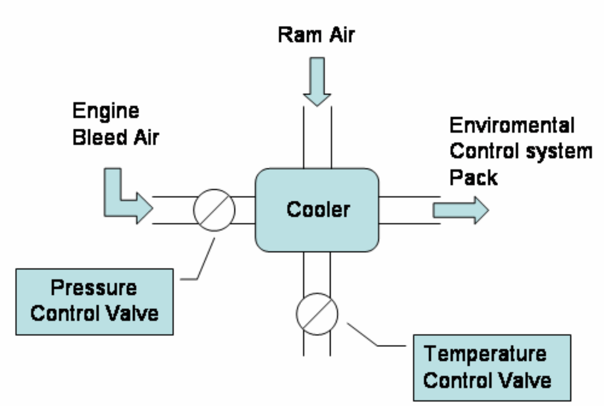


Figure 4. Aircraft environmental control system

The pressure control valve (PCV) is a pneumatic valve that regulates the engine bleed air flow so that a desired set point of pressure is reached. This air is sent to a heat exchanger and cooled until a temperature set point is achieved. The flow of ram air that passes through the cooler is controlled by the temperature control valve (TCV). This flow influences directly the heat exchanged between ram air and engine bleed air.

The aircraft used in this example has two of the subsystems presented (comprising PCV1, TCV1 and PCV2, TCV2), each of them located near one of the engines. For the correct operation of the ECS it is necessary that at least one of these subsystems is working properly and for that, both PCV and TCV need to be working.

The fault tree presented in Figure 5 is used to represent the ECS architecture.

The top event represented in the fault tree is related to the loss of the ECS. It is possible to see that this event only happens when both subsystems are defective. Each subsystem is considered defective if one of its components (PCV or TCV) is not working. The system in Figure 5 is presented in the minimal cut sets representation in Figure 6.

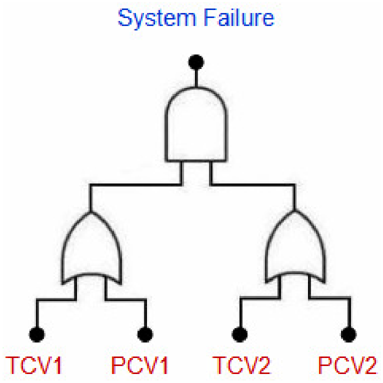


Figure 5. Environment system fault tree

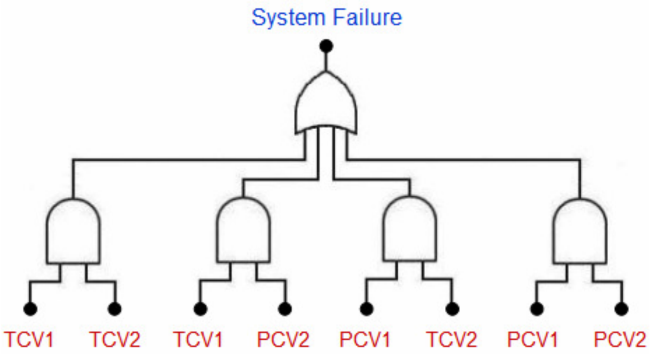


Figure 6. Minimum cut sets representation

4.1 PHM System

In the ECS system under consideration, the PCV and the TCV are the components with the highest failure rates observed on field, and this high number of failures leads to a high number of unscheduled component removals. For the purpose or reducing the number of unscheduled removals, PHM systems were developed for both the PCV and the TCV.

4.1.1. Pressure Control Valve

The pressure control valve is a pneumatic actuated valve. Its purpose is to keep the downstream pressure at a controlled set point value. The most common failure modes of this valve are related to wear of the spring or an increase in friction caused by the wear of the bearings. These failure modes affect the dynamic behavior of the valve. The performance of the pressure controller is also affected.

Field observations indicate that pressure signals exhibit variation in amplitude before a failure event. Such variations motivated the PHM methodology proposed in this work. Figure 7 shows an example of data collected from both a healthy valve and a degraded one. In Figure 7(A) a typical pressure signal for a healthy valve is shown. Figure 7(B) shows the pressure signal collected from a degraded valve just before a failure event.

The standard deviation of the pressure signal was then chosen as a degradation index (DI) for the PCV, as it can be related to a loss in regulation performance that may evolve to a failure, as described above. Figure 8 shows an example of how this degradation index changes over time. In this figure, the DI is normalized. Each cycle corresponds to one flight.

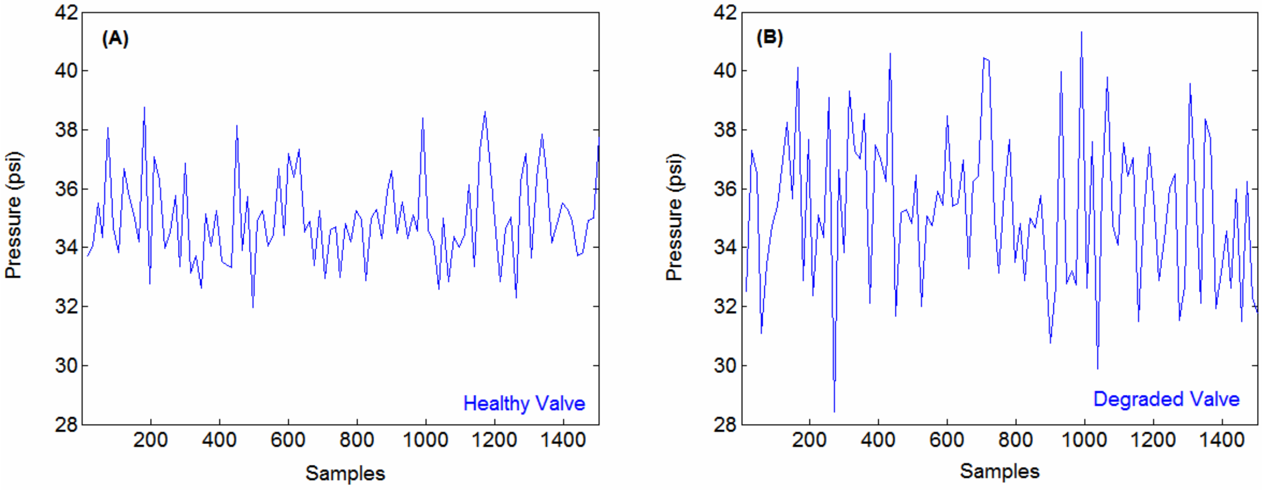


Figure 7. Pressure signals collected from a healthy and a degraded valve

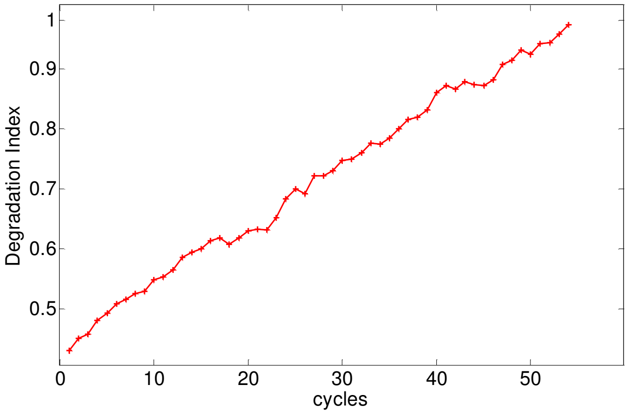


Figure 8. DI evolution for a pressure control valve

For failure prognostics implementation, a Kalman filter was employed. Concerning the dynamic model necessary for filtering and extrapolation, no first principles model was used. The state space representation of a linear degradation evolution with unknown slope was used for this purpose. This model was empirically chosen based on the aspect of the DI evolution. An example of this aspect can be observed in Figure 8. The slope and the degradation were estimated, resulting in the following model:

$$\begin{aligned} d_{k+1} &= a_k + d_k + v_k^1 \\ a_{k+1} &= a_k + v_k^2 \\ DI_k &= d_k + w_k \end{aligned} \tag{3}$$

where *d* is the estimated degradation, *a* is the slope, v^1, v^2 and *w* are gaussian noises, *DI* is the degradation index and *k* is the discrete time instant. In this case, *k* represents aircraft cycles. State noise v^1 and observation noise *w* represent, respectively, the actual state and the observation noises present in the data, while v^2 is an artificial noise added for the estimation of the fixed parameter *a*.

In the Kalman filter, the information concerning the variance of the parameter estimates at instant *k* is contained in the covariance matrix P_k. Using this information, the variance $\sigma^2_{v_k^2}$ can be obtained. An adaptive noise estimation procedure was used. This procedure is described in details in Leão (2011). In this procedure, $\sigma^2_{v_k^2}$ is calculated according to Eq. (4):

$$\sigma^2_{v_k^2} = \left(-1 + \frac{1}{\lambda}\right) P_k \tag{4}$$

where λ is a fixed positive value in the range [0.5 1).

Using the *d* and *a* distributions estimated at a given instant and the model presented in Eq. (3), Monte Carlo simulations were performed until *d* reaches a failure threshold. Failure thresholds were chosen according to the concept of Hazard Zone (HZ) (Orchard & Vachtsevanos, 2009).

The HZ defines a region, modeled by a bounded distribution, with high probability of failure occurrence. In this work, failure thresholds were sampled according to the chosen HZ distribution. The HZ was defined as a normal distribution with mean of 0.975 and standard deviation of 0.008. The HZ was defined using a set of run-to-failure DI series.

4.1.2. Temperature Control Valve

The temperature control valve is a pneumatic valve designed to control the air flow that passes through the cooler in order to control the temperature of the air sent to the ECS pack.

TCV failure reports showed that cabin temperature often presented variations few days before an event of failure. In an attempt to capture this behavior, the temperature standard deviation was chose as a DI. Figure 9 shows an example of the DI proposed. Each cycle corresponds to one flight.

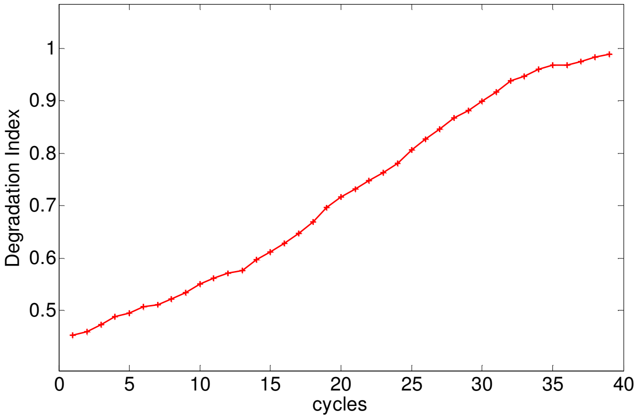


Figure 9. DI evolution for a temperature control valve

For degradation estimation and failure prognostics, a framework comprising a Kalman filter and a linear degradation progression model was used. This framework is similar to the presented for the PCV.

In this application, the HZ was defined as a normal distribution with mean of 0.99 and standard deviation of 0.0034.

4.2. Scenario Description and S-RUL Application

The scenario described in this section consists of the operation of a real aircraft. Although PCVs and TCVs were monitored for systems 1 and 2, no maintenance action was taken using this information. Figure 10 and Figure 11 show the degradation index progressions for PCVs and TCVs. The degradation index increases until a failure occurs at the last data point presented.

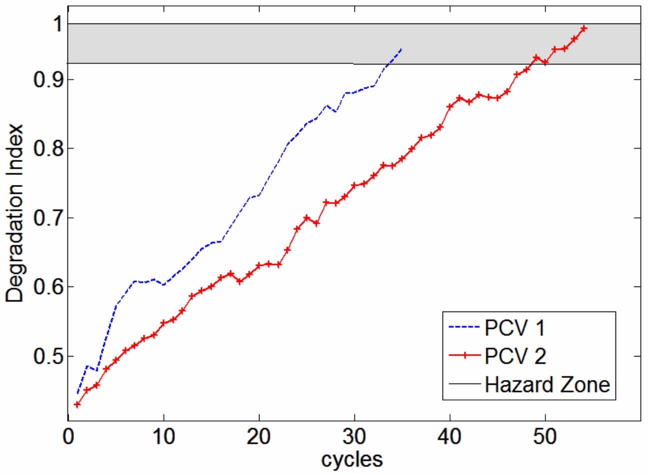


Figure 10. DI evolutions for two pressure control valves

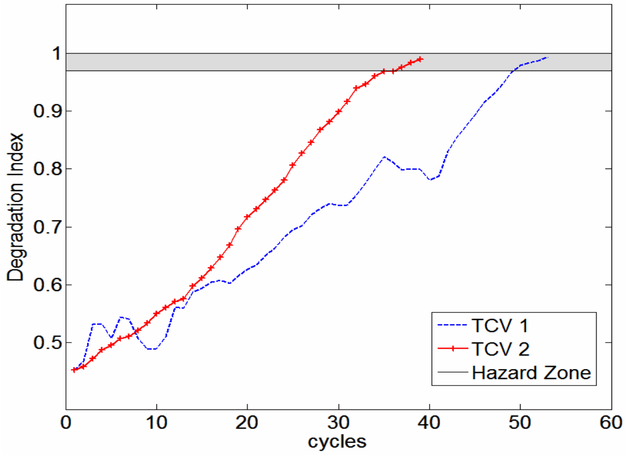


Figure 11. DI evolutions for two temperature control valves

The sequence of events and the consequences of each event of this real life example can be summarized as follows:

- On cycle 35, PCV 1 failed. Maintenance team removed the valve. The aircraft lost subsystem 1 but continued its normal operation.
- On cycle 39, TCV 2 failed. Maintenance team removed the valve. The aircraft lost subsystem 2. With both systems inoperative, the aircraft was grounded. Flights were delayed and the company had to rearrange other aircraft and passengers.
- After this event, both PCV 1 and TCV 2 were replaced and the aircraft continued its normal operation.
- On cycle 53 TCV 1 failed. Maintenance team removed the valve. The aircraft lost subsystem 1 but continued its normal operation.
- On cycle 54, PCV 2 failed. Maintenance team removed the valve. The aircraft lost subsystem 2. With both systems inoperative, the aircraft was grounded. Flights were delayed and the company had to rearrange other aircraft and passengers.

Analyzing the sequence of events presented and observing the degradation indexes in Figure 10 and Figure 11, it is possible to conclude that both AOG (aircraft on ground) events could be avoided by using the monitoring system. A prognostic system could be used to predict, with some degree of confidence, failure instants for all components thus allowing the maintenance plan to be modified to avoid the occurrences.

Although this seems to be a reasonable task, the workload for the decision maker could be reduced by using the concept of S-RUL. In the situation presented herein, the decision maker would have to analyze four RUL predictions (PCV 1, PCV 2, TCV 1 and TCV 2) to take the necessary actions. Using the concept of S-RUL, these four RUL estimations could be transformed in one S-RUL related to the remaining useful life of the whole environmental control system.

To illustrate this concept, consider a situation where the decision maker needs to analyze the data available up to cycle 25. Figure 12 and Figure 13 show, respectively, the RUL predictions for PCVs and TCVs at cycle 25.

The S-RUL was calculated following steps 1-6, presented in section 3. Figure 14 shows the S-RUL thus obtained.

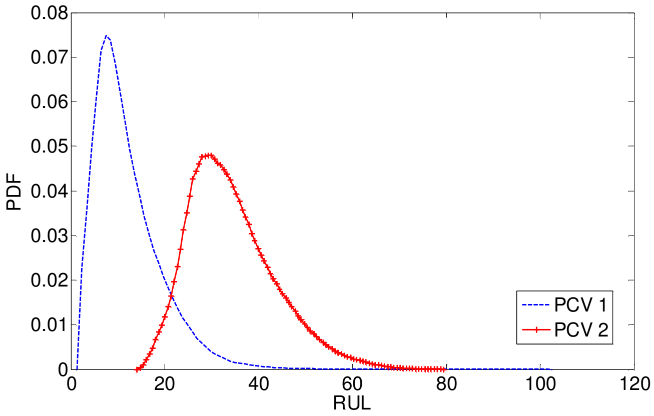


Figure 12. RUL estimations for the pressure control valves

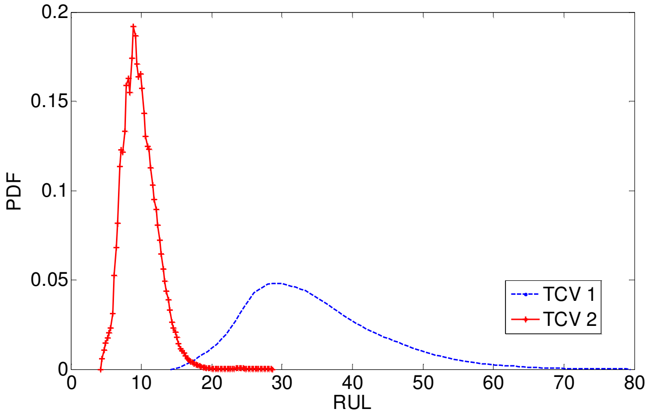


Figure 13. RUL estimations for the temperature control valves

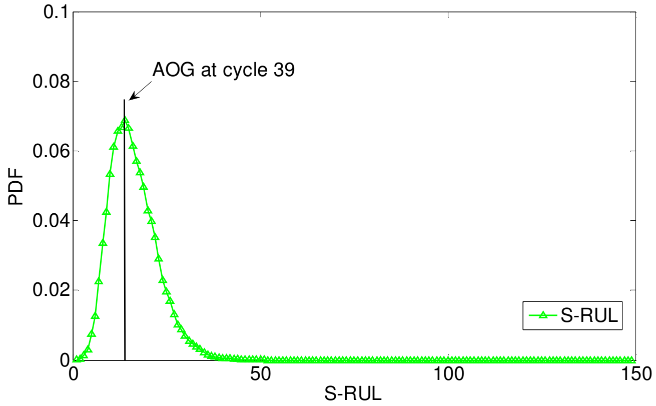


Figure 14. S-RUL estimation before any maintenance action

Observing Figure 14, it can be noticed that the first AOG could be predicted and a maintenance action could be planned. After replacing PCV 1 and TCV 2, the new S-RUL is presented in Figure 15.

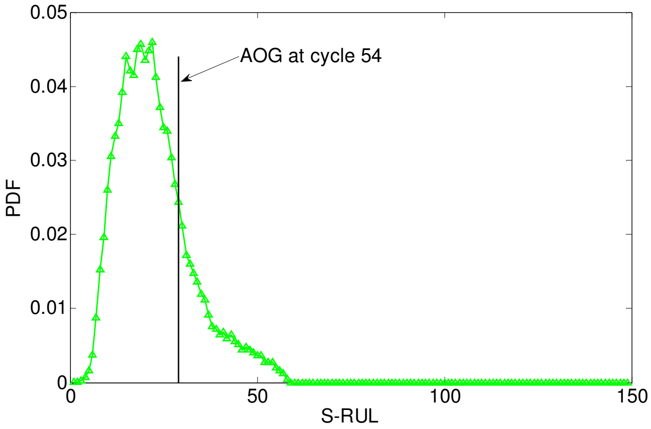


Figure 15. S-RUL estimation after replacing PCV 1 and TCV 2

Figure 15 shows that the second AOG could also be avoided. It is important no notice that the variance of the S-RUL presented in Figure 15 is greater than the variance of the S-RUL presented in Figure 14. This fact is explained by the fact that the second S-RUL prediction has a larger prognostic horizon, which leads to a greater uncertainty.

5. Conclusions

We found that the methodology discussed in this work could have been successfully used in a real life case study in order to estimate when a failure event would happen. This estimation could have been used to plan a maintenance intervention and prevent an AOG event from happening.

The methodology combines individual components RUL estimations into a single system level RUL (S-RUL) estimation. This characteristic becomes more relevant when the number of components within the system increases.

In complex systems, it may not be obvious to determine which component is the most critical for the system operability in a given scenario, even when RUL estimations for all components are available. The methodology discussed in this work is an alternative to evaluate the impact of each component in system operation by analyzing the changes in the S-RUL distribution.

The results presented in this paper were derived under the assumption that the failure events are independent. This assumption may not be realistic for some engineering systems. In many practical systems, the failure of one component may affect the condition or even cause a failure of other components. One relevant topic for future research is to consider the dependencies among all system components in the estimation of the system level RUL. Moreover, it would be of interest to investigate the computational complexity of the proposed method with respect to the number of components and the level of connectivity within the system.

Future research could also investigate how to adapt the methodology for the situation in which RUL estimations are not available for all components. Combining multiple top events in one single analysis can also be an interesting topic for further research.

ACKNOWLEDGMENT

The authors acknowledge the support of FAPESP (grant 2011/17610-0) and CNPq (research fellowships).

REFERENCES

Babbar, A., Ortiz, E. M., Syrmos, V. L. & Arita, M. M. (2009). Advanced Diagnostics and Prognostics for Engine Health Monitoring. In *Proceedings of IEEE Aerospace Conference*, Big Sky.

Ferri, F. A. S., Rodrigues, L. R., Gomes, J. P. P., Medeiros, I. P., Galvão, R. K. H. & Nascimento Jr, C. L. (2013). Combining PHM Information and System Architecture to Support Aircraft Maintenance Planning. In *IEEE International Systems Conference*, Orlando.

Gomes, J. P. P., Leão, B. P., Vianna, W. O. L., Galvão, R. K. H. & Yoneyama, T. (2012). Failure Prognostics of a Hydraulic Pump Using Kalman Filter. In *Proceedings of* the *Annual Conference of the Prognostics and Health Management Society*, Portland.

Leão, B. P. (2011). Failure Prognosis Methods and Offline Performance Evaluation. Ph.D. thesis. Instituto Tecnológico de Aeronáutica.

Moreira, R. P. & Nascimento Jr, C. L. (2012). Prognostics of Aircraft Bleed Valves Using a SVM Classification Algorithm, In *Proceedings of IEEE Aerospace Conference*, Big Sky.

Orchard, M. & Vachtsevanos, G. (2009). A Particle Filtering Approach for Online Fault Diagnosis and Failure Prognosis. Transactions of the Institute of Measurement and Control, no. 3-4, p. 221–246.

Rodrigues, L. R., Gomes, J. P. P., Bizarria, C. O., Galvão, R. K. H. & Yoneyama T. (2010). Using prognostic system forecasts and decision analysis techniques in aircraft maintenance cost-benefit models. In *Proceedings of IEEE Aerospace Conference*, Big Sky.

Rodrigues, L. R. & Yoneyama, T. (2012). Spare Parts Inventory Control for Non-Repairable Items Based on Prognostics and Health Monitoring Information, In *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, Portland.

SAE. (1996). ARP4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment.

Sandborn, P. (2005). A Decision Support Model for Determining the Applicability of Prognostic Health Management (PHM) Approaches to Electronic Systems, In *Proceedings of RAMS*.

Sandborn, P. A. & Wilkinson, C. (2007). A Maintenance Planning and Business Case Development Model for the Application of Prognostics and Health Management (PHM) to Electronic Systems. Microelectronics Reliability, Volume 47, Issue 12, pp. 1889–1901.

BIOGRAPHIES

João Paulo Pordeus Gomes holds a bachelor's degree on Electrical Engineering from Universidade Federal do Ceará (UFC, 2004), Brazil, master's (2006) degree on aeronautical Engineering and doctorate's (2011) degree in electronics engineering from Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, SP, Brazil. He is with EMBRAER S.A., São José dos Campos, SP, Brazil, since 2006. He works as a Development Engineer at EMBRAER focusing on PHM technology applications on aeronautical systems.

Leonardo Ramos Rodrigues holds a bachelor's degree in Electrical Engineering from Universidade Federal do Espírito Santo (UFES, 2003), Brazil, and a Master Degree in Aeronautical Engineering from Instituto Tecnológico de Aeronáutica (ITA, 2008), Brazil. He is currently pursuing his doctorate in Aeronautical Engineering at ITA. He is with EMBRAER S.A. since 2006. His current research interests are the application of health monitoring techniques for electronic components and the usage of PHM information for inventory optimization.

Roberto Kawakami Harrop Galvão holds a bachelor's degree on Electronic Engineering (Summa cum Laude, 1995) from Instituto Tecnológico de Aeronáutica (ITA), Brazil. He also obtained the master's (1997) and doctorate (1999) degrees in Systems and Control from the same institution. Since 1998 he has been with the Electronic Engineering Department of ITA as a full-time academic. Dr. Galvão is a Senior Member of the IEEE and an Associate Member of the Brazilian Academy of Sciences. He has published more than 200 papers in peer-reviewed journals and conferences. His main areas of interest are fault diagnosis and prognosis, wavelet theory and applications, and model predictive control.

Takashi Yoneyama is a Professor of Control Theory with the Electronic Engineering Department of ITA. He received the bachelor's degree in electronic engineering from Instituto Tecnológico de Aeronáutica (ITA), Brazil, the M.D. degree in medicine from Universidade de Taubaté, Brazil, and the Ph.D. degree in electrical engineering from the University of London, U.K. (1983). He has more than 250 published papers, has written four books, and has supervised more than 50 theses. His research is concerned mainly with stochastic optimal control theory. Prof. Yoneyama served as the President of the Brazilian Automatics Society in the period 2004-2006.

A Stochastic Modeling Approach of Quantized Systems with Application to Fault Detection and Isolation of an Automotive Electrical Power Generation Storage System

Sara Mohon[1] and Pierluigi Pisu[2]

[1,2]*Clemson University International Center for Automotive Research, Greenville, South Carolina, 29607, USA*

smohon@clemson.edu
pisup@clemson.edu

ABSTRACT

This paper introduces a stochastic modeling approach for a quantized system for the purpose of fault detection and isolation in an automotive alternator system. Three common alternator faults including belt slip, diode failure, and incorrect reference voltage for the voltage controller are considered and analyzed. A continuous nonlinear model of the alternator system is quantized into discrete states in order to simplify diagnostic efforts. The paper describes a stochastic modeling approach that creates a time-varying probability transition matrix that can be computed in real-time without the need for Monte Carlo simulation. Fault detection and isolation occurs through comparison of the most probable state from the transition matrix and the quantized output state.

1. INTRODUCTION

Today's vehicles require higher electrical demands than ever before due to more mandated safety technology and popular consumer technology integrated within the vehicle. The purpose of the vehicle's electrical power generation storage (EPGS) system is to maintain the necessary electrical power needed to start the vehicle and keep it running smoothly. A healthy EPGS system is crucial for proper operation of a vehicle.

Faults within the EPGS system do occur with age. Typical faults include belt slippage between the engine crankshaft and alternator pulley, failure of a diode in the bridge diode rectifier, and change in reference voltage of the voltage controller. These faults however can be detected and isolated with a carefully chosen diagnostic algorithm.

Diagnostics of the EPGS system is important for the vehicle owner and mechanic. Early diagnostics of a faulty EPGS system can warn the owner that the vehicle needs repair before more costly damage to other components occur. Early detection saves the owner further loss of time and money for repair. Furthermore, diagnostics stored in a vehicle's electronic control unit can be accessed by a mechanic to quickly and effectively determine the problem and steps needed to solve it.

Scacchioli, Rizzoni, and Pisu (2006) proposed a fault isolation approach for an EPGS system using two equivalent alternator models. One equivalent model for a healthy alternator and one equivalent model for an alternator with one broken diode. Parity equations and three residuals with constant thresholds were used for fault isolation. The approach assumed a 3000 second Federal Urban Driving Schedule (FUDS) cycle.

Zhang, Uliyar, Farfan-Ramos, Zhang, and Salman (2010) proposed a fault isolation approach for an EPGS system using parity relations trained by Principal Component Analysis (PCA). Three residuals with constant thresholds were used for isolation. The approach assumed a staircase profile for both load current and alternator speed input, which is not a realistic scenario.

Hashemi and Pisu (2011) proposed a fault isolation approach for an EPGS system using two observers and three residuals. The approach assumed a staircase profile for load current and a portion of the FUDS cycle for alternator speed. Adaptive thresholds were used for isolation. In other similar work, Hashemi and Pisu (2011) showed the same approach but created a reduced order adaptive threshold model using Gaussian fit of data. The second approach was less computationally intensive.

Scacchioli, Rizzoni, Salman, Onori, and Zhang (2013) proposed a fault isolation approach for an EPGS system using one equivalent EPGS model that used parity equations to produce three residuals for fault isolation. The approach used a staircase profile for both load current and alternator speed input.

Sara Mohon et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

As stated, previous work for fault isolation in an EPGS system has included observers and parity relations. The approaches with observers were built for linear systems that approximate the nonlinear behavior of the EPGS system. These approaches cannot be extended for direct use on the nonlinear system itself. At least three residuals are required for all previous approaches. It is also concerning that some approaches were not validated using real driving situations. Therefore these approaches have limited scopes.

In this paper, the EPGS system is modeled as a quantized system. The motivation for using a quantized system stems from the qualitative change in system behavior during EPGS faults and the need for a simpler real-time diagnostic algorithm. The approach in this paper uses a time-varying probability transition matrix and only one residual to detect and isolate faults. The approach requires much less real-time computational effort than previous works, which required at least 3 residuals. EPGS system data was created using a portion of the FUDS cycle to emulate a real-world situation. This approach is shown here in the context of an EPGS system but could be used for diagnostics in other systems as well including nonlinear systems.

The first section of this paper describes the EPGS model and an approximation of this model named the Equivalent EPGS model. The second section discusses three common faults in the model and how each affects the model output. The third section introduces the general concept of a stochastic model of a quantized system for the purpose of fault detection. The fourth section describes a new method to calculate the probability transition matrix for a quantized EPGS system. The last section provides simulation assumptions and results for each of the three faults in the EPGS system.

2. EPGS Model and Equivalent EPGS Model

This paper analyzes the EPGS system shown in Figure 1 as modeled by Scacchioli et al. (2006). It consists of a voltage controller, alternator, and battery. The controller can be an electronic control unit or a voltage controller on the alternator itself. In this paper, the controller is a part of the alternator to regulate field voltage. The alternator model consists of an AC synchronous generator, three phase full bridge diode rectifier, voltage controller, and excitation field.

The engine crankshaft mechanically spins the generator's rotor by use of a belt and pulley. The rotor is a ferrous metal wrapped with a single conductive winding. When the controller applies a small field voltage to the winding, a small field current flows through the winding. The flow of current through the winding produces a magnetic rotor with a north and south pole. However, the stator is composed of three phase stationary windings. As the magnetic rotor moves relative to the conductive stator windings, an electromotive force is induced in the stator windings. If the stator windings are connected to an electrical load, then AC current will flow in each of the three stator windings. The three currents are sent to a diode bridge rectifier to produce DC current for electrical loads or for recharging the battery. Therefore, the alternator takes mechanical energy of the engine and produces electrical energy for the battery or loads of the vehicle.

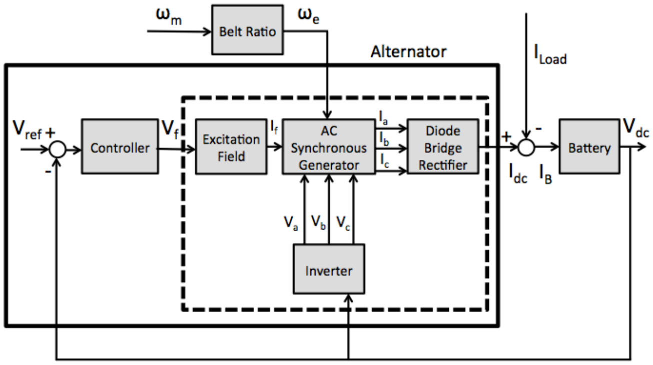


Figure 1. EPGS model

The model for the EPGS system results in a complex nonlinear system but can be more easily modeled by an equivalent DC electric machine as described by Sacchioli et al. (2006). The dashed line in Figure 1 encompasses the components represented by the DC model. This approximation gives the equivalent EPGS model shown in Figure 2.

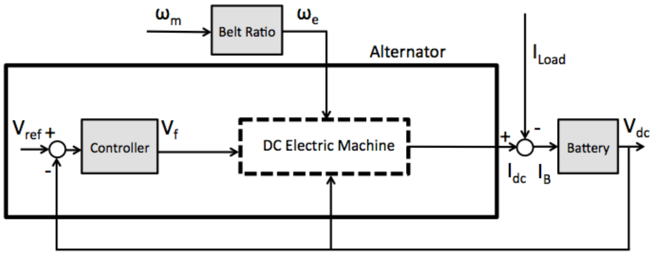


Figure 2. Equivalent EPGS model

The DC electric machine is modeled by the state space system in Eq. (1) as shown by Hashemi (2011).

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & a_{12}(\omega_e) \\ 1 & a_{22}(\omega_e) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} b_{11}(\omega_e) & b_{12}(\omega_e) & b_{13}(\omega_e) \\ 0 & b_{22}(\omega_e) & b_{23}(\omega_e) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{1}$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

Equation (1) has two states z_1 and z_2 and inputs u_1, u_2, and u_3. The system inputs represent the alternator field voltage V_f, angular frequency of alternator ω_e, and dc voltage of the battery V_{dc} also shown in Eq. (2). The coefficients a_{12}, a_{22} and $b_{11}…b_{23}$ are functions of engine speed and were found using system identification by Hashemi (2011) using test data at different constant engine speeds. In this model, state z_2 is the measurable quantity I_{dc} which is the rectified output current of the alternator.

$$\begin{aligned} y_2 &= I_{dc} = z_2 \\ u_1 &= V_f \\ u_2 &= \omega_e \\ u_3 &= V_{dc} \end{aligned} \tag{2}$$

3. Possible Faults in EPGS System

The EPGS system is important in every vehicle and faults in the system need to be detected and isolated as quickly as possible to prevent costlier damage. This paper considers three common faults that occur in an EPGS system. Possible fault locations in EPGS system are bolded in Figure 3.

1. *Voltage controller fault*: This fault occurs when the reference voltage V_{ref} is incorrectly raised or lowered by a percentage of the nominal V_{ref}. The fault can cause the alternator to overcharge or undercharge the battery.
2. *Open diode rectifier fault*. This fault occurs when a diode in the diode bridge rectifier breaks. The fault results in a large ripple in battery voltage V_{dc} and alternator output current I_{dc} thereby decreasing the efficiency of alternator output.
3. *Belt slip fault*. This input fault occurs when the belt between the engine crankshaft and alternator pulley slips due to insufficient tension. The belt slip causes a decrease in alternator rotational speed ω_e and a decrease in alternator output voltage. To compensate, the voltage controller increases the field voltage and/or the battery must discharge more often to meet load demand. This can age the battery prematurely. Belt slip can signify the belt is worn and needs to be replaced.

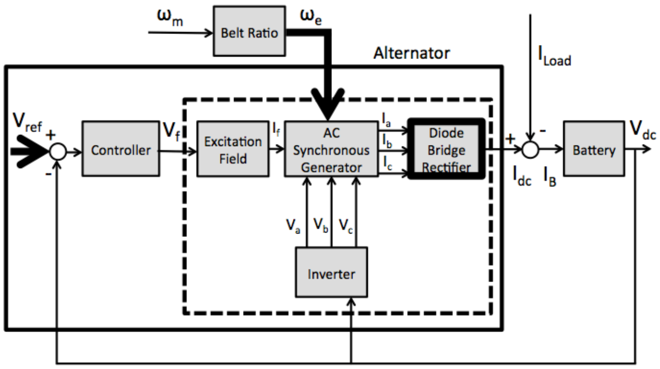


Figure 3. Possible faults in EPGS model

4. Stochastic Model for Quantized System

Equation 1 gives the continuous model of the DC electric machine. A continuous system provides much more information than a discrete system for a diagnostic algorithm to sift through to find a fault. If a continuous system could be simplified in a discrete manner without significant loss of information, the diagnostic algorithm will have a simpler task when searching for a fault.

Suppose we diagnose a continuous system as a quantized system shown in Figure 4 (Blanke, Kinnaert, Lunze, and Staroswiecki, 2006). Input u(t) is the continuous input at time t, f(t) is the amount of fault at time t, and y(t) is the output at time t. The output y(t) passes through a quantizer and produces [y(t)]. The real valued signal y(t) is assigned a new name [y(t)] that corresponds to an interval of real valued signals given a set of real valued intervals. For example, if y(t=10)=70 and the interval of real values 50 through 75 is assigned the name [y(t=10)]=2, then y(t=10) will be assigned the quantized name of 2.

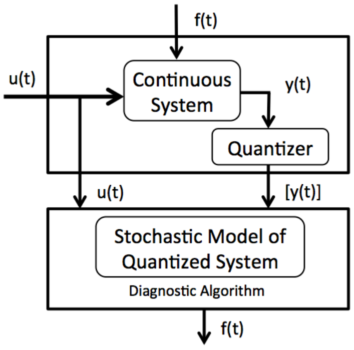


Figure 4. Diagnostics of quantized system

Assuming output y(t) contains two observable outputs $y_1(t)$ and $y_2(t)$ that can be quantized, a plot of y_2 vs y_1 can be visualized as a set of rectangles as shown in Figure 5. Output $y_1(t)$ can be quantized into intervals a-e and output $y_2(t)$ can be quantized into intervals 1-5. The grey section represents the system space. As input u(t) continuously changes over time, the discrete output [y(t)] will transition from one grey rectangle to another. Some rectangles will be more favorable for a system to move into for a given input and a current rectangle. The unfavorable rectangle transitions can be interpreted as having a low probability of occurrence. Transition probabilities from one rectangle to another could be arranged in a probability transition matrix and could be very useful for diagnostics. If a transition occurs that has very low probability, then a fault may be present.

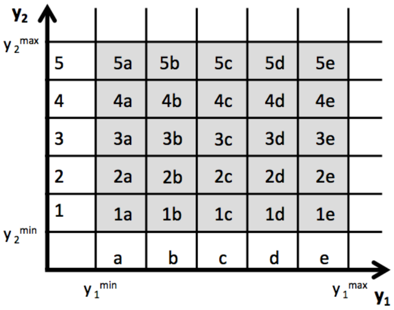


Figure 5. Example of quantizing two outputs

A typical approach to obtain a static probability transition matrix entails using a healthy model of the plant and the use of Monte Carlo simulation as shown by Alam (1995). In this paper, a new method for obtaining a time-varying probability transition matrix while monitoring the data for faults is introduced in the next section in application to the DC electric machine.

5. New Method for Probability Transition Matrix

Using Eq. (1) the DC electric machine system can be viewed as a 2D space with z_1 and z_2 axes. The system output will be contained on this plane. Since z_2 is the only output, the user knows where the output is in relation to the z_2 axis. The user does not know where the output is in relation to the z_1 axis except that it must exist between some minimum value z_1^{min} and maximum value z_1^{max} shown in Figure 6.

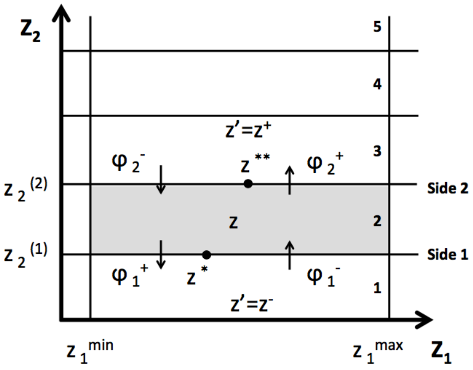


Figure 6. Graph of quantized DC electric machine system with flow definitions

For the purpose of diagnostics, the system space is divided into quantized states across the z_2 axis and assigned names such as 1, 2, 3, etc. as shown on right hand side in Figure 6. In this paper, the user assumes that with each event, in this case one time step, the current state z can only transition up or down to an adjacent state z' or remain in the same state z. The state may not jump over other states to nonadjacent states. The state may not move left or right outside of the z_1 boundaries. The selection of state boundaries will depend on the system being investigated to ensure only adjacent states are used by the system in healthy conditions.

The objective is to calculate the probability of transitioning out of current state z and the probability of future state z' = z. The probabilities are calculated using a two-dimensional form of the divergence theorem. The three-dimensional form of the divergence theorem is defined in Eq. (3). We define V as a closed volume, A as the surface area of V, $\bar{n}$ as the outward pointing normal vector of the closed volume V, and $\bar{F}$ as a continuously differentiable vector field in volume V. A picture for a cubic volume is shown in Figure 7.

$$\iiint_V \left(\nabla \cdot \vec{F}\right) dV = \iint_A \left(\vec{F} \cdot \vec{n}\right) dA \qquad (3)$$

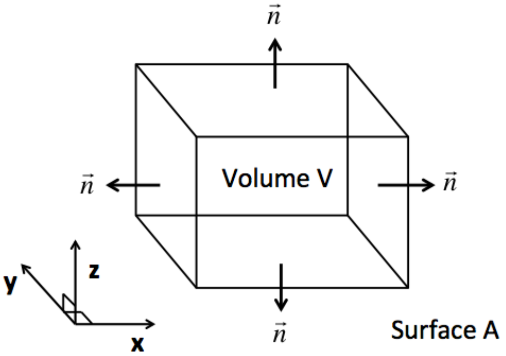


Figure 7. Graph of 3D Divergence Theorem

For the alternator problem, one can imagine multiple cubes stacked in the z direction and then collapsing the picture to only contain the x-z plane. This yields the 2D space with desired upward and downward flow consistent with the alternator problem in Figure 6.

A two-dimensional form of the divergence theorem is defined in Eq. (4). We define C as a closed curve, A as the 2D region in the plane enclosed by C, $\bar{n}$ as the outward pointing normal vector of the closed curve C, and $\bar{F}$ as a continuously differentiable vector field in region A. A graph of the 2D divergence theorem for the alternator problem is shown in Figure 8.

$$\iint_A \left(\nabla\cdot\vec{F}\right)dA=\int_C\left(\vec{F}\cdot\vec{n}\right)dr \tag{4}$$

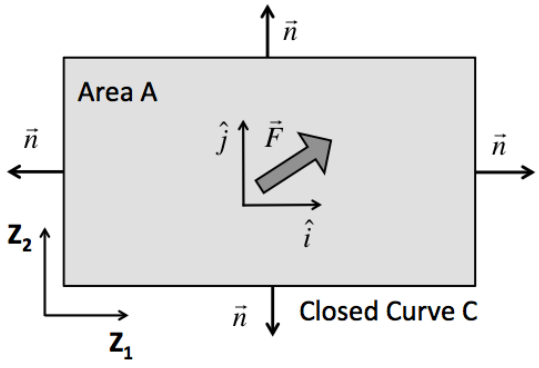


Figure 8. Graph of 2D Divergence Theorem for state z in DC electric machine state space

We consider that the vector field $\bar{F}$ describes transition flow in and out of the current state along the state boundaries. For the DC electric machine model, $\bar{F}$ is defined as Eq. (5) where $\hat{\imath}$ and $\hat{\jmath}$ are coordinates of vector field F and functions f_1 and f_2 are defined by states z_1 and z_2 from the state space model in Eq. (1).

$$\begin{aligned}\vec{F}&=f_1\,\hat{i}+f_2\,\hat{j}\\ \dot{z}_1&=f_1(z_1,z_2,u_1,u_2,u_3)\\ \dot{z}_2&=f_2(z_1,z_2,u_1,u_2,u_3)\end{aligned} \tag{5}$$

The flow through the left and right sides of the area A in Figure 8 will be assumed zero for the alternator problem. The line integrals along the state boundaries shown in Figure 6 will determine flow in and out of the state. Flow out of state z is defined as a positive value φ^+ and flow into state z is a negative value φ^-. Since each side may have flow in and flow out sections, the flow transition point z** or z* is found if necessary and the appropriate limits of integration for flow in and flow out are integrated for each side. Transition points are shown in Figure 6. Without loss of generality assume $f_2 < 0$ if $z_1 < z^*,z^{**}$ and $f_2 > 0$ if $z_1 > z^*,z^{**}$ such that Eq. (6) holds. The upward and downward flow through each side of state z is given by Eq. (7).

$$\begin{aligned}&f_2(z^*,z_2^{(1)},u_1,u_2,u_3)=0\\ &f_2(z^{**},z_2^{(2)},u_1,u_2,u_3)=0\end{aligned} \tag{6}$$

$$\begin{aligned}\varphi_1^+&=-\int_{z_1^{\min}}^{z_1^*} f_2(z_1,z_2^{(1)},u_1,u_2,u_3)\,dz_1 \quad >0\\ \varphi_1^-&=-\int_{z_1^*}^{z_1^{\max}} f_2(z_1,z_2^{(1)},u_1,u_2,u_3)\,dz_1 \quad <0\\ \varphi_2^-&=\int_{z_1^{\min}}^{z_1^{**}} f_2(z_1,z_2^{(2)},u_1,u_2,u_3)\,dz_1 \quad <0\\ \varphi_2^+&=\int_{z_1^{**}}^{z_1^{\max}} f_2(z_1,z_2^{(2)},u_1,u_2,u_3)\,dz_1 \quad >0\end{aligned} \tag{7}$$

Next we define φ_{in}, φ_{out}, and φ_{total} in Eq. (8) in order to build probabilities. The sum of the absolute value of all inward flow in defined as φ_{in}. The sum of all outward flow is defined as φ_{out}. The total flow φ_{total} is the sum of φ_{in} and φ_{out}.

$$\begin{aligned}\varphi_{in}&=\left|\varphi_1^-+\varphi_2^-\right|\\ \varphi_{out}&=\varphi_1^++\varphi_2^+\\ \varphi_{total}&=\varphi_1^++\left|\varphi_1^-\right|+\left|\varphi_2^-\right|+\varphi_2^+\end{aligned} \tag{8}$$

The value φ_{net} is the sum of all flows along the boundaries of state z without the use of absolute values. A positive value of φ_{net} represents a net outward flow out of state z. A negative value of φ_{net} represents a net inward flow into state z. The sum of the flows along the boundary of state z is given by Eq. (9).

$$\varphi_{net}=\varphi_1^-+\varphi_1^++\varphi_2^++\varphi_2^- \tag{9}$$

While the sign of φ_{net} is important to determining if current state z will transition to a new state, it does not contain information about which state it transitions to. Instead, careful manipulation of Eq. (8) builds transition probabilities through each side.

The notion of probability can be interpreted as counting types of occurrences and then normalizing the count of each type by the total occurrences. Suppose the occurrences of outward and inward flow defined in Eq. (7) are normalized by the total flow defined in Eq. (8). For example, the probability to transition up will be defined as the outward flow through side 2, φ_2^+, divided by the total flow φ_{total}. We can then define z^+ as the state above current state z and define z^- as the state below current state z. Equation (10) gives the probability to stay within the current state and the probability to transition up or transition down to an adjacent state. Uniform probability distribution is assumed along the borders of each state.

$$
\begin{aligned}
1 &= \frac{\varphi_{in}}{\varphi_{total}} + \frac{\varphi_{out}}{\varphi_{total}} \\
1 &= \frac{\left|\varphi_1^- + \varphi_2^-\right|}{\varphi_{total}} + \frac{\varphi_2^+}{\varphi_{total}} + \frac{\varphi_1^+}{\varphi_{total}} \\
1 &= \Pr(z' = z \mid z) + \Pr(z' = z^+ \mid z) \\
&\quad + \Pr(z' = z^- \mid z)
\end{aligned} \tag{10}
$$

At each time step the probability to stay or transition up or transition down is calculated using the current state boundaries and the current input. This information builds a time-varying probability transition matrix named L that can be constructed as shown in Table 1 for the example of current state z=2 at time t.

Table 1. Example of probability transition matrix L for current state z=2 at a time t

		Future State z'			
		1	2	3	4
Current State z	1	0	0	0	0
	2	$\Pr(z' = z^- \mid z)$	$\Pr(z' = z \mid z)$	$\Pr(z' = z^+ \mid z)$	0
	3	0	0	0	0
	4	0	0	0	0

6. Simulation Results

Previous work by Scacchioli et al. (2006) yielded a complete nonlinear EPGS model. This nonlinear model uses ω_e, I_{load}, and V_{ref} as inputs and yields V_f, V_{dc}, and battery dc current I_{dc} as output as shown in Figure 9.

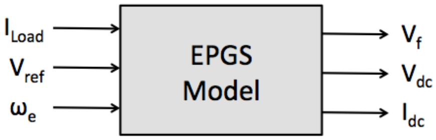


Figure 9. Schematic of EPGS model

Diagnostics for the belt fault case, diode fault case, and voltage controller fault case are accomplished by using the EPGS model and the new method for the probability transition matrix L. The EPGS output I_{dc} is quantized and sent to the flow calculator. The flow calculator uses the outputs of EPGS model and the f_2 equation from the DC electric machine model to calculate the flow φ through each side of the current quantized state. The flow and current quantized state are used to construct the probability transition matrix L. The quantized state and probability transition matrix L are used in diagnostics for fault detection and isolation. The procedure is illustrated in Figure 10.

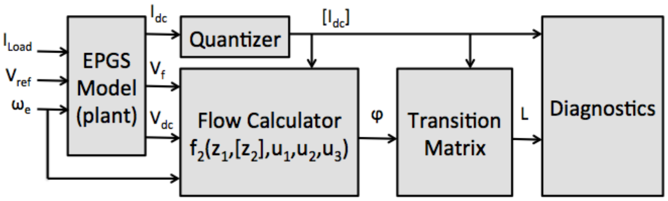


Figure 10. EPGS model with diagnostics

The following parameters were used to craft the inputs for a nonlinear EPGS Simulink model.

1. Simulate vehicle driving 289 seconds of FUDS cycle compressed to 72 seconds during simulation.
2. Simulation time step is 1e-4 seconds.
3. Tire radius of vehicle is 0.391 meters.
4. Final drive gear ratio is 4.72:1.
5. Belt ratio of 2.92 between engine crankshaft and alternator pulley.
6. Reference battery voltage V_{ref} is a constant 14.46 volts.
7. Current load profile is a square wave shown in Figure 11.

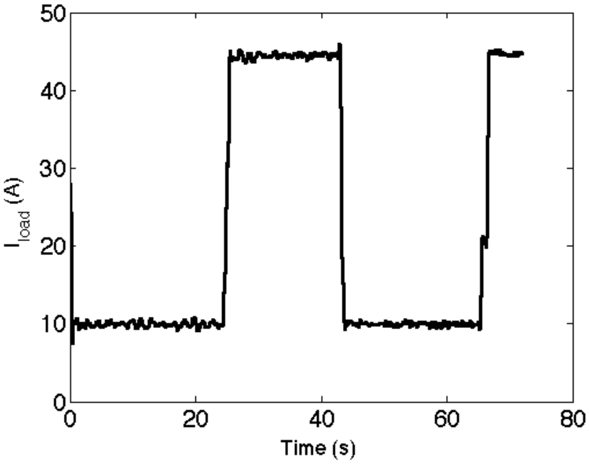


Figure 11. Current load profile

Using the aforementioned assumptions, ω_e can be easily calculated and is shown in Figure 12. The V_{dc} and V_f outputs of the EPGS model are given in Figure 13 and 14 respectively. The V_{dc} and V_f data will be used as inputs for the DC electric machine model but were downsampled to time step of 0.1 seconds.

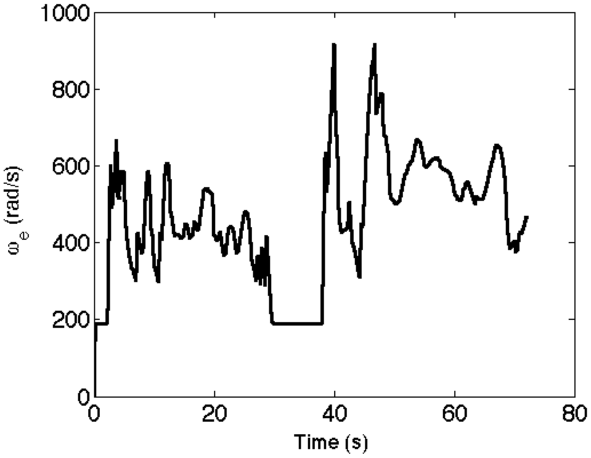


Figure 12. Alternator rotational speed input

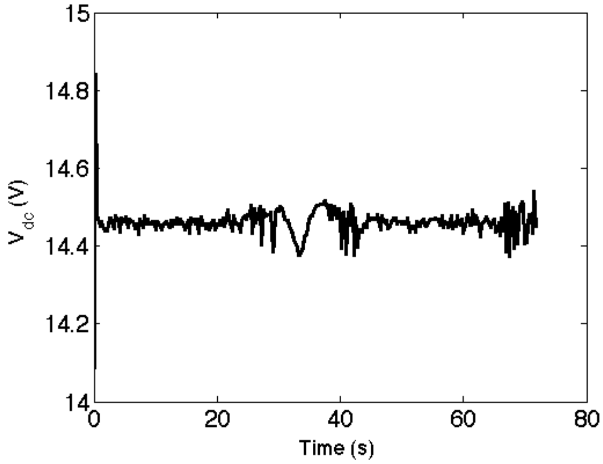


Figure 13. Battery DC voltage input

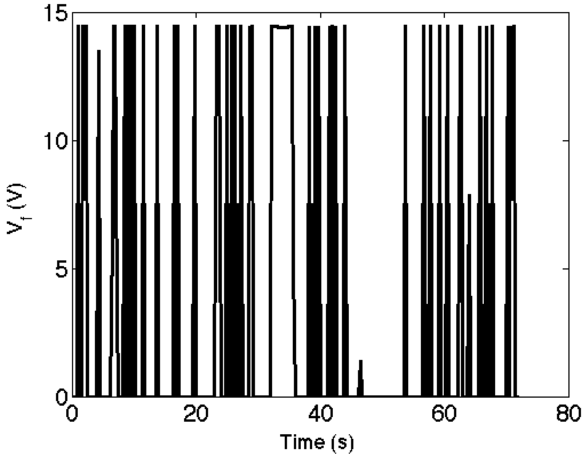


Figure 14. Alternator field voltage input

Figures 12, 13, and 14 represent the nominal inputs to the DC electric machine model to which faults will be injected.

Table 2 details the selected injection time and magnitude of fault relative to nominal that were injected during simulation. In other words, the nominal inputs were modified to simulate a fault.

Table 2. Fault injection time and magnitude

Fault	Injection time (s)	Modified Input	Resulting % drop with respect to nominal
Belt Slip	10	ω_e	0.8
Open Diode	10	V_{dc}	one broken diode
Voltage Controller	10	V_{ref}	0.3

Output z_2 range for nominal and faulty cases must be quantized into rectangles to find the probability transition matrix over time. Output z_2 is quantized into 12 states with names 1-12. The same boundaries and names will be used for faulty cases as well.

The z_1 range for this simulation is z_1^{min} is -2.210e+06 and z_1^{max} is 6.683e+06. Given the z_1 range, the quantized states, and u_1, u_2, and u_3, the probability transition matrix can now be calculated using the f_2 function from Eq. (1).

The probability transition matrix L contains the prediction of the most likely quantized state z' = z_L and its probability P(z' = z_L) at the next time step. The most likely probability and most likely predicted state can be compared with the quantized output state [I_{dc}] that actually occurs. If there is a relatively high probability of a particular state transition occurring and that state transition does not occur, then a fault may be present. An example of the most likely transition probabilities, most likely states, and output states over time for belt fault case is shown in Figure 15.

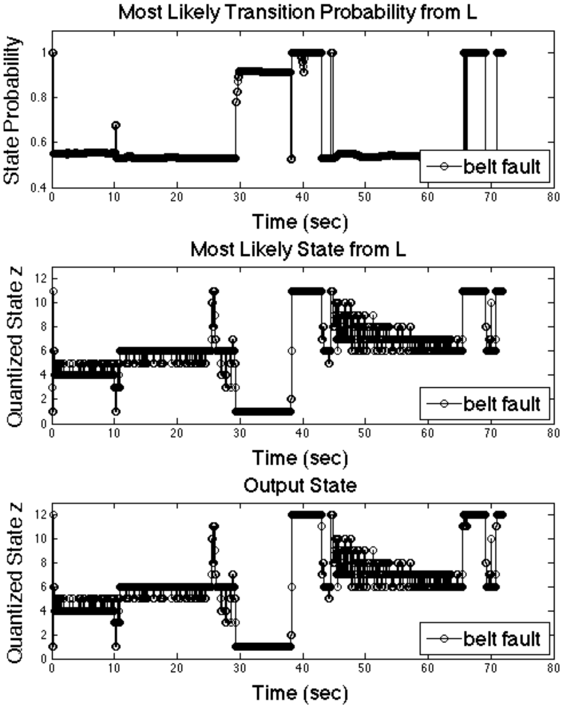


Figure 15. Belt fault outputs

Disagreement between predicted and output states are clear after calculating the difference of quantized output state [I$_{dc}$] and the predicted state from probability transition matrix L. This difference is defined as the residual r in Eq. (11). The residual results for each fault case are shown in Figures 16, 17, and 18.

$$r = [I_{dc}] - z_L \tag{11}$$

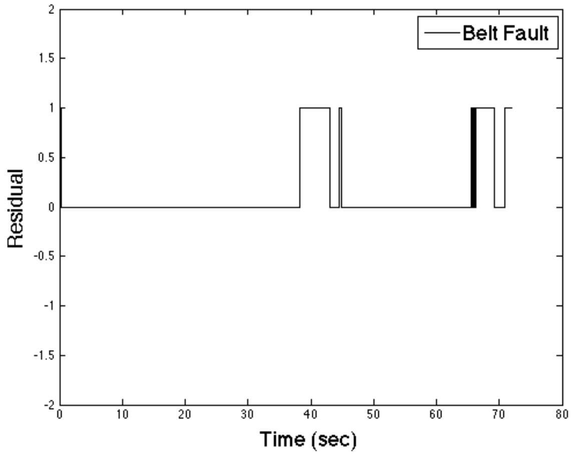


Figure 16. Belt fault residual

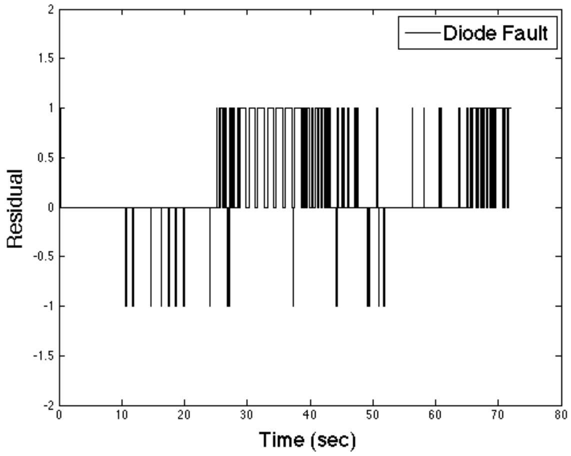


Figure 17. Diode fault residual

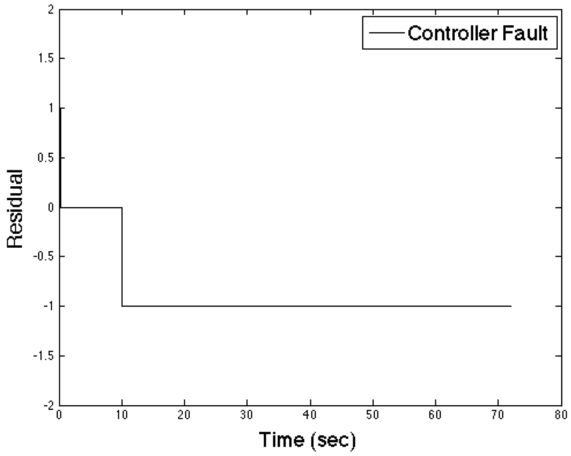


Figure 18. Voltage controller fault residual

All three fault cases show a short-term disagreement r ≠ 0 between predicted and output states at time t=0.2 seconds but returns to agreement r = 0 immediately at t=0.3 seconds. The disagreement occurs before a fault is injected at time t=10 seconds. This disagreement at t=0.2 could trigger a false alarm during fault detection. Similar rapid switching behavior also occurs in the diode fault residual in Figure 17. To distinguish between the similar switching behavior of false alarms with real faults and to build confidence in the diagnostic algorithm, a fault will only be detected if the residual shows disagreement for at least 0.2 seconds. The belt fault will be detected at t=38.4 seconds. The diode fault will be detected at t=10.7 seconds. The controller fault will be detected at 10.2 seconds.

Isolation of a detected fault will be achieved by monitoring the switching behavior during a finite time window following detection. The belt fault appears in the residual when the load current increases or decreases. Due to the quick duration of load current change, the belt fault is also present for a short time in the residual lasting between two to four seconds. The diode fault causes a large ripple in the alternator output current. This ripple causes frequent and rapid switching behavior from agreement to disagreement in the residual. The controller fault is the only fault case where there is residual disagreement for the entire duration of the fault. Therefore, the mean $\bar{r}$ of the absolute value of the residuals during a finite time window can be used to isolate each fault as defined in Eq. (12). The time window is chosen based on data behavior. For the data in this paper, a six second window was used. Table 3 shows the mean value calculations for each fault using the six second window immediately after fault detection.

$$\bar{r} = \frac{\sum_{i=1}^{n} |r_i|}{n} \qquad (12)$$

Table 3. Mean $\bar{r}$ for six second window

Fault	Mean $\bar{r}$
Belt Slip	0.75
Open Diode	0.08
Voltage Controller	1

Appropriate constant thresholds for $\bar{r}$ can isolate the fault. For this paper, if $\bar{r}$ is between 0.5 and 1 the fault is due to belt slip. If $\bar{r}$ is 1 the fault is due to the controller. Otherwise, the fault is due to an open diode.

Based on this approach, the belt fault will be isolated at t=44.4 seconds; the diode fault will be isolated at t=16.7 seconds; the controller fault will be isolated at time t=16.3 seconds.

Different fault magnitudes might require different isolation thresholds. This paper only considers three discrete fault modes.

7. Conclusion

This paper presents a novel method for calculating a time-varying probability transition matrix L for a quantized nonlinear system with the purpose of fault detection and isolation. Matrix L exploits the linear state space system that approximates the nonlinear system thereby reducing computational effort. Simple comparison of most probable state transitions from L and the quantized output states over time leads to fault detection and isolation. The merit of using matrix L for diagnostics is shown through the successful application of fault isolation in a 2D quantized alternator system.

Acknowledgements

Support for this research has been provided in part by the US Department of Energy GATE program and National Science Foundation under Grant No. 0825655.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Nomenclature

ω_m	engine rotational speed
ω_e	alternator rotational speed
V_{dc}	battery DC voltage
V_f	field voltage
V_{ref}	voltage controller reference
I_{dc}	alternator output current
I_{load}	vehicle load current
I_B	battery charging current
z_1	first state space state
z_2	second state space state and output
u	state space input
$a(\omega_e)$	state space parameter dependent on alternator rotational speed
$b(\omega_e)$	state space parameter dependent on alternator rotational speed
z	current state
z’	possible future state
z_1^{min}	minimum z_1 value
z_1^{max}	maximum z_1 value
z*	flow transition point on z_1 axis on side 1 of state z
z**	flow transition point on z_1 axis on side 2 of state z
$z_2^{(1)}$	upper boundary of state z
$z_2^{(2)}$	lower boundary of state z
φ^+	flow up
φ^-	flow down
f	general function
$\bar{F}$	Field vector
$\bar{n}$	normal vector
C	general closed curve
A	area within curve C
r	line integral direction along curve C
φ_{in}	total flow into state z
φ_{out}	total flow out of state z
φ_{net}	net flow for given state z
z^+	state above state z
z^-	state below state z
L	time varying probability transition matrix
$[I_{dc}]$	quantized alternator output current
z_L	predicted future state using L
r	residual
$\bar{r}$	mean of absolute value of residual
n	number of data points

REFERENCES

Alam, M. & Lundstrom, M. (1995). Transition Matrix Approach for Monte Carlo Simulation of Coupled Electron/Phonon/Photon Dynamics, *Applied Physics Letters* (Vol 67. No. 4 pp. 512-514), July 24. doi: 10.1063/1.114553

Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2006). *Diagnosis and Fault-Tolerant Control*. Germany: Springer.

Hashemi, A., & Pisu, P. (2011). Adaptive Threshold-based Fault Detection and Isolation for Automotive Electrical Systems (pp. 1013-1018), *World Congress on Intelligent Control and Automation*. June 21-25, Taipei, Taiwan. doi: 10.1109/WCICA.2011.5970668

Hashemi, A., & Pisu, P. (2011). Fault Diagnosis in Automotive Alternator System Utilizing Adaptive Threshold Method. *Annual Conference of Prognostics and Health Management Society*. September 25-29, Montreal, Canada.

Hashemi, A., (2011). *Model-Based System Fault Diagnosis Utilizing Adaptive Threshold with Application to Automotive Electrical Systems*. Masters dissertation. Clemson University, Clemson, South Carolina, USA. http://etd.lib.clemson.edu/documents/1314212419/Hashemi_clemson_0050M_11327.pdf

Scacchioli, A., Rizzoni, G., & Pisu, P., (2006). Model-Based Fault Detection and Isolation in Automotive Electrical Systems, *ASME International Mechanical Engineering Congress and Exposition* (pp. 315-324), November 5-10, Chicago, Illinois, USA. doi: 10.1115/IMECE2006-14504

Scacchioli, A., Li, W., Suozzo, C., Rizzoni, G., Pisu, P., Onori, S., Salman, M., Zhang, X., (2010). Experimental Implementation of a Model-Based Diagnosis for an electric Power Generation and Storage Automotive System, Submitted *ASME Tran. Dynamic Systems, Measurement, and Control*.

Scacchioli, A., Rizzoni, G., Salman, M., Onori, S., & Zhang, X. (2013). Model-based Diagnosis of an Automotive Electric Power Generation and Storage System. *IEEE Transaction on Human Machine Systems* (pp. 1-14), March 7. doi: 10.1109/TSMCC.2012.2235951

Zhang, X., Uliyar, H., Farfan-Ramos, L., Zhang, Y., & Salman, M. (2010). Fault Diagnosis of Automotive Electric Power Generation and Storage Systems, *IEEE International Conference on Control Applications* (pp. 719-724), September 8-10, Yokohama, Japan. doi: 10.1109/CCA.2010.5611179

BIOGRAPHIES

Sara Mohon was born in Groton, Connecticut in 1987. She received her B.S. in Physics from the College of William and Mary (Williamsburg, VA, USA) in 2009 and M.S. in Automotive Engineering from Clemson University (Clemson, SC, USA) in 2012. She is currently a Ph.D. student at Clemson University studying Automotive Engineering. She has completed summer internships at NASA Langley Research Center (Hampton, VA, USA) Thomas Jefferson National Accelerator Facility (Newport News, VA, USA), NOAA David Skaggs Research Center (Boulder, CO, USA), and Johns Hopkins University Applied Physics Laboratory (Laurel, MD, USA). She has completed a battery research project at BMW Manufacturing Company (Spartanburg, SC, USA) that resulted in filing a patent about methods to determine the condition of a battery. Her research interests are control, diagnostics, and prognostics for hybrid vehicles and electric vehicles. She is a member of ASME, SAE, SWE, and IEEE and received the national SEMA Top Student Award in 2012.

Pierluigi Pisu was born in Genoa, Italy in 1971. He received his Ph.D. in Electrical Engineering from Ohio State University (Columbus, Ohio, USA) in 2002. In 2004, he was granted two US patents in area of model-based fault detection and isolation. He is currently an Associate Professor in the Department of Automotive Engineering at Clemson University and holds a joint appointment with the Department of Electrical and Computer Engineering at Clemson University. He is also a faculty member at the Clemson University International Center for Automotive Research. His research interests are in the area of fault diagnosis with application to vehicle systems, and energy management control of hybrid electric vehicles; he also worked in the area of sliding mode control and robust control. He is member of the ASME and SAE, and a recipient of the 2000 Outstanding Ph.D. Student Award by the Ohio State University Chapter of the Honor Society of Phi Kappa Phi.

On Optimizing Anomaly Detection Rules for Gas Turbine Health Monitoring

Weizhong Yan[1], Lijie Yu[2], Jim Sherbahn[3], and Umang Brahmakshatriya[4],

[1,4]*General Electric Global Research Center, Niskayuna, NY, 12309, USA*
yan@ge.com
brahmaks@ge.com

[2,3]*General Electric Power and Water Engineering, Atlanta, GA, 30339, USA*
Lijie.yu@ge.com
James.sherbahn@ge.com

ABSTRACT

Gas turbine health monitoring is a critical process in preventing costly unplanned maintenance and secondary damage. To monitor gas turbine health, control signals are typically collected and analyzed using anomaly detection rules and models to assess failure likelihood based on observed data patterns. An analytic designer will often deal with rule optimization tasks in order to maximize failure detection and reduce false alarms. Manual tradeoff analysis is typically time consuming and suboptimal. In this paper, we attempt to address this issue by introducing a strategy for automatic and efficient rule optimization. By focusing on optimizing rule parameters while keeping rule structure intact, we maximize the rule performance by integrating domain knowledge with data driven optimization techniques. Realizing that automated rule tuning can be computationally expensive and infeasible to complete in reasonable time, we will leverage our recently-developed scalable learning framework - iScale that allows for automatically distributing rule tuning tasks to a large number of cloud computers, which not only dramatically speeds up tuning process, but also enables us to handle big size of historical data for tuning. We also explore different search methods to make rule tuning more efficient and effective and finally demonstrate our rule optimization strategy by a real-world application.

1. INTRODUCTION

Today thousands of GE manufactured gas turbines are serving customers worldwide for a wide variety of industrial applications. Most customers are adopting a contractual service agreement (CSA) with GE and rely on GE's OEM expertise for actively monitoring turbine health, i.e., to proactively detect anomalies and prevent costly unplanned maintenance. Aiming for more accurate and robust detection of incipient faults as early as possible, over the years we at GE have developed and fielded a spectrum of advanced analytics models (both rule-based and data-driven models as well).

Pure data-driven modeling techniques work well if sufficient labeled data are available. However in real-world applications like in gas turbine monitoring, obtaining sufficient labeled data is labor-intensive, if ever possible. In particular, true positive cases might be sparse or noisy. Using small set of labeled data for data-driven modeling may cause model over-fitting or ill-formed model representation. In addition, pure data model may not have explicit knowledge structure or explainable reasoning logic that engineers prefer, which often hinders user acceptance of the model.

Consequently, for gas turbine health monitoring applications, rule-based models are still dominantly used. In fact, most PHM systems make use of diagnostic rules in one form or another. For the sake of clarity, consider one of the simplest forms, which has the following form.

$$\text{IF } (T < X) \text{ THEN } (\text{STATE} = A)$$

What this rule basically says is that if a parameter (X) has a value that exceeds a limit (T), the system is in a specific state (A). Generally the state identified reflects a particular, degraded state. It is worth stressing that real-world rules in gas turbine monitoring are typically much more complex, not only consisting of a large number of such simple-form rules, but also having complex-form rules.

Traditionally decision rules and their associated rule constants are determined by domain or engineering experts

W. Yan, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

based on their understanding of the physical system. The rules are continuously refined until each rule produces acceptable detection accuracy. Such manual refinement not only is labor-intensive, but also often fails to find true optimal values.

To address the above-mentioned challenge, in this paper we take a different approach, that is, to maintain the existing rule knowledge forms, but leverage machine learning based optimization platform to improve rule performance. By maintaining the expert rule forms we ensure correct physics understanding is maintained in rule logic. Tunable parameters, or knobs, are selected to optimize rule performance using training and testing data sets. The derived analytics benefits from both domain knowledge capturing as well as data driven optimization. The impact of having such a capability is significant in that instead of requiring detailed "face time" of an expensive engineer, an analyst could use this process to learn from labeled data and the tuning could be done in an automated and even online manner.

To that end a generic rule optimization platform has been developed, which is independent of specific rules or rule platforms. It allows a user to create a rule tuning job through a web-based configuration interface. A user has the flexibility to choose among unit level or fleet wise optimized rule.

In the remainder of this paper, we will first present the architecture of cloud-based machine learning system - iScale, followed by a discussion of available optimization method available. A case study will then be presented using iScale to perform a specific rule tuning job, and then the conclusions summary will be provided in the last section.

2. Scalable Learning Framework - iScale

Creating solutions for analytically hard problems is presently a time- and cost-intensive process. This is largely due to the fact that the design of advanced analytic solutions is largely manual, requiring involvement of one or more analysts. These analysts apply their specific knowledge and expertise within a given area of analytic problem-solving to create an acceptable solution. This process has resulted in a major bottleneck in the company’s ability to create advanced analytic solutions rapidly. Aiming at tackling this bottleneck, we at GE have been developing a cloud-enabled analytics framework (called iScale) [Yan et al (2011)].

iScale is primarily designed to be a distributed computing environment for creating, refining, deploying and maintaining analytic solutions. As shown in Figure 1, the core of the framework consists of several key components, including the job manager, the resource manager, the job scheduler, the executer, and the optimizer. These components serve as an “orchestrator” among users, compute machines and algorithms. Specifically it takes user’s inputs (data and performance requirements, etc.), picks a subset of algorithms in the library that are most relevant to the problem, intelligently distributes the search tasks to different computer resources, and outputs the best combination of models and associated model parameters that maximally meet user specified performance requirements. The framework provides a web-service that can be accessed from a laptop computer and other mobile devices as well. The framework also maximally leverages

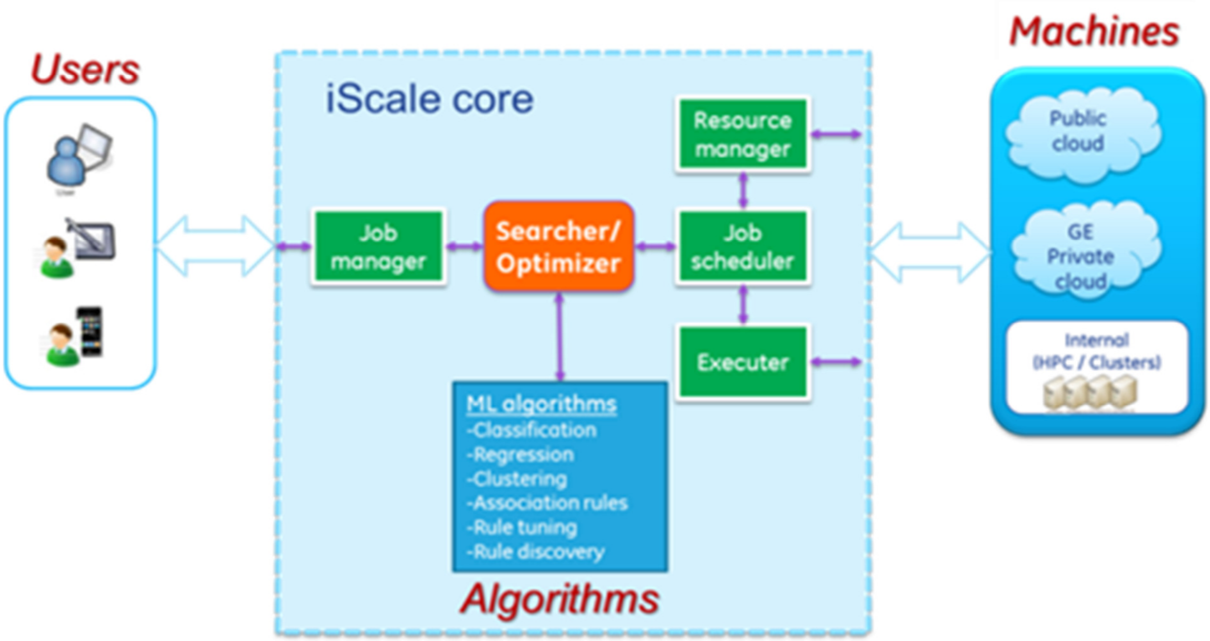


Figure 1: iScale Framework

heterogeneous compute machines (cloud machines and internal HPC/clusters) and is flexible in integrating different algorithms written in different languages, e.g., C/C++, Java, and R. More importantly, it is highly scalable, that is, it is capable of handling large size of data by leveraging distributed computing technology.

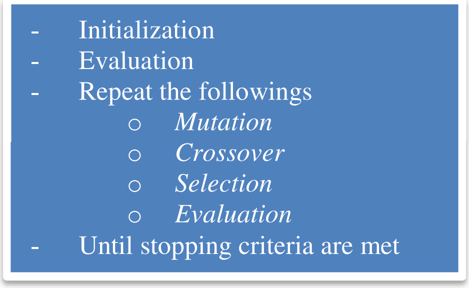
A critical component of iScale involves automating a significant portion of the currently manual process involved in problem formulation, data preparation, model selection, model tuning, domain knowledge integration and ensemble creation. The larger aspiration of iScale is to move analytic development activity from a one-off, largely manual process to a one-click, largely automated process. This is expected to significantly increase the rate at, and the ease with, which a significantly larger employee population in the company will create analytic solutions. It will help to make the creation of analytics an increasingly pervasive activity across the entire company. iScale will be the ecosystem in which analytic modules are born, sustained and continuously improved. A by-product of having such an ecosystem is it can bring to bear a large number of diverse analytic approaches to a single problem, thereby increasing the likelihood of finding a solution of very high quality. As an ecosystem, it is also expected to release next generation innovation that is evident in other similar analogues in the market today, like the Apple AppStore. By harnessing the virtually infinite and elastic compute power of the cloud, iScale is able to conduct a comprehensive and iterative search for the optimal analytic solution to a problem from across a diverse array of applicable approaches. Thus iScale is well suited for rule tuning as well.

3. OPTIMIZATION METHODS

Rule tuning is considered as an optimization problem where design space is defined by the tunable variables and objective function is defined by the rule performance metrics, i.e., probability of detection (POD) and false alarm volume (FAV). The rule tuning optimization problem has an important feature, that is, its objective values are available, but the derivative of the objective function is not computable. Another feature associated with rule tuning optimization is that the objective function evaluation is computationally expensive. These two features call for derivative-free (also called zero-order based) optimization methods [Rao (2009)] for rule tuning. Also, since multiple objectives (performance metrics) are involved in rule tuning, rule tuning is characterized as a multi-objective optimization (MOO) problem [Marler & Arora (2004)].

In literature there are many different derivative-free optimization methods [Conn et al (1997)]. In this paper we employ two different optimization methods, grid search (GS) and differential evolution (DE), for rule tuning. While both GS and DE fall to the category of global optimization methods [Rao (2009)], GS is a deterministic optimization method and DE, on the other hand, is a stochastic (also called heuristic or meta-heuristic) optimization method.

GS performs optimization as follows: dividing the n-dimensional design space into a n-dimensional grid, evaluating the objective function at all of these grid points, and picking the grid point that gives the minimal (or maximum) objective function value. Grid search is a simple global optimization method and can be easily distributed to many computer nodes to speed up the search process. The main drawback is that it suffers from the *curse of dimensionality*, i.e., the number of objective function evaluations grows exponentially with the number of design parameters.



- Initialization
- Evaluation
- Repeat the followings
    - *Mutation*
    - *Crossover*
    - *Selection*
    - *Evaluation*
- Until stopping criteria are met

Figure 2 – General procedure for differential evolution

DE is a simple, but powerful at the same time, population-based, stochastic optimization method [Storn & Price (1997)]. Like other population-based optimization methods (e.g., GA, PSO), DE optimization follows the general procedure as shown in Figure 2. Essentially, an initial population of solutions is randomly generated and evaluated; and those solutions are improved upon by applying mutation, crossover, and selection operators until a stopping criterion is met.

Compared to other EA optimization methods, DE optimization has several advantages, including fast convergence, having fewer control parameters, and ease in programming. As a result, DE has been used to solve a wide range of real world optimization problems [Das & Suganthan (2011)].

For multi-objective optimization problems, since there rarely exists a single solution that optimizes all objectives simultaneously, the optimum is given by a set of solutions known as the Pareto optimal set. The elements in this set are said to be non-dominated since none of them is better than the others in terms of all objectives. Using DE for MOO problems involves changes to its operators, mutation, crossover, and selection. Many different design strategies have been proposed. For details, refer to [Xue, et al (2003) and Reyes-Sierra & Coello (2006)].

Figure 3 illustrates the flow diagram of using DE for optimizing rules. In the optimization process shown in the diagram, the step where rule engine execution based on a specified parameter set and the data is the most computational expensive one. That is where iScale helps by distributing the rule engine execution tasks to many compute nodes so that rule evaluation can be performed simultaneously. In the diagram, GS and DE differ in that GS is a one-pass operation while DE involves many iterative steps until convergence condition is met. Another difference between GS and DE is that the parameter set is pre-defined grid in GS while in DE the initial parameter set is randomly generated.

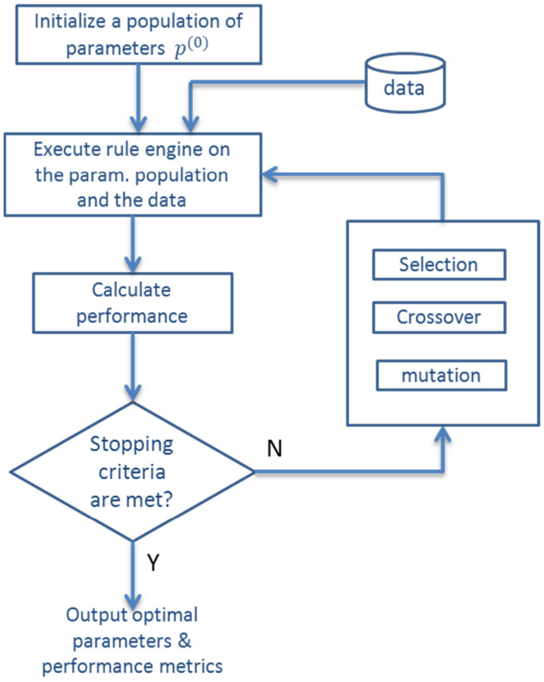

Initialize a population of parameters $p^{(0)}$
data
Execute rule engine on the param. population and the data
Selection
Crossover
mutation
Calculate performance
Stopping criteria are met?
N
Y
Output optimal parameters & performance metrics

Figure 3: Flow diagram for rule optimization

4. Rule Tuning Case Study

In this section we provide a use case study about performing a specific rule tuning job using iScale.

We will first give a brief introduction to the rule to be tuned. We then provide details of rule tuning process (the critical steps and data sets, etc.). At the end of the section, rule tuning results will be discussed.

4.1. The Vibration Rule for Turbine Vibration Monitoring

The rule concerned in this paper is the gas turbine vibration rule. Most gas turbines are equipped with proximity probes and seismic sensors located on the bearing housings. They provide key indicators of gas turbine hot section system integrity, including bearing damage, rotor imbalance, sudden mass loss, etc. Figure 4 shows an example of proximity sensor signals with step shift resulted after turbine blade migration due to lock wire failure. Event occurred shortly after unit was restarted and reaching high load, when multiple sensors had a step shift with increased vibration level.

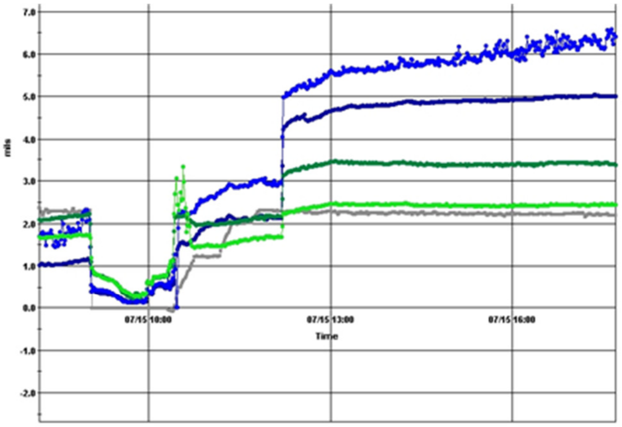


Figure 4 - Vibration Step Change Due to Turbine Blade Migration

The vibration rule is developed to examine probe signals to detect trend or step change. Once significant change is detected, statistical test is performed to assess confidence of the change. Both seismic and proximity probes will be analyzed and fused to improve alarm confidence.

Vibration signal alone may not be sufficient to determine system condition. Variation caused by operating condition change such as load shift or turbine speed change may also cause similar vibration signal shift. To separate true HGP failure from operation status change, more logic is added in the vibration to establish enabling criteria and stability criteria.

4.2. Vibration Rule Tuning Process

Within iScale platform, performing a rule tuning job is standardized into a straightforward process as shown in Figure 5. These can be carried out during the initial analytics design and development phase, or during the life cycle management for rule improvement. On the high level there are four steps from end to end.

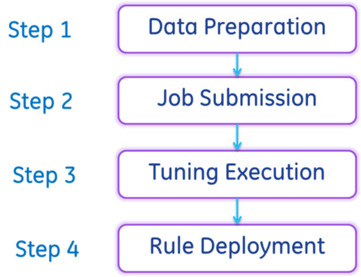


Figure 5 - iScale Rule Tuning Process

4.2.1. Data Preparation

As in any data driven modeling work, high quality data with representative feature set is a key for gaining predictive capability of the underlying analytics. Data preparation is certainly important for rule optimization.

Three different types of data set are required for rule tuning. First is historical event ground truth data. Since we focus on anomaly detection rules, these include both abnormal units and normal units, which are referred as POD cases and FAR cases, respectively.

Secondly, time series of rule input data of each historical case are extracted from the data historian, which will be used in rule evaluation. One minute resolution is used for all vibration case data. A week or so before and after POD events are prepared, and five months of FAR data is used in the vibration case study.

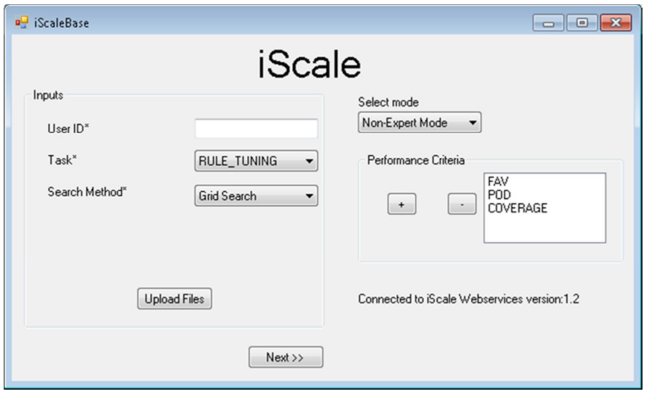
Thirdly, tunable rule parameters are identified and their valid ranges are also specified. In the vibration rule tuning case, there are thirty or so parameters can be adjusted outside the rule. Some values are set based on unit configuration or material properties. Among which four high sensitivity parameters are selected as the tuning target. For intellectual property protection, we are not allowed to give details of the four parameters. Valid ranges of each tunable parameter are also specified in the template, which defines the optimization search space.

4.2.2. Job Submission

iScale rule tuning is provided as a web service. A user can login to the web interface to create and monitor a tuning job. To create a new job, the need to provide information of the rule platform, rule executable package, training and testing data set, and evaluation criteria. The main interface screen is shown in Figure 6.

iScale itself is rule platform independent. However, platform specific rule wrapper and rule analysis engine will be required to execute a rule tuning within iScale to perform evaluation. The vibration rule is implemented based on CCAP platform, a GE in-house developed platform originally developed for US NAVY, specialized in plant equipment monitoring and diagnostics.

Performance evaluation criteria are also defined during rule tuning job creation, which can be multiple objectives. For anomaly detection rules, it is typically to maximize probability of detection (POD) and minimize false alarm volume (FAV). For vibration rule, an additional criterion is added to maximize rule enabling coverage. A weighting mechanism can be established to merge different criteria into an overall objective function.


iScaleBase

iScale

Inputs

User ID*

Task* RULE_TUNING

Search Method* Grid Search

Select mode Non-Expert Mode

Performance Criteria

FAV
POD
COVERAGE

Connected to iScale Webservices version:1.2

Figure 6 - iScale Rule Tuning Interface

4.2.3. Tuning Execution

Once a rule tuning job is submitted by user, iScale will create a unique job ID to trace a specific task. It will also manage job schedule, data file transfer, resource allocation, and configure rule execution and evaluation. Rule configuration search space will be traversed based on the optimization method selected by user. The execution of search is performed on a number of compute nodes simultaneously.

Execution time varies depending on the data set size, search space size, and number of available nodes. A user may login to the iScale job status checking screen to monitor job execution progress, modify or abort submitted job.

After search completed, iScale will aggregate results from different computing node to rank and summarize final result. Based on user configuration, optimization can be performed in either fleet level or unit level. Both summary result and detailed raw result are maintained for user review.

4.2.4. Rule Deployment

The final result provided to user includes the optimized rule configuration parameters and the corresponding evaluation criteria metrics. iScale can be configured as a fully integrated component within analytics development system. The integration is fairly simple since only the optimized rule configuration parameters are required to be deployed. After user review, the validated rule configuration information can be directly deployed to production system.

4.3. Rule Tuning Results

The data: 6-month of historical data for 18 turbines of the 7FA fleet were retrieved from our database and used for demonstrating iScale rule tuning. The data has a sample rate of once per minute, which leads to approximately 259k (6*30*24*60) data points per turbine unit. Out of the 18 turbine units, eight of them are considered as POD cases, i.e., have valid events at some point of the 6-month period. The rest of the 10 turbines units have no valid events.

The search grid for grid search: As discussed in Section 4.2.1, four rule parameters were selected as the tunable parameters. Due to IP issue, we are not allowed to give details of the specific tunable parameters. Here we designate the four tunable parameters as v_1, v_2, v_3, and v_4. We use 3 levels for each of the parameters and we arrive in grid with 3^4=81 grid points (or DOE experiments).

The configuration for DE optimization: For DE optimization, we integrate the ECJ package (http://cs.gmu.edu/~eclab/projects/ecj/) into the iScale framework. The population size and number of generations are set to 50 and 20, respectively. Other DE parameters (e.g., mutation and crossover rates) are set to ECJ default values.

Table 1: Rule tuning results

	Original design	After tuning
design parameters	[20,0.04, 0.3,0.4]	[10,0.04, 0.5,0.4]
POD	100%	100%
FAV	255	12

Rule tuning performance results: Table 1 shows rule tuning results using grid search. For comparison purpose, also shown in Table 1 are the performance metrics (POD and FAV) for the default setting. As seen from the table, rule tuning reduces the number of false alarms from 255 to 12 for the 7FA fleet data concerned in the paper. The reduction of false alarms improves monitoring engineer's productivity and prevents unnecessary inspection or troubleshooting. We have to point out that the DE optimization is still a work in progress, and we would like to share the results in a later time.

5. Conclusions

Gas turbine health monitoring is critical in preventing costly unplanned maintenance and in reducing life-cycle costs of power plant operations. Currently a great majority of anomaly detection engines are rule-based; and the rules and their associated thresholds/constants are initially designed based on domain and engineering knowledge and manually modified based on the rules' performance in the field. To improve fault detection performance (accuracy and robustness), systematical and efficient approaches allowing for optimally determining rules (rule discovery) and their associated constants (rule tuning) are needed. This paper is our initial effort towards addressing the need. Specifically we propose a way to automatically find optimal rule constants based on historical data; that is, we attempt to address the rule tuning need. Realizing that automated rule tuning can be computationally expensive and infeasible to complete in reasonable time, in this paper we leverage our recently-developed scalable learning framework - iScale that allows for automatically distributing rule tuning tasks to a large number of cloud computers. Such distribution not only dramatically speeds up tuning process, but also enables us to handle big size of historical data for tuning. In this paper we also explore different search methods to make rule tuning more efficient and effective.

By tuning the vibration rule, a real-world gas turbine detection rule, we demonstrate that the proposed rule tuning can be effective and efficient. The iScale-enable rule optimization not only eliminates the needs for manual tweaking thus a productivity gain for rule development, but also enables fully automated rule deployment and future adaptation, reduces the overall rule life cycle maintenance cost.

In future we would like to explore other optimization methods to further improve efficiency and effectiveness of rule tuning. It is also our great interest to extend our current automated rule tuning to automated rule discovery.

Nomenclature

CSA	contractual service agreement
DE	differential evolution
EA	evolutionary algorithm
FAV	false alarm volume
POD	probability of detection
MOO	multi-objective optimization

REFERENCES

Conn, K. Scheinberg, and P. L. Toint (1997), On the convergence of derivative-free methods for unconstrained optimization, in *Approximation theory and optimization*, M. D. Buhmann and A. Iserles, eds., Cambridge, 1997, Cambridge University Press, pp. 83-108.

Das, S. and Suganthan, P.N. (2011), "Differential Evolution: A Survey of the State-of-the-Art," IEEE Trans. On *Evolutionary Computation,* vol.15, no.1, pp.4-31, Feb. 2011.

Marler, R.T. and Arora, J.S. (2004), Survey of multi-objective optimization methods for engineering, Structural and Multidisciplinary Optimization, 26 (6) (2004), pp. 369–395.

Rao, S. (2009), *Engineering Optimization: Theory and Practice*. 4th Edition, John Wiley & Sons, Inc., Hoboken, New Jersey.

Reyes-Sierra, M. and Coello, C.A.C. (2006), Multi-objective Particle Swarm Optimizers: A Survey of the State-of-the-Art, International Journal of Computational Intelligence Research, Vol.2, No.3, 2006, pp.287-308.

Storn, R. and Price, K. (1997), “Differential Evolution, A Simple and efficient Heuristic Strategy for Global Optimization over Continuous Spaces", Journal of Global Optimization, Vol. 11, pp. 341-359.

Xue, F., Sanderson, A. C., and Graves, R. J. (2003), “Pareto-based multiobjective differential evolution.” *Proc., 2003 Congress on Evolutionary Computation (CEC’2003)*, Vol. 2, IEEE, New York, 862–869.

Yan, W., Iyer, N., Bonissone, P. and Varma, A. (2011), "iScale – Next Generation Framework for Creating Machine Learning Models", GE Global Research Center Whitepaper 2011.

Development of Virtual Sensor Networks to Support Accident Monitoring Systems

Rizwan Ahmed[1], Pak Sukyoung[1], and Gyungyoung Heo[1], Jung-Taek Kim[2], Seop Hur[2], Soo Yong Park[2], Kwang-II Ahn[2]

[1]*Department of Nuclear Engineering,Yongin-si,Gyeonggi-do, 446-701, South Korea*

rahmed.ne@gmail.com
pak.sukyoung@gmail.com
gheo@khu.ac.kr

[2]*Korea Atomic Energy Research Institute,Yuseong-gu, Daejeon, 305-353, South Korea*

jtkim@kaeri.re.kr
shur@kaeri.re.kr
sypark@kaeri.re.kr
kiahn@kaeri.re.kr

ABSTRACT

In a nuclear power plant (NPP), most of the systems are linked due to processes of fluid flow, heat transfer etc., and their natural tendency to respond to changes during accident conditions. These relationships can be utilized to develop smart applications for plant accident monitoring and management. In this research, the statistical relationships among the process parameters have been analyzed. It has been embarked that the characteristics of a safety system during a particular interval can be estimated by utilizing the other affected parameters, employing statistical correlation and regression models developed from the simulation data offline, when evaluated for the same set of conditions on accident sequence and safety systems. The proposed methodology has been demonstrated for a specific loss of coolant accident scenario using correlation coefficient and neural networks, for the time interval when containment spray system was initiated at the particular stage of accident progression and remained operational for some designed time. Virtual sensor networks were constructed for the estimation of reactor vessel level during that time period, which demonstrates the realization of methodology. The estimations from such virtual sensor networks are expected to improve by utilizing the importance measures and concepts to generalize the neural networks. Also, correlation voting index (CVI) provides a capability to select a set of related outputs, which would be used as a yardstick for comparing results in case, missing or uncertain inputs are present.

1. INTRODUCTION

Nuclear instrumentation and control (I&C) system is to provide reliable information on process parameters during normal and abnormal conditions. It should also have the capability to represent information regarding process and safety parameters in easily interpretable manner by numbers/displays. The display capabilities of I&C systems have been greatly improved after the TMI event 1979 where operators were failed to take right actions due to misinterpretation of available signals. Modern I&C systems are programmed with computers where they can simultaneously utilize the monitoring data for sensor or equipment fault diagnosis. Such intelligent systems have made online calibration and testing of sensors a reality. However, during the Fukushima accident in Japan, we could observe the modern I&C was again failed due to a combination of reasons that appeared to include: loss of power, evaporation of liquid in sensing lines, failure of sensors due to environmental conditions, instrument ranges that were not suitable for monitoring plant conditions, and lack of alternative data for use in validating instrument readings. Therefore, the capability of I&C systems is extremely important during severe accidents characterized by a combination of basic events and followed by failures of designed safety systems.

Nuclear industry has launched exhaustive research projects to address safety challenges in the severe accidents. At a broader level, for a complete accident management and emergency planning the areas to be focused are transitional procedures, onsite and offsite interactions, design and equipment and, and human and organizational factors. It has

Ahmed, Rizwan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

been suggested that the severe accident management guidelines (SAMG) could only be useful if the monitoring of critical parameters is somehow made available to the operator, even in station blackout condition as mentioned by (American Nuclear Society, 2012) and (US Nuclear Regulatory Commission, 1983). Loss of information on process parameters increases the chances of information misinterpretation at control room which may lead to wrong operator actions.

This research was motivated as a method to resolve aforementioned challenges. Our strategy is based on the development of an indirect way that is, a method to assess the safety critical parameters from other statistically related parameters.

2. Materials and Methods

US NRC's Regulatory Guide 1.97 provides the basic requirements for human-machine interface systems including I&C systems and for the monitoring of radioactivity releases following an accident. The SAMG classifies the important parameters in classes A-E (referred as SAMG parameters herein). Type A parameters are to provide primary information for manual operation. Type B parameters to provide information regarding the accomplishment of safety functions such as reactivity control, core cooling, maintain reactor coolant integrity, maintaining reactor containment integrity. Type C parameters are to provide information regarding variables that have a potential for causing a containment breach such as core exit temperature (CET), reactor coolant system (RCS) pressure, hydrogen concentration, containment pressure, and so on. Type D parameters are to indicate the operation of safety systems such as residual heat removal system, safety injection systems, refueling water storage tank level, primary coolant system, condensate storage tank level, containment cooling systems, radwaste systems, ventilation systems, power supplies etc., and Type E parameters are to indicate the amount of radioactive material to be released in case of containment breach. IEEE has also developed standards to support the specification, design, and implementation of accident monitoring instrumentation of NPPs. The recent document IEEE Std-497 (revision 2010) provides criteria for selection, performance, design, qualification, display and quality assurance of the nuclear I&C system (IEEE Power and Energy Society, 2010).Westinghouse has proposed an advanced system for post-accident monitoring (PAM) to implement the SAMG parameters (Westinghouse Electric Company, 2012). The representative parameters are CET, reactor vessel level, hot and cold leg temperatures, RCS pressure, and so on (referred as PAM parameters herein).

The focus of this research was on the PAM and SAMG parameters which should be secured during a severe accident to see the working of safety functions and their influence on accident progression. In this study the methodology to improve the information availability, by utilizing the statistical correlations among the PAM and SAMG parameters has been presented with a case study. Accident simulation data generated from the MAAP code for a probable loss of coolant accident (LOCA) scenario that led to containment damage (Park, 2009). The MAAP code generates time series data for more than 800 parameters following an initiating event for 2 days. This study was initiated to explore the following technical areas (Ahmed, 2013),

1. The statistical correlation of a process parameter with other parameters provides a basis for securing PAM parameters from SAMG parameters. The correlations among the process parameters can be utilized to estimate one parameter from the others. This would increase the virtual redundancy of the critical information.
2. The relationships among the process variables can be used to develop several virtual networks to generate an important parameter. Therefore, we can have capability of virtually supplying a safety-related sensor's information during normal operation of critical sensor and also this information will be available if the original signal is unavailable to the operator.

3. Development of PAM-SAM Relationship

Thousands of sensors are installed at an NPP to measure parameters that are important to draw metrics for its performance and safety condition. However, a smaller set of parameters is vital for safety management. US NRC has provided a SAMG on preferred process parameters be monitored during and following an accident. We hypothesized that the statistical relationships among the process parameters can be utilized to serve as virtual sensor networks where PAM parameters could be estimated by several sets of SAMG parameters. However, the sets of SAMG parameters used to estimate a PAM parameter are expected to differ due to the underlying boundary conditions and involved safety systems.

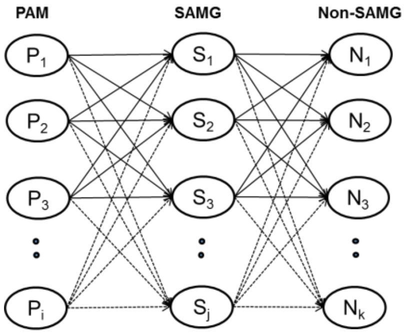


Figure 1: Structure of virtual sensor network.

The methodology can be directly extended to develop virtual networks to estimate SAMG parameters from other non-SAMG parameters as well. A system of such virtual networks is shown in figure 1. The connecting lines mean the statistical correlation and not a physical connection by wire or other data transferring mean. The complete stages for developing virtual networks for estimating PAM parameters are shown in figure 2. The remaining subsections are to discuss major processing step in brief.

3.1. Simulation Database

Major initiating events (1) large loss of coolant accident (LBLOCA), (2) medium loss of coolant accident (MBLOCA), (3) small loss of coolant accident (SBLOCA), (4) station blackout accident (SBO), (5) loss of off-site power accident (LOOP), (6) steam generator tube rupture accident (SGTR), and (7) loss of feed-water accident (LOFW) were simulated by using the MAAP code for Korean standard NPP, OPR-1000 (Park, 2009). The database comprised of a total of 70 accident scenarios analyzed on the basis of probabilistic safety analysis of OPR-1000 and presents the data for more than 800 thermal hydraulic and source term parameters for 72 hours following an accident.

3.2. Scenario Analysis

Accident management strategies have been developed and safety systems are designed to initiate when certain set of conditions meet and work for a particular time interval. With the MAAP code, the generated accident scenarios were to represent severe accident conditions, where several safety systems were assumed to fail. A set of representative LOCA scenarios are shown in figure 3 via event tree diagram, where the working of safety systems such as high pressure safety injection system (HPSIS), low pressure safety injection system (LPSIS), containment spray system (CSS) and cavity flooding system are conceivable. Scenarios having end state marked by a prominent yellow colored circle represent that the final Plant Damage State (PDS) was containment damage. To develop a particular application for severe accident monitoring system, the time intervals associated with the working or failure of safety systems should be considered to identify relationship among the parameters.

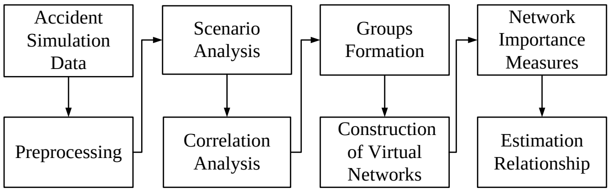


Figure 2: Steps for the development of parameter estimation relationships.

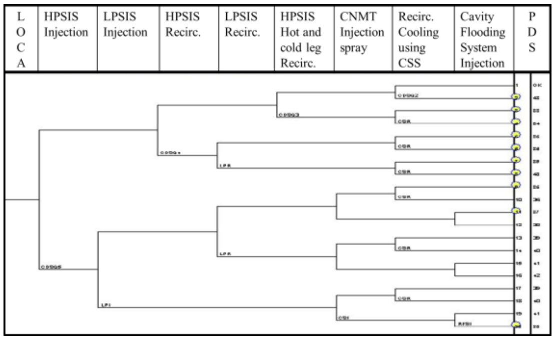


Figure 3: Event tree for a LOCA scenario.

3.3. Groups Formation

For the development of virtual sensor networks, statistical relationship among the process parameters was required during working or failure of a safety system. Correlation coefficient is the most widely used statistical measure and is being used in process industry as a basis for grouping related variables for online monitoring applications (Ahmed, et al., 2012; Heo, et al., 2012). Equation (1) defines the simple correlation for two variables ‘x’ and ‘y’ having ‘n’ values in each.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} \qquad (1)$$

Where s_x and s_y are sample standard deviations for given x and y vectors, respectively. An algorithm was implemented to collect top correlated SAMG parameters with PAM parameters on the basis of absolute value of correlation coefficient.

3.4. Construction of Virtual Networks

A virtual sensor network is a group of statistically related physical sensors where some of the signals are taken as input to produce others employing a regression model. Possible input combinations for the estimation of a parameter P_n from three inputs S_1, S_2 and S_3 are shown in figures 4 (a), 4 (b) and 4 (c) . Where the subscripts i, j and k of input signal ‘S’, can assume any values from the set (1, 2, 3) for generating the same output P_n. Therefore, seven (7) virtual networks can be developed for a system having three inputs and one output. For each virtual sensor network to be operational a regression model is indispensible. Among several regression models, ANN is widely used to map between a set of inputs and a set of targets and is quite robust. An ANN is an information processing system characterized by its architecture, training algorithm and activation function. A two-layer feed-forward network with sigmoid hidden neurons and linear output neurons is mostly used (Fausett, 1994). For ANN training Levenberg-Marquardt back propagation algorithm is generally preferred, in case there is not enough memory, the scaled conjugate gradient back propagation is recommended (Beale, et al., 2012).

3.5. Network Importance Measures

The estimations from a virtual sensor network are subjected to changes depending upon characteristics of underlying regression model and sensors' uncertainties. The characteristics of regression models are beyond the scope of this paper however, a concept of importance measures is introduced here to characterize the influence of sensors' uncertainties on the accuracy of estimations computed from a regression model. It is therefore, quite conceivable that the importance measures will adhere to the characteristics of underlying regression model (Ahmed, 2013). In this study the outputs of virtual sensor networks are produced by using ANN therefore, a mean square error (mse) was used to define the importance measures, since it is the basic measure of neural network performance and is widely used due to its ease of computation and quick optimization (Masters, 1993). The mathematical relationship to calculate 'mse' is given in equation (2).

$$mse = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (2)$$

Where y_i represents the actual value of ith member of output vector and $\hat{y}_i$ is the corresponding value estimated value. Two importance measures, accuracy improvement factor (AIF) and accuracy reduction factor (ARF) have been proposed to characterize the importance of a particular sensor in the virtual network owing to input perturbations. AIF for a particular sensor is to reflect a sensor's importance on network estimations when the signal from that particular sensor is correct and remaining sensors are uncertain. AIF can be computed from equation (3)

$$AIF^{(i)} = \frac{mse^{(i)} - mse^{(n)}}{mse^{(n)}} \qquad (3)$$

Where $mse^{(i)}$ and $mse^{(n)}$ are the values of '*mse*' when only the signal from ith sensor is correct and when signals from all other 'n-1' sensors are uncertain. A lower value (close to zero) of $AIF^{(i)}$ would indicate the higher importance to network estimations and a relatively lesser influence of uncertainties in the remaining members of the network. This means that the particular virtual sensor network would produce accurate results provided the signal at the sensor having smallest AIF is secured.

Another metric to assess the importance of a sensor is ARF which is to represent a sensor importance when it is uncertain and the remaining sensors are correct. ARF can be computed from equation (4)

$$ARF^{(i)} = \frac{mse^{(n)}}{mse^{(i,n-1)}} \qquad (4)$$

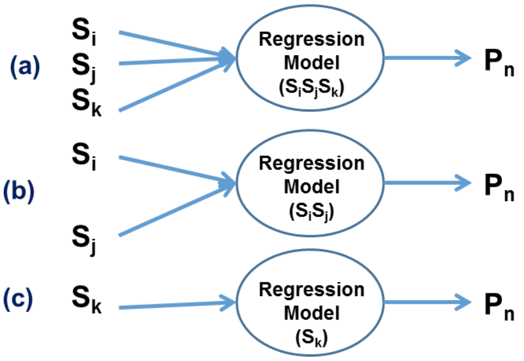


Figure 4: Input combinations for generating similar output parameter from neural networks.

Where $mse^{(i,n-1)}$and $mse^{(n)}$ are the values of *'mse'* when only the signal from ith sensor is corrupted and all other signals from *'n-1'* sensors are correct. The value of ARF for a particular sensor will always lie between zero and one. A value close to zero would indicate higher sensitivity of model estimation to the perturbations in a particular sensor, and a value close to unity would indicate otherwise. This means that the estimations from a particular virtual sensor network are quite sensitive to the perturbations in the signal of sensor having smallest ARF value.

It should be noted that AIF and ARF are based on estimations computed from a regression model and would adhere to the characteristics of that regression model. Therefore, AIF and ARF reflect a sensor's characteristics in a virtual sensor network, which is a group of sensors whose inputs are integrated using a regression model.

4. Analysis and Results

This section describes the application of the presented methodology for developing system-specific relations for containment spray system during a specific LOCA scenario. The LOCA scenario was assumed to follow the sequence of events shown in table 1. We can recognize three important phases of this accident in time, 1) starting from pipe break in the primary system followed by success/failure of several systems till the start of containment sprays at 1,280 sec, 2) time interval during which containment spray system worked and eventually expired at 7,730.2 seconds and recirculation system came into play, and 3) time interval starting with the recirculation system's action to the failure of containment.

The first part within the first 1,280 seconds following the accident was marked by action and failure of several safety features which was quite rapid and therefore, an effort to apply proposed methodology would be quite uncertain and have limited applicability due to many influencing parameters in practical situation.

Table1: Sequence of events following LOCA simulated in MAAP for OPR-1000.

Action	Time (seconds)
PZR heaters on	1.2
PZR heaters off	10.4
HPI on	11.7
Reactor scram	11.7
Motor-driven Auxiliary feed-water on	11.7
MSIV closed	11.7
Main FW off	11.7
Upper compartment spray on	1280.1
Upper compartment sprays off	7730.2
Recirculation system in operation	7730.2
Accumulator water depleted	247447.2
Containment failed	247447.2

The second part (1,280 – 7,730 seconds) where only the containment spray systems was operational, the proposed methodology was implemented and correlations were computed for PAM parameter with the SAMG parameters. For this part, the explanation for reactor vessel level (RVL) is given. For other parameters the strategy can be directly extended. The parameter RVL was found to have correlation with containment (CNMT) water level measuring sensors, radioactive waste storage tank (RWT) level, containment gas temp, pressurizer (PZR) temperature, cold leg temperature, reactor vessel gas temperature, and temperature measuring devices installed in reactor core, at a correlation coefficient higher than 0.85.

The third part (after 7,730 second), was marked by the build-up of pressure in the containment building which was not controlled by recirculation system and eventually led to the containment rupture.

4.1. Estimations of PAM

Since, many virtual sensor networks can be developed depending upon the number of correlated parameters. One representative example for RVL signal recovery in case of original sensor failure is presented here, where candidates for the input were three sensors CNMT water level (m), RWT level (m) and cold leg temperature (K). The variation of input and output parameters is shown in figure 5. The trained neural network was tested against an arbitrary sample taken from the data. The estimation of virtual network against the actual normalized value of RVL is shown in figure 6. The accuracy and sensitivity issues were explored by computing the network importance measures.

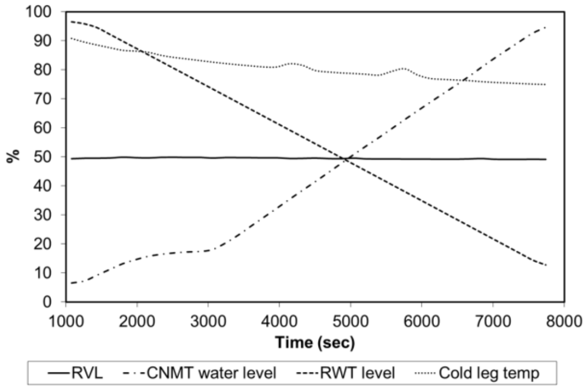


Figure 5: Variation of few correlated parameters with reactor vessel level.

4.2. Comparison of Importance Measures

In our network importance measure calculations, AIF for each sensor was computed by assuming a uniform random noise of ±5% in other members of the network, while during the calculation of ARF the error was considered to be present only in that sensor whose ARF was required. AIF and ARF computed values for the network members are given in table 2. From AIF values the signals CNMT water level, RWT level and cold leg temperature are important to the correct estimations in descending order.

On the basis of ARF values, the network estimations are less sensitive to perturbations (<±5%) to the CNMT water level and more sensitive to the perturbations in RWT level and cold leg temperature.

4.3. Unavailability Problem

A common problem of concern is the unavailability of all of instrumentation or a part of it. Of course, the problem of absolute loss of information cannot be resolved by methods relying upon information, therefore the problem of partial loss of information was considered.

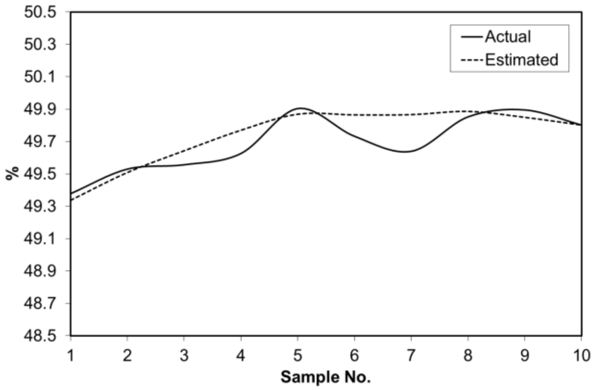


Figure 6: Estimation of RVL using neural network.

Table 2: Network importance measures.

Parameter	AIF	ARF
CNMT water level	41.0685	0.0238
RWT level	71.1913	0.0139
Cold leg temperature	238.0302	0.0043

As mentioned in section 3.4, several virtual networks of varying size and combinations of inputs can be used to estimate the same parameter. A new measure, correlation voting index (CVI) helps to identify faulty sensor and to identify the outputs to be relied upon (Ahmed, 2013). The mathematical form of CVI is given by

$$CVI(i) = \sum_{j=1}^{n} corr(P(i,j)) \qquad (5)$$

In equation (5) *CVI*(*i*) is correlation voting index for ith neural network, P is matrix of estimated outputs from ‘n’ neural networks and ‘*corr*(*P*(*i*,*j*))’is used here to represent function to calculate correlation coefficient between the ith and jth estimation of ‘n’ neural networks stored in matrix P. The small values (especially negative) of CVI indicate the outliers due to their poor correlation with the rest of the estimations. The highest values of CVI represent the set of outputs from networks with lesser uncertainty.

For the RVL estimation, a set of neural networks like shown in figure 4 were developed. Three cases, representing failures of one signal, 1) S_1: CNMT water level, 2) S_2: RWT level and 3) S_3: cold leg temperature, respectively were analyzed. The CVI values for each network for each case are given in table 3. The positive values represent the consistent set of outputs. For instance for case-I (S1 unavailable), the acceptable outputs set are produced by networks S_2, S_3, S_1S_3, and S_2S_3. A unity value 1 for S_1 corresponds to the faulty sensor. The final estimation can be computed either by relying only upon the highest value of CVI, in this case for S_2S_3 network (Ahmed, 2013) or by using the mixing models techniques discussed by (Bishop, 2006).

Table 3: CVI computed values for six neural network outputs.

Neural network	CVI (case I)	CVI (case II)	CVI (case III)
S_1	1.0000	0.7950	0.5826
S_2	1.4255	1.0000	1.2569
S_3	1.5624	0.8183	1.000
S_1S_2	-1.2737	-0.6058	1.2781
S_1S_3	1.0229	0.9360	-0.3763
S_2S_3	2.0604	-0.6208	-0.3605

5. Conclusions

In this research, the statistical relationship among the process parameters has been analyzed. The proposed methodology has been demonstrated for a specific LOCA scenario for the time interval where containment spray system was initiated at a particular instant of accident propagation. Virtual sensor network constructed for the estimation of RVL demonstrates the realization of methodology and its improvement is expected by utilizing other networks and importance measures. The CVI performs to select a set of related outputs and gives a yardstick for comparing results in case exact values are not known. However, to extend this strategy for real power plant application requires the evaluation of system-specific relationships via neural networks at safety system’s operation set-points and for a set of conditions expected to occur at a power plant. There is a need to bring improvements and refinements to the proposed methodology in the areas of parameter grouping and generalization and optimization of neural networks. Also, the neural networks can also be replaced by other regression technique. The importance measures presented in this study can be defined on the basis of any other performance measures for a regression technique however, it should be remembered that these measures represent importance determined by the characteristics of underlying regression technique and not the importance in physical sense.

The application of the proposed methodology has been demonstrated in the aspect of virtual redundancy of a sensor’s information, and unavailability problem. The first would lead to the capability of online validation of critical sensors without installing more physical sensors and the second would provide the capability of estimating critical parameters in case of partial loss of instrumentation along-with the identification of faulty sensors.

Acknowledgement

This work was supported by I-NERI funded by Ministry of Education, Science and Technology (Grant Number: NRF-2012M2A8A2056760).

Nomenclature

AIF	accuracy improvement factor
ANN	artificial neural network
ARF	accuracy reduction factor
CNMT	containment
Corr	function to compute correlation coefficient
CSS	containment spray system
CVI	correlation voting index
FW	feed water
HPI	high pressure injection
HPSIS	high pressure safety injection system
LPSIS	low pressure safety injection system
mse	mean square error

MSIV	main steam isolation valve
PAM	post-accident monitoring
PZR	pressurizer
RVL	reactor vessel level
RWT	radioactive waste storage tank
SA	severe accident
SAM	severe accident management
SAMG	severe accident management guidelines

REFERENCES

Ahmed, R., (2013). Study on Process Parameters to Optimize the Monitoring System for Severe Accidents, PhD Thesis. Yongin-si, Giheung-gu: Kyung Hee University.

Ahmed, R., Heo, G. & An, S. H., (2012). Intelligent Condition-based Maintenance using Plant Health Index. San Diego CA, s.n.

Ahmed, R. et al., (2013). Study on Virtual Redundancy among Process Parameters for Accident Management Applications. Gwangju, s.n.

American Nuclear Society, (2012). Fukushima Daiichi: ANS Committee Resport, s.l.: ANS.

Beale, M. H., Hagan, M. T. & Demuth, H. B., (2012). Neural Network toolbox User's Guide 2012a. s.l.:The MathWorks Inc. .

Bishop, C. M., (2006). Pattern Recognition and Machine Learning. LLC, USA: Springer Science + Business Media.

Fausett, V. L., (1994). Fundamentals of Neural Networks: Architectures, Algorithms And Applications. s.l.:Prentice Hall International.

IAEA, (2002). Accident Analysis of Nuclear Power Plants, Safety Reports Series No. 23, Vienna: International Atomic Energy Agency.

IEEE Power and Energy Society, (2010). IEEE Standard Criteria for Accident Monitoring Instrumentation for Nuclear Power Generating Stations: IEEE Std 497 2010 (Revision of IEEE Std 497-2002. New York: IEEE.

Masters, T., (1993). Practical Neural Network Recipes in C++. San Diego, California: Academic Press, Inc..

Park, K. A. a. S., (2009). Development of a Risk-informed Accident Diagnosis System to Support Severe Accident Management. Nuclear Engineering and Design, Volume 239, pp. 2119-2133.

US Nuclear Regulatory Commission, (1983). Regulatory Guide 1.97: Revision 3, Instrumentation for Light-water Cooled Nuclear Power Plants to Assess Plant and Environs Conditions During and Following an Accident, s.l.: USNRC.

Westinghouse Electric Company, (2012). Nuclear Automation: Advanced Logic System Post-accident Monitoring System. [Online] Available at: www.westinghousenuclear.com
[Accessed May 2013].

BIOGRAPHY

Ahmed, Rizwan was born at Sialkot, Pakistan on January 28th 1983. He graduated from M.Sc. (Geophysics) program in 2004 from Quaid-i-Azam University, Islamabad, Pakistan. He got fellowship in MS Nuclear Engineering at Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad, Pakistan in 2004 and graduated from the program in 2006. He joined Mainformatics Laboratory, Department of Nuclear Engineering at Kyung Hee University (KHU), South Korea in September 2009 and completed his Ph.D. in August 2013. Now, he works as a faculty member at Department of Nuclear Engineering, PIEAS in Pakistan. He has published several journal and conference papers on areas related to nuclear safety problems. His research interests include accident simulation, numerical computing, and PSA. He was awarded with best presentation and paper award by Japanese Society of Mechanical Engineers (JSME) for his publication in ICEM-2010, Japan. He is a member of Korean Nuclear Society (KNS) Korea.

Modeling localized bearing faults using inverse Gaussian mixtures

Pavle Boškoski, Đani Juričić

Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
pavle.boskoski@ijs.si
dani.juricic@ijs.si

Abstract

Localized bearing faults exhibit specific repetitive vibrational patterns. Due to the constant angular distance between the roller elements, the vibrational patterns occur on regular angular intervals. Under constant operating conditions such patterns become easily detectable as "periodic" events. Slippage or small variation in rotational speed are commonly modeled by introducing normally distributed time variations, which allows for occurrence of "negative" time intervals. In this paper we present an approach which models the occurrences of localized bearing fault patterns as a realization of random point process whose inter-event time intervals are governed by inverse Gaussian mixture. Having support on $(0, \infty)$, the random impact times can acquire strictly positive values. The applicability of the model was evaluated on vibrational signals generated by bearing models with localized surface fault.

1. Introduction

Bearing faults are one of the most common causes for mechanical failures (MRWG, 1985; Albrecht, Appiarius, & Shrama, 1986). Consequently, the majority of the proposed fault detection methods address the issue of bearing fault detection. Commonly, the well adopted methods focus on extracting and analyzing the behavior of a set of features that describe bearing surface faults, so-called bearing fault frequencies (Tandon & Choudhury, 1999). Inferring about bearing condition using such a feature set is possible if the monitored bearing is operating under constant rotational speed. However, rotational speed fluctuations, which are quite common in real world, reduce the effectiveness of these features. In this paper we model the vibrational patterns generated by bearings with localized surface fault modeling as a point process with inverse Gaussian mixture inter-event distribution.

From a practical point of view, condition monitoring of bearings operating under variable regimes is the most plausible real world scenario. As a result, recently many authors proposed new approaches for condition monitoring of machinery operating under non-stationary regimes (Zhan, Makis, & Jardine, 2006; Combet & Zimroz, 2009; Wang, Makis, & Yang, 2010; Boškoski & Juričić, 2012a; Cocconcelli, Bassi, Secchi, Fantuzzi, & Rubini, 2012; Boškoski & Juričić, 2012b; Heyns, Godsill, Villiers, & Heyns, 2012). Despite the non-stationarity of the generated vibrations, these approaches manage to exploit the statistical properties of some specific vibrational patterns, hence performing sufficiently accurate condition monitoring. Focusing on bearing fault detection, the main source of information are the time occurrences of particular vibrational patterns. Based on the statistical properties of these time occurrences several effective fault detection methods have been developed (Antoni & Randall, 2003; Borghesani, Ricci, Chatterton, & Pennacchi, 2013). In the same manner we propose an approach that describes the impacts generated by localized bearing surface damage as a realization of a point process whose inter-event times are governed by pure or inverse Gaussian mixture.

Initially, inverse Gaussian distribution was developed by Schrödinger (1915) as the distribution of the first passage time of a Wiener process with positive drift and fixed threshold. The first detailed in-depth analysis of the statistical properties of inverse Gaussian distribution was derived much later by Tweedie (1957) and afterwards by Folks and Chhikara (1978). Since then inverse Gaussian distribution has been applied in many different areas for instance: production modeling (Desmond & Chapman, 1993), reliability (Lemeshko, Lemeshko, Akushkina, Nikulin, & Saaidia, 2010), neural spike train modeling (Vreeswijk, 2010), condition monitoring (Boškoski & Juričić, 2011) etc. Although in many cases the application of pure inverse Gaussian model suffices, in this paper we show that under variable rotational speed inverse Gaussian mixture is more suitable model for describing localized bearing faults.

The paper is organized as following. Section 2 contains the definition and the basic statistical properties of the inverse Gaussian distribution. The selection between models describing pure or mixture of inverse Gaussian distributions is presented in Section 3. The actual modeling of localized bearing

Pavle Boškoski et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

faults with in the proposed framework is presented in Section 4. Finally, the experimental validation of the models is given in Section 5.

2. Pure and mixed inverse Gaussian distributions

2.1. Pure inverse Gaussian distribution

Let a stochastic process $\alpha(t)$ be

$$\alpha(t) = \nu t + \sigma^2 W(t),\ \nu > 0, \tag{1}$$

where ν is the positive drift, σ^2 is the variance and $W(t)$ is Wiener process (Matthews, Ellsworth, & Reasenberg, 2002). Schrödinger (1915) showed that the first passage time of the process (1) over a fixed threshold a follows the Inverse Gaussian distribution (Folks & Chhikara, 1978):

$$\begin{gathered} f(t;\mu,\lambda) = \sqrt{\frac{\lambda}{2\pi t^3}} \exp\left(-\frac{\lambda(t-\mu)^2}{2\mu^2 t}\right), \\ t > 0, \mu = a/\nu > 0, \lambda = a^2/\sigma^2. \end{gathered} \tag{2}$$

Since the parameters μ and λ in (2) are time invariant, the resulting stochastic process is stationary. A simple realization of such a process is shown in Figure 1.

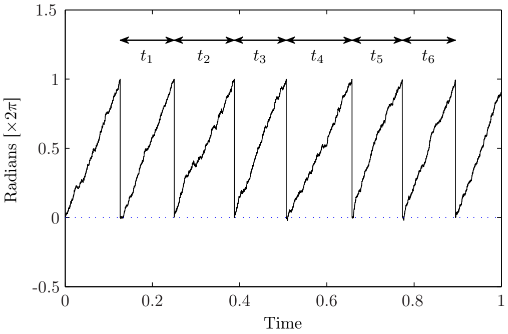


Figure 1. Simulated realization of the stochastic process (1). The time intervals t_i are distributed by inverse Gaussian distribution (2)

2.2. Mixed inverse Gaussian distribution

When modeling data generated by Wiener process (1) there are many situations in which parameters μ and λ in (2) should be considered as random variables. Under such circumstances, the distribution of the first passage time can be described by inverse Gaussian mixtures (Whitmore, 1986). Physically more sound is to allow the positive drift ν in (1) to vary randomly according with some pre-defined distribution. In order to keep the relation with the positive drift ν more clearly visible, Desmond and Chapman (1993) reparametrized (2) by setting $\delta = 1/\mu$:

$$f(t;\delta,\lambda) = \sqrt{\frac{\lambda}{2\pi t^3}} \exp\left(-\frac{\lambda(\delta t-1)^2}{2t}\right), \tag{3}$$

where $t > 0, \delta > 0, \lambda = a^2/\sigma^2$.

In such a form the parameter δ is linearly related to the positive drift ν in (1). By allowing δ to be random variable with distribution $p_\delta(\delta)$, the marginal distribution reads:

$$h(t;\theta) = \int_\triangle f(t;\lambda|\delta) p_\delta(\delta)\, d\delta, \tag{4}$$

where θ is the vector comprising of λ and all hyper parameters of $p_\delta(\delta)$.

3. Model selection

The likelihood functions (2) and (4) specify two different models M_1 and M_2 respectively that can be used for describing the time occurrences t. The selection of which model is more appropriate can be performed by using Bayes' factor.

The application of the Bayes' factor incorporates the concepts of parsimony, unlike the standard likelihood which suffers from the problems of overfitting (MacKay, 2005; Berkes & Fiser, 2011). For the observed data t the Bayes' factor between two models M_1 and M_2 reads:

$$\frac{P(M_1|t)}{P(M_2|t)} = \underbrace{\frac{P(t|M_1)}{P(t|M_2)}}_{Bayes factor} \times \frac{P(M_1)}{P(M_2)}, \tag{5}$$

where $P(M_1)$ and $P(M_2)$ are prior distributions associated with each model.

The two likelihoods entering the Bayes' factor can be calculated by integrating over the complete set of parameters as:

$$\begin{aligned} P(t|M_1) &= \int f(t|\theta_1, M_1) p(\theta_1|M_1)\, d\theta_1 \\ P(t|M_2) &= \int h(t|\theta_2, M_2) p(\theta_2|M_2)\, d\theta_2, \end{aligned} \tag{6}$$

where $f(t|\theta_1)$ is defined by (2), $h(t|\theta_2)$ is defined by (4) and θ_1 and θ_2 are their corresponding parameter sets.

3.1. Specification of the prior $p_\delta(\delta)$

In order to complete the calculation of the Bayes' factor (5), one has to specify the distribution of the random positive drift δ in (4). One possible model of the drift fluctuations, similar to the one specified by Desmond and Yang (2011), reads:

$$\delta = d + \varepsilon,\ \text{where } \varepsilon \sim \mathcal{N}(0, \sigma_\delta^2), d \geq 0, \delta > 0. \tag{7}$$

For cases when the parameter $\sigma_\delta = 0$, the drift parameter δ becomes deterministic, thus the mixture inverse Gaussian (4) reduces into its standard form (2).

The limitation $\delta > 0$ imposes additional limitation on the distribution of ε in (7). Consequently, one has to use Gaussian distribution of ε truncated so that $\varepsilon > -d$.

Using the model (7) with truncated Gaussian distribution as a prior for the speed fluctuations, the marginal likelihood (4) becomes:

$$\begin{aligned}\hat{h}(t;\lambda,\sigma_\delta,d) = &\sqrt{\frac{\lambda}{2\pi t^3(1+\lambda\sigma_\delta^2 t)}}\\ &\times \exp\left(-\frac{\lambda(dt-1)^2}{2t(1+\lambda\sigma_\delta^2 t)}\right)\\ &\times \frac{\Phi\left(\frac{d+\lambda\sigma_\delta^2}{|\sigma_\delta|\sqrt{1+\lambda\sigma_\delta^2 t}}\right)}{\Phi\left(\frac{d}{\sigma_\delta}\right)},\end{aligned} \tag{8}$$

where $\Phi(\cdot)$ is the cumulative function of the standard normal distribution.

The proposed speed model (7) defines random and stationary speed profile. When necessary, an arbitrary speed profile can be used instead. The only problem would be to specify a proper definition of the prior $p_\delta(\delta)$ and calculate new marginal likelihood (8).

Finally, it has to be emphasized that the modeled parameter in (7) is the standard deviation σ_δ instead of the variance. By modeling through the variance an additional limitation will be imposed i.e. $\sigma_\delta^2 \geq 0$. Such a parametrization introduces a limitation since the parameter under null hypothesis $\sigma_\delta^2 = 0$ lies on the limit of the acceptable region. Therefore standard likelihood tests become inapplicable (Lehmann & Casella, 1998, Chapter 5).

4. Bearing fault detection by means of inverse Gaussian models

Bearing faults are surface damages that occur on the bearing elements. Each time when a rolling element passes over the damaged surface, a specific vibrational pattern is generated directly connected to one of the bearings eigenmodes. Usually, under constant operating conditions the generated vibrations are modeled as (Randall, Antoni, & Chobsaard, 2001):

$$x(t) = \sum_i A_i s(t - iT - \tau_i), \tag{9}$$

where A_i is the amplitude of the i^{th} impact, $s(t)$ is the impulse response of the excited eigenmode, T is the period of rotation and τ_i is random fluctuation due to slippage. Generally, τ_i is modeled as zero mean normally distributed with sufficiently small variance σ_τ^2. Regardless of the variance σ_τ^2, model (9) allows for τ_i to acquire sufficiently low negative values. Consequently, the occurrence of the $i+1^{\text{th}}$ impact might be modeled as if it occurs before the i^{th} one.

4.1. Using inverse Gaussian distribution

Avoiding the issues of negative time delays, present in model (9), we propose the following model of generated vibrations:

$$x(t) = \sum_i A_i s(t - t_i), \tag{10}$$

where A_i is the amplitude of the ith impact, $s(t)$ is the impulse response of the excited eigenmode and t_i is the time of the occurrence modeled as inverse Gaussian random variable. A typical vibrational pattern is shown in Figure 2.

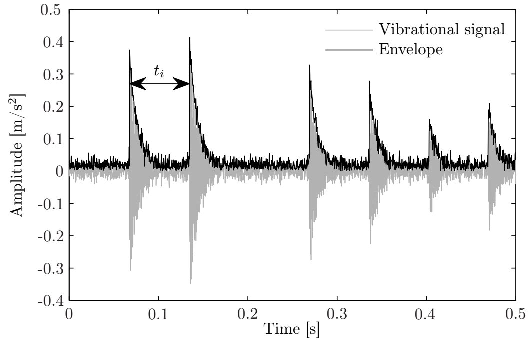


Figure 2. Simulated conceptual vibrational pattern generated by damaged bearing

Due to the mechanical characteristics of the bearings, the angular distance between the adjacent rolling elements is constant. Therefore, the angular distance between two consecutive impacts can be regarded as constant too. So, one can easily apply the stochastic process (1) to model the angular distance traveled by a rolling element towards the damaged surface. The threshold a in (1) is the actual angular distance between the roller elements and ν is directly related to the rotational speed. Consequently, the time intervals t_i between two adjacent excitations of $s(t)$ can be modeled as a realization of either pure or mixture inverse Gaussian, depending on the statistical characteristics of the rotational speed.

Pure inverse Gaussian model (2) for the inter-impact times t_i should be regarded as a special case, valid when the bearing rotational speed is “constant” i.e. there are no significant speed fluctuations. Under such circumstances pure inverse Gaussian model (2) is applicable for localized bearing surface faults (Boškoski & Juričić, 2011).

A more realistic scenario is the one where the rotational speed of a bearing varies randomly. Under such circumstances the angle covered by a rolling element can be modeled as a realization of the stochastic process (1) by allowing the positive drift $\nu \propto \delta$ to vary randomly according to the random speed fluctuations. Consequently, the observed time intervals t_i between two consecutive impacts can be modeled as a realization of an inverse Gaussian mixture (8).

4.2. Multiple localized faults

The case of multiple localized surface faults can be also described in the framework of point processes with inverse Gaussian inter-event distribution. For that purpose one can consider a Wiener process, similar to (1), with two barriers a and b. Starting from an initial point the time required to reach the barrier a is T_1, and time to reach the barrier b from a is T_2. Chhikara and Folks (1989) showed that T_1 and T_2 are independent inverse Gaussian random variables defined as:

$$\begin{aligned} T_1 &\sim IG\left(\frac{a}{\nu}, \frac{a^2}{\sigma^2}\right) \\ T_2 &\sim IG\left(\frac{b-a}{\nu}, \frac{(b-a)^2}{\sigma^2}\right) \end{aligned} \tag{11}$$

Measuring from the initial starting point reaching the threshold b can be specified as time $T_3 = T_1 + T_2$. Since the ratio

$$\frac{\lambda_i}{\mu_i^2} = \frac{\nu^2}{\sigma^2} = \text{const.}, \tag{12}$$

the time T_3 is also inverse Gaussian random variable distributed as (Chhikara & Folks, 1989):

$$T_3 \sim IG\left(\mu_1 + \mu_2, \frac{\nu^2(\mu_1 + \mu_2)^2}{\sigma^2}\right). \tag{13}$$

In the context of bearings, the threshold a is the angular distance of the first fault in the direction of rotation measured from some initial point. The threshold b, on the other hand, is the angular distance measured from the first fault in the direction of rotation.

By extending the concept of two thresholds (13) to multiple thresholds, one can model multiple localized bearing faults by employing the generalized distribution of inter-event times (Chhikara & Folks, 1989, Chapter 11).

5. EXPERIMENTS

The proposed model based on mixture of inverse Gaussian distribution of the inter-event times was evaluated on simulated vibration signals. The signals were generated using the dynamic bearing model developed by Sawalhi, Randall, and Endo (2007) enhanced with the EHL (Elastohydrodynamic Lubrication) model developed by Sopanen and Mikkola (2003a, 2003b). The simulated bearing had localized surface fault on the outer ring. The fault was simulated to be 2° wide and has average surface depth of $30\mu m$.

Simulations were performed using several different speed profiles according to the model (7) with mean value d = 38 Hz. The standard deviation σ_δ changed from 0% up to 10% of the mean speed d.

5.1. Detection of impacts times

The main information required for the application of proposed inverse Gaussian based models are the time intervals between two consecutive impacts. Therefore, the first step in the analysis is the detection of impact times. In our approach, the detection of impact times was performed using wavelet transform thresholding. The main parameter that has to be selected is the mother wavelet. Schukin, Zamaraev, and Schukin (2004) suggested that for signals containing repetitive impulse responses, an optimal detection of impacts can be performed by using mother wavelet that will closely match the underlying vibrational patterns. However, Unser and Tafti (2010) provided thorough analysis that the crucial parameter for sparse wavelet representation of signals containing repetitive impulse responses, is the number of vanishing moments of the mother wavelet rather then the selection of the "optimal" mother wavelet that will closely match the underlying signal. Therefore, by selecting a wavelet with sufficiently high number of vanishing moments one can sufficiently accurately analyze vibrational patterns containing the impulse responses from the excited eignemodes regardless of their variable form due to the changes of the transmission path. The schematic representation of the impact detection process is shown in Figure 3.

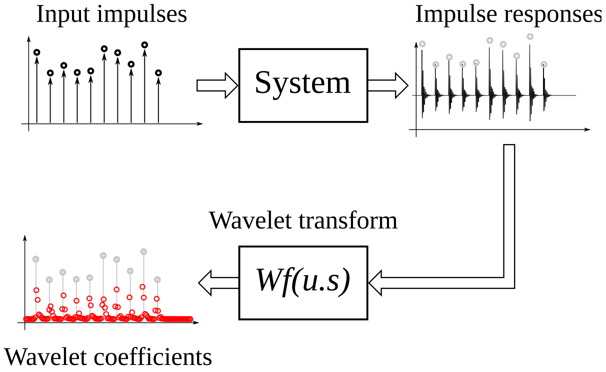


Figure 3. Detection of impact times using wavelet as differential operator

In our approach, the generated vibrations were analyzed using Daubechies 10 mother wavelet (Daubechies, 1992). For our particular system such a number of vanishing moments has shown to be sufficient for accurate impulse detection.

5.2. Numerical calculation of the Bayes' factor

Having the impact times t_i the next step is to calculate the Bayes' factor by calculating the marginal distributions (6). The marginal likelihoods were calculated using Monte Carlo integration. Since the model selection depends on the standard deviation σ_δ (not the variance σ_δ^2), the selected prior was so-called folded non-central t distribution (Gelman, 2006) which reads:

$$p(\sigma_\delta) \propto \left(1 + \frac{1}{\gamma}\left(\frac{\sigma_\delta}{A}\right)^2\right)^{-(\gamma+1)/2}, \qquad (14)$$

where A is scale parameter and γ represents the degrees of freedom. The prior for the mean value d in (7) was chosen to be uniform in sufficiently wide interval. The prior for the remaining parameter $\lambda = 1/\sigma^2$ in (1) was also chosen to be uniform in the interval that contains 2% of initial speed fluctuations due to slippage (Randall et al., 2001).

5.3. Experimental results

One realization of the speed fluctuations, modeled according to (7) with $d = 38$ Hz, is shown in Figure 4. The speed fluctuations are smooth but sufficiently fast. Consequently even during a single bearing revolution the rotational speed varies.

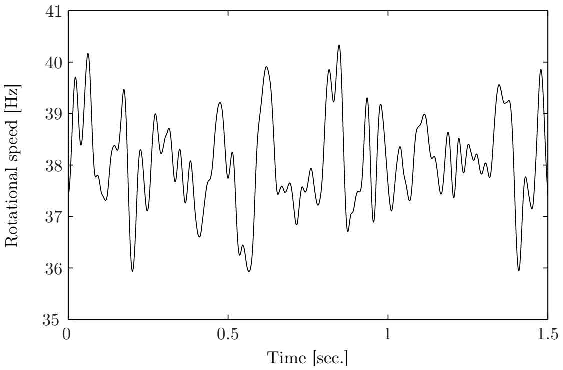


Figure 4. A typical speed fluctuation profile

For small speed deviations $\sigma_\delta < 0.5\%$ of the mean speed value d, the Bayes' factors (6) overwhelmingly favor simpler model (2) i.e. pure inverse Gaussian distribution of the inter-event times. For speed fluctuations with $\sigma_\delta > 0.5\%$ the Bayes' factors favor mixture inverse Gaussian model for the inter-event times. Changes of the Bayes' factor with respect to the changes in the speed fluctuations σ_δ are shown in Figure 5.

Such results are somewhat expected since under small speed fluctuations pure inverse Gaussian distribution of the inter-event times sufficiently well describes the observed impact times. At the same time, due to the principle of parsimony, the simpler model is preferred. The cost of more complex model becomes justified when the speed fluctuations become more intense.

5.4. Comments on results

The effectiveness of the proposed approach becomes apparent if one compares it with other methods. Due to the random speed fluctuations, the standard spectral methods are inapplicable and the only choice is time-frequency analysis of the signal. Therefore, we calculated the wavelet transform of the envelope of the generated vibrations, which is shown in Figure 6. One can easily notice that the envelope contains some patterns in the vicinity of 90 Hz. However, the patterns exhibit no particular structure and it is quite difficult to draw any conclusions from such a time-frequency plot.

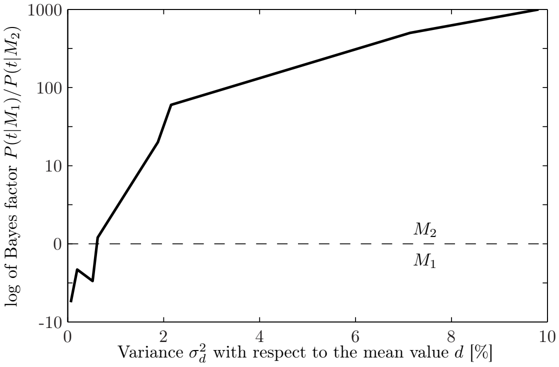


Figure 5. Changes of the Bayes' factors for different σ_δ

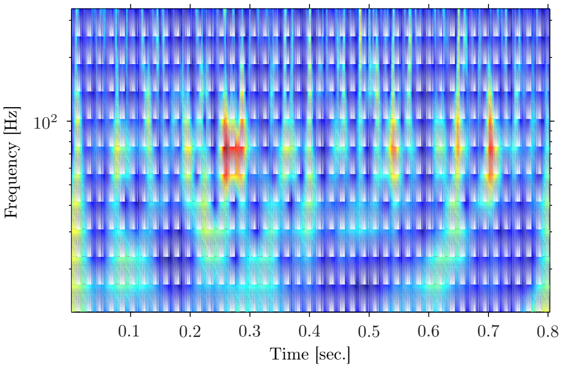


Figure 6. Wavelet transform of the envelope of the generated vibrations

The analysis of the impacts as a realization of a point process with pure or inverse Gaussian mixture offers a framework for proper statistical testing about the origin of the observed events. Testing whether the observed impacts are related to a specific angular position is fairly straightforward. Furthermore, the same analysis offers an insight about the possible mixing distribution, i.e. the distribution of the variable rotational speed.

6. Conclusion

The experimental results show that the specific vibrational patterns generated by bearings with surface faults can be treated as a realization of a point process whose inter-event times are distributed according to either pure or inverse Gaussian mixture. The pure inverse Gaussian distribution is applicable for the special case when fault bearings operate under constant rotational speed. The inverse Gaussian mixture, on the other hand, is a general solution applicable also for modeling the inter-impact times of faulty bearings operating under variable rotational speed. Finally, unlike the commonly adopted models for bearing vibrations, the proposed model is inline with the physical limitations by modeling random time fluctuations with distribution with support on interval $(0, \infty)$.

The application of the proposed approach on acquired vibrations starts by calculating the time intervals between adjacent impacts through the wavelet coefficients calculated from the generated vibration signals. When the observed impacts are generated by a phenomenon that occurs on regular angular intervals, the corresponding inverse Gaussian model can be employed. Determining the validity of such a claim can be performed by a straightforward calculation of the Bayes' factors. This approach is applicable to both constant and variable operating conditions.

Acknowledgment

We like to acknowledge the support of the Slovenian Research Agency through Research Programmes P2-0001, L2-4160 and the Competence Centre for Advanced Control Technologies. The Competence Centre for Advanced Control Technologies is partly financed by the Republic of Slovenia, Ministry of Education, Science, Culture and Sport, and European Union (EU)-European Regional Development Fund.

References

Albrecht, P. F., Appiarius, J. C., & Shrama, D. K. (1986). Assesment of the reliability of motors in utility applications. *IEEE Transactions of Energy Conversion*, *EC-1*, 39-46.

Antoni, J., & Randall, R. B. (2003). A stochastic model for simulation and diagnostics of rolling element bearings with localized faults. *Journal of Vibration and Acoustics*, *125*(3), 282-289.

Berkes, P., & Fiser, J. (2011). A frequentist two-sample test based on Bayesian model selection. *ArXiv e-prints*.

Borghesani, P., Ricci, R., Chatterton, S., & Pennacchi, P. (2013). Diagnostic of rolling element bearings with envelope analysis in non-stationary conditions. In *Condition Monitoring of Machinery in Non-Stationary Operations.*

Boškoski, P., & Juričić, Đ. (2011, September). Point processes for bearing fault detection under non-stationary operating conditions. In *Annual Conference of the Prognostics and Health Management Society* (pp. 427–434). Montreal, QC, Canada.

Boškoski, P., & Juričić, Đ. (2012a). Fault detection of mechanical drives under variable operating conditions based on wavelet packet Rényi entropy signatures. *Mechanical Systems and Signal Processing*, *31*, 369—381.

Boškoski, P., & Juričić, Đ. (2012b). Rényi Entropy Based Statistical Complexity Analysis for Gear Fault Prognostics under Variable Load. In T. Fakhfakh, W. Bartelmus, F. Chaari, R. Zimroz, & M. Haddar (Eds.), *Condition Monitoring of Machinery in Non-Stationary Operations* (p. 25-32). Springer Berlin Heidelberg.

Chhikara, R., & Folks, J. L. (1989). *The Inverse Gaussian Distribution, Theory, Methodology, and Applications* (Vol. 59). New York: Marcel Dekker.

Cocconcelli, M., Bassi, L., Secchi, C., Fantuzzi, C., & Rubini, R. (2012). An algorithm to diagnose ball bearing faults in servomotors running arbitrary motion profiles. *Mechanical Systems and Signal Processing*, *27*(0), 667 - 682.

Combet, F., & Zimroz, R. (2009). A new method for the estimation of the instantaneous speed relative fluctuation in a vibration signal based on the short time scale transform. *Mechanical Systems and Signal Processing*, *23*(4), 1382 - 1397.

Daubechies, I. (1992). *Ten Lectures on Wavelets*. Philadelphia: SIAM.

Desmond, A. F., & Chapman, G. R. (1993). Modelling Task Completion Data with Inverse Gaussian Mixtures. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *42*(4), 603–613.

Desmond, A. F., & Yang, Z. L. (2011). Score tests for inverse Gaussian mixtures. *Applied Stochastic Models in Business and Industry*, *27*(6), 633–648. Available from `http://dx.doi.org/10.1002/asmb.876`

Folks, J. L., & Chhikara, R. S. (1978). The Inverse Gaussian Distribution and Its Statistical Application–A Review. *Journal of the Royal Statistical Society. Series B (Methodological)*, *40*(3), 263–289.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, *1*(3), 515—533.

Heyns, T., Godsill, S., Villiers, J. de, & Heyns, P. (2012). Statistical gear health analysis which is robust to fluctuating loads and operating speeds. *Mechanical Systems and Signal Processing*, *27*(0), 651 - 666.

Lehmann, E., & Casella, G. (1998). *Theory of Point Estimation* (2nd ed.). New York: Springer – Verlag.

Lemeshko, B. Y., Lemeshko, S. B., Akushkina, K. A., Nikulin, M. S., & Saaidia, N. (2010). Mathematical and Statistical Models and Methods in Reliability: Applications to Medicine, Finance, and Quality Contro. In V. Rykov, N. Balakrishnan, & M. Nikulin (Eds.), (1st ed., pp. 433–453). Birkhäuser Boston.

MacKay, D. J. C. (2005). *Information Theory, Inference, and Learning Algorithms* (4th ed.). Cambridge University Press.

Matthews, M. V., Ellsworth, W. L., & Reasenberg, P. A. (2002). A Brownian Model for Recurrent Earthquakes. *Bulletin of the Seismological Society of America*, *92*(6), 2233–2250.

MRWG, M. reliability working group. (1985). Report of Large Motor Reliability Survey of Industrial and Commercial Installations, Part III. *IEEE Transactions of Industry Applications*, *IA-23*, 153-158.

Randall, R. B., Antoni, J., & Chobsaard, S. (2001). The relationship between spectral correlation and envelope analysis in the diagnostics of bearing faults and other cyclostationary machine signals. *Mechanical Systems and Signal Processing*, *15*, 945 - 962.

Sawalhi, N., Randall, R., & Endo, H. (2007). The enhancement of fault detection and diagnosis in rolling element bearings using minimum entropy deconvolution combined with spectral kurtosis. *Mechanical Systems and Signal Processing*, *21*, 2616-2633.

Schrödinger, E. (1915). Zur Theorie der Fall- und Steigversuche an Teilchen mit Brownscher Bewegung. *Physikalische Zeitschrift*, *16*, 289–295.

Schukin, E., Zamaraev, R., & Schukin, L. (2004). The optimisation of wavelet transform for the impulse analysis in vibration signals. *Mechanical Systems and Signal Processing*, *18*(6), 1315—1333.

Sopanen, J., & Mikkola, A. (2003a). Dynamic model of a deep-groove ball bearing including localized and distributed defects. Part 1: Theory. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, *217*(3), 201–211.

Sopanen, J., & Mikkola, A. (2003b). Dynamic model of a deep-groove ball bearing including localized and distributed defects. Part 2: Implementation and results. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, *217*(3), 213–223.

Tandon, N., & Choudhury, A. (1999). A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. *Tribology International*, *32*, 469-480.

Tweedie, M. C. K. (1957). Statistical Properties of Inverse Gaussian Distributions. I. *The Annals of Mathematical Statistics*, *28*(2), 362–377.

Unser, M., & Tafti, P. (2010). Stochastic models for sparse and piecewise-smooth signals. *IEEE Transactions on Signal Processing*, *PP*(99), 1–1.

Vreeswijk, C. van. (2010). Analysis of Parallel Spike Trains. In S. Grün & S. Rotter (Eds.), (Vol. 7, pp. 3–20). Springer.

Wang, X., Makis, V., & Yang, M. (2010). A wavelet approach to fault diagnosis of a gearbox under varying load conditions. *Journal of Sound and Vibration*, *329*(9), 1570 - 1585.

Whitmore, G. A. (1986). Normal-Gamma Mixtures of Inverse Gaussian Distributions. *Scandinavian Journal of Statistics*, *13*(3), 211–220.

Zhan, Y., Makis, V., & Jardine, A. K. (2006). Adaptive state detection of gearboxes under varying load conditions based on parametric modelling. *Mechanical Systems and Signal Processing*, *20*(1), 188–221.

A Cryogenic Fluid System Simulation in Support of Integrated Systems Health Management

John P. Barber[1], Kyle B. Johnston[2], and Matthew Daigle[3]

[1]*SGT, Inc., Kennedy Space Center, FL. 32899, USA*
john.p.barber@nasa.gov

[2]*ENSCO, Inc., ASE, Kennedy Space Center, FL. 32899, USA*
kyle.b.johnston@nasa.gov

[3]*NASA Ames Research Center, Moffett Field, CA. 94035, USA*
matthew.j.daigle@nasa.gov

ABSTRACT

Simulations serve as important tools throughout the design and operation of engineering systems. In the context of systems health management, simulations serve many uses. For one, the underlying physical models can be used by model-based health management tools to develop diagnostic and prognostic models. These simulations should incorporate both nominal and faulty behavior with the ability to inject various faults into the system. Such simulations can therefore be used for operator training, for both nominal and faulty situations, as well as for developing and prototyping health management algorithms. In this paper, we describe a methodology for building such simulations. We discuss the design decisions and tools used to build a simulation of a cryogenic fluid test bed, and how it serves as a core technology for systems health management development and maturation.

1. INTRODUCTION

In modern systems engineering practices, modeling and simulation serve as foundational elements throughout the design process. Systems health management (SHM) technologies, which focus on monitoring system behavior, detecting faults and other anomalies, isolating and identifying faults, and predicting component failures and other significant events, all rely on some type of system model. System simulations capable of modeling both nominal and faulty behavior can help in developing these models and in testing and validating SHM algorithms, and have an additional application for operator training with failure scenarios.

Simulations can effectively serve as virtual testbeds. For development and validation of SHM algorithms, such simulation testbeds are extremely useful since validation requires injecting faults, which is often difficult, costly, or unsafe to perform on real systems. In (Poll et al., 2007) an electrical power distribution system testbed and its corresponding simulation testbed are described. In (Balaban et al., 2013) a simulation testbed for a planetary rover is described. In (Goodrich et al., 2009) a simulation testbed for a cryogenic fluid system is discussed. Each of these simulation testbeds have the ability to inject faults and are used for development and prototyping of health management algorithms. Other examples of simulation-based SHM include (Agusmian, 2013) and (Biswas, 2007).

We describe in this paper the development of another simulation package for a cryogenic fluid system, extending in many respects the preliminary work presented in (Goodrich et al., 2009). The simulation is being developed for a cryogenic testbed (CTB) that, through a network of pipes, valves, pumps, and filters, transfers liquid nitrogen from a storage tank to an external tank representing that of a space vehicle. The purpose of the CTB is to mature SHM technologies for ground systems operations. Developing a simulation for this system presents many challenges, due to the large number of components, the large number of possible system modes, and complex two-phase physics.

This paper focuses on the development of the CTB simulation software, named CryoSim. We discuss the tools used to build the simulation model, and how the challenges of building such a simulation are addressed. The system architecture used for CryoSim is both model and domain agnostic. It can be easily adapted for use with other system models and simulation domains, thus serving as a general architecture for designing virtual testbeds for SHM purposes.

Barber et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

In Section 2, we present a brief overview of the SHM system and other factors that motivated the development of CryoSim. In Section 3, we discuss the internal architecture of CryoSim and present the design methodology used to develop the model. Section 4 details the external interfaces we have developed for CryoSim to facilitate its use as a virtual testbed. We present a number of results in Section 5 to illustrate the key features of CryoSim and its use in an SHM context. Section 6 concludes the paper.

2. Background

CryoSim is one element of an integrated SHM system being developed for the CTB. The SHM system architecture consists of a set of health management tools connected via a message bus based on a publish/subscribe protocol. During operation, the target system periodically publishes a set of messages containing system measurement data (pressures, temperatures, flow rates, etc.). The health management tools receive this data, perform analysis, and then publish messages indicating the health status of the monitored system.

CryoSim was developed as a drop-in replacement data source for the CTB. CryoSim uses a simulation to produce data which accurately represents a range of system behaviors, including fault scenarios. This data is then published to the message bus using the same protocols as the CTB. In this manner, CryoSim acts as a virtual testbed, enabling the development and testing of a suite of health management tools without large numbers of costly test runs on the CTB.

The architecture and implementation of CryoSim described in this paper were primarily motivated by its intended use as a testbed for SHM systems. This application drove the development of features such as component-level fault simulation and the message bus interface, which we discuss in later sections. The possibility of using a simulator package like CryoSim as a data source for other systems dictated the modular, model-agnostic design approach that we followed. Additional considerations during development included its potential application as an operator training environment, and the ability to support varying levels of simulation fidelity with the same model. The remainder of this paper describes the methods we used to ensure that CryoSim could meet this set of objectives.

3. CryoSim Architecture

This section describes the architecture of CryoSim, starting with an overview of its modular architecture. We then discuss key elements of the approach we used to develop the system model. Finally, we describe the operation of the initialization and control modules, and how they interact with the model and the external interfaces present in CryoSim.

3.1. Overview

The CryoSim software was developed to meet the following objectives:

- Provide a medium-fidelity, multi-domain system model incorporating cryogenic fluid flow elements, a pneumatic actuation system, and an I/O and control system
- Model both nominal system behavior and the effects of any of a discrete set of failure modes injected at any location in the system
- Publish model input and output signal values to a message bus interface using the same protocols as the CTB system
- Allow user specification of input signals, model parameters and fault injection commands
- Record all simulation data, including inputs, parameter values, outputs and status/warning messages to a file for offline analysis
- Provide an interactive graphical interface allowing full control over the simulation environment, including system inputs and fault injection

In order to support the various use cases and configurations of CryoSim, the package was developed using a modular architecture. The core consists of two required modules: the CTB system model which is implemented as a hierarchical Simulink® model, and an initialization and control module consisting of a set of MATLAB® functions. If desired, the message bus and GUI modules can also be enabled for a simulation run or set of runs, but are not required to access any of the core functionality of CryoSim. Figure 1 shows a block diagram of the CryoSim architecture.

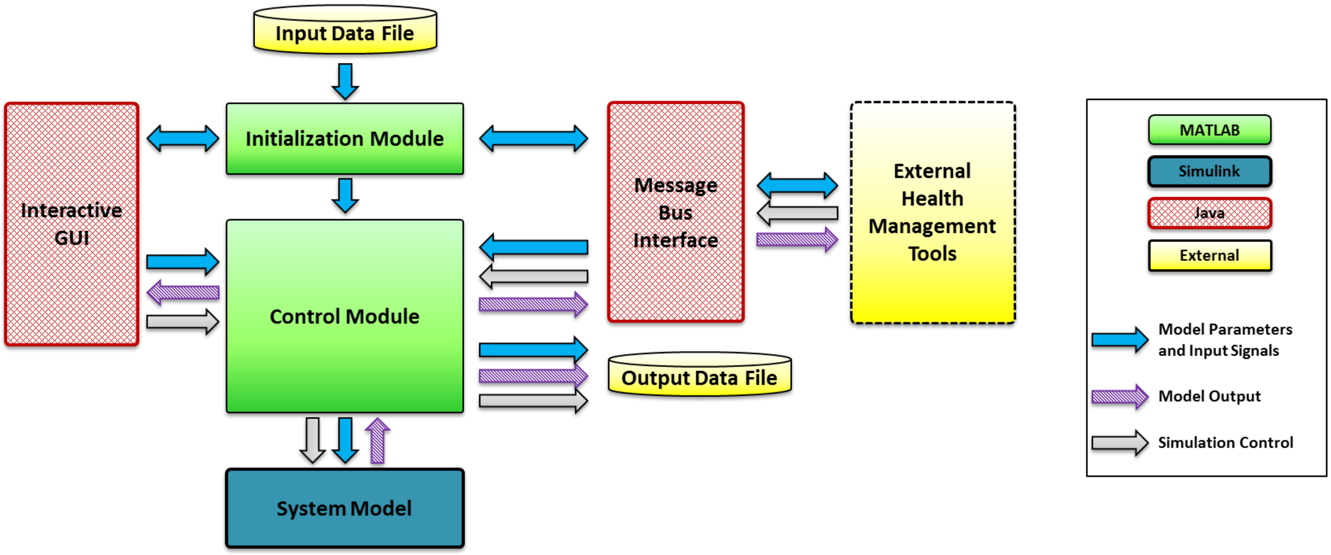


Figure 1. CryoSim block diagram

3.2. System Model

The CryoSim system model is a variable-fidelity, multi-domain physics-based model of the CTB. The model was developed using Simulink, without dependencies on additional toolboxes. We developed a set of component libraries to represent the various physical domains included in the system. Interaction between elements in different domains is incorporated in the component designs. The current implementation has libraries for cryogenic fluid flow, pneumatics, and electrical systems (including transducers and system I/O). Library components are instantiated in the model and connected to match the topology of the physical system. Related sets of interconnected components are organized into subsystems, which are connected together to form the complete model.

3.2.1. Component-based Design

The CryoSim system model is composed of a set of component models which are connected to match the CTB system topology. The component models are parameterized representations of a component's behavior in both nominal and faulty operating regimes. This methodology allows a single component model residing in a library to be used in the system model to represent a number of similar physical components, each having unique physical characteristics and behavior. For example, the library component used to represent a pneumatically actuated control valve, shown in Figure 2, has parameters describing the orifice diameters and flow coefficients of both the fluid flow path and the pneumatic actuator. Instances of this component are used in the model to represent valves with different geometries, simply by changing the parameters used for each component.

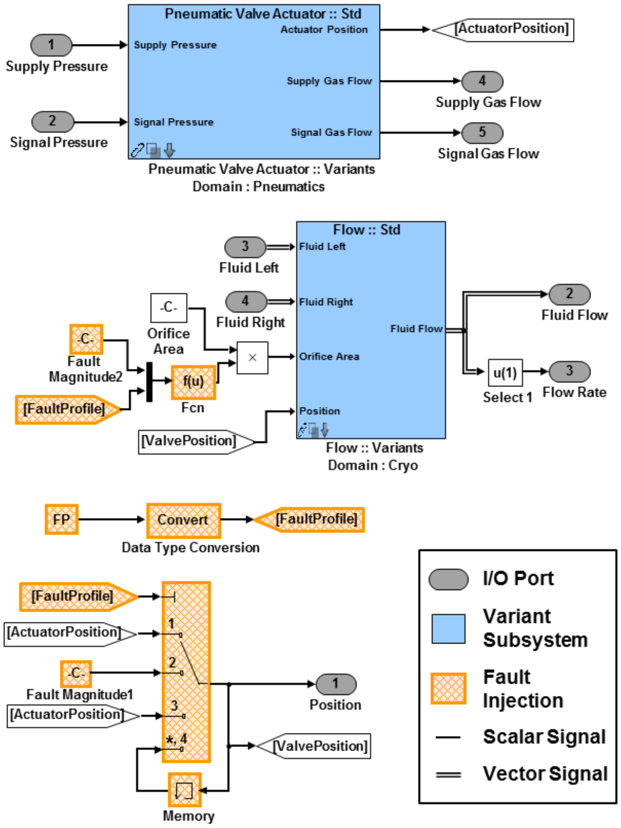


Figure 2. Control valve (CV) component model

3.2.2. Variable Model Fidelity

A core feature of CryoSim is support for different levels of simulation fidelity without requiring a user to make changes to the system model. We accomplish this by implementing multiple component or subcomponent-level models, each providing a different level of fidelity and corresponding computational burden. The models can range from a non-computing element such as a constant output or signal pass-through, to a low-order model based on empirical behavior, to a high-fidelity model based on the underlying physics of the component being modeled. Our approach uses the variant subsystem functionality in Simulink to implement this behavior. When using variant subsystems, each component or subcomponent may have one or more variant instances, each representing a different model of the component's behavior. Before a simulation is run, one of the set of possible variants for each component is selected and made active for the simulation, while the remaining variant instances are disabled. This implementation allows the end user to select the desired simulation fidelity at run time using a single parameter, without the possibility of errors introduced by editing the model.

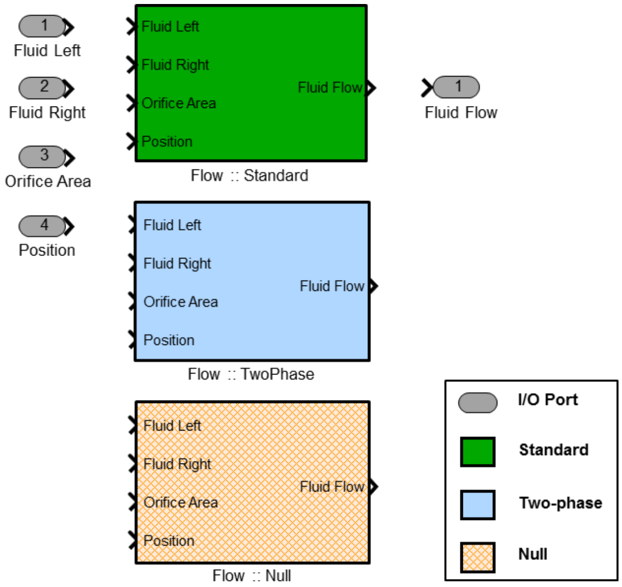


Figure 3. Internal block diagram of a variant subsystem

Figure 3 shows the internal configuration of the variant subsystem used in our model to compute fluid flow, which is representative of the cryogenic fluid domain in our model. The variant subsystem contains three variant instances: a standard fluid flow model, a two-phase fluid flow model, and a null model. The standard fluid model is based on the Bernoulli equation for laminar, incompressible, inviscid flow (Granger, 1995). This option provides a computationally-efficient model which yields usable accuracy for our SHM system when used to simulate the post-chilldown phase of the CTB system's operation, where the flowing fluid exists primarily in the liquid phase. However, it is not accurate for operating regimes such as system chilldown, where the cryogenic fluid is in a mix of liquid and vapor phases.

The two-phase fluid flow model is based on a stratified flow approximation that assumes the gas and liquid are split into two layers with gas on top and liquid on the bottom. The model considers heat transfer with the walls and between the layers, including evaporation/condensation and boiling. The two-phase model provides a much higher level of fidelity, especially in the chilldown phase where there are large temperature variations in the system resulting in significant quantities of liquid being converted to vapor. The tradeoff associated with the two-phase model is increased computation time for higher model fidelity compared to the standard flow model.

The null component variants for the cryogenic fluid library consist of signal terminations on the input ports, and output ports set to constant values. The null variants are effectively empty blocks that require no computation during a simulation. The use of null variants allows unneeded portions of the system model to be disabled for a given simulation run, resulting in significant increases in simulation speed. In general, care must be taken when implementing the null variants in order to provide appropriate boundary conditions for the non-null portion of the system. For example, the null variant for a fluid tank connected to a pipe network should provide output signals representing a static state (pressure, flow, temperature, etc.), rather than null or grounded signals.

It should be noted that the effective use of variant subsystems requires that each variant instance for a given component have the same connectivity. We implement this by using vector-based signals to connect component instances. For example, two scalar signals are needed for the variables representing cryogenic flow in the standard flow model, while four signals are needed for the two-phase model. Similarly, the variables needed to describe a fluid element require either two or seven scalar signals. We combine the groups of scalar signals into vector signals, and use mux/demux blocks at the input and output ports of the variant instances to route the signals internally, as shown in Figure 4. Signals that are not needed by a particular set of variants are grounded or terminated inside of the variant instances. To avoid problems associated with Simulink's ability to propagate signal data types and dimensionality, it is good practice to explicitly specify signal properties at the input and output ports of each component variant. When implemented in this way, changing from one set of variants to another does not require any changes to the model's topology. This allows an end-user to safely change the model's variants and simulation fidelity without the risk of altering connections within the model, and without requiring the model designer to maintain separate system models for each set of variants.

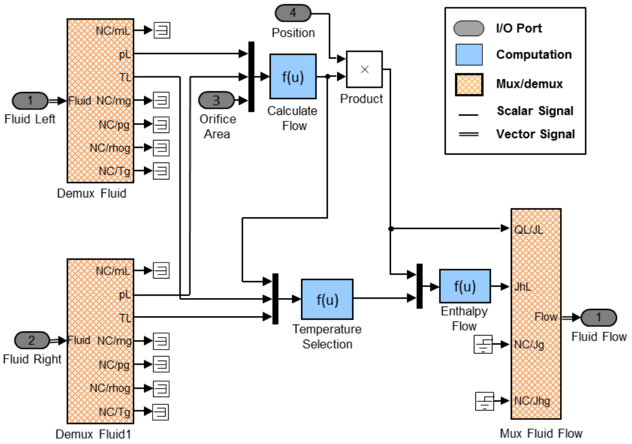


Figure 4. Block diagram for the Standard Fluid Flow variant

The use of variant subsystems easily extends to mixed-domain components. The component model of a pneumatically-actuated control valve shown in Figure 2 includes elements from both the cryogenic fluid flow and pneumatics domains. Because the variant subsystems are implemented as domain-specific sets of variant instances, the control valve model contains two variant subsystems, which can be changed independently.

Our implementation allows the user to specify desired variants via a set of control parameters. For each simulation domain, a global variant parameter determines the particular variant implementation for that domain. Thus, a user desiring a high-fidelity simulation of the cryogenic domain that does not require high fidelity in the pneumatics domain can simply specify the appropriate variant control parameters before running a simulation.

A second group of variant controls in CryoSim allows particular segments of the system to be toggled between the usual domain-specific variant components and null components. If a particular segment is not required for a simulation, the components in that segment can be set to use the null variant instances instead of the normal component models, reducing the computational load required to model the entire system. Implementing this functionality in CryoSim approximately doubles the simulation speed in cases where all unneeded segments are disabled.

3.2.3. Fault Modeling

In order for the simulation data to accurately represent system faults, the desired failure mechanisms must be incorporated into the component designs. During a simulation, when a given fault is injected onto a particular component, the effects of the fault will automatically propagate through to the entire system in accordance with the underlying model behavior. The fault mode, magnitude and injection time are implemented as parameters for each component, enabling fine-grained control over fault behavior in the model.

Faults can be injected or cleared either before a simulation is run, or during the run.

In keeping with our system model's design, the implementation of a specific fault mode will depend on the desired level of fidelity. For many of our component models, a multiport switch is introduced in the path of a signal of interest, allowing different transformations to be applied to the signal depending on the selected fault mode. An advantage of this approach is that new fault modes can be added to a component without requiring any rework of the model or other library components. Similarly, a higher fidelity representation of a particular fault can be incorporated into a component if there is a specific need.

The control valve component model, shown in Figure 2, illustrates this approach to fault modeling. The blocks used to implement the fault modes are shown with a hatched background. This component model has four available fault modes:

1. Nominal behavior: when this mode is active, the actuator position calculated by the "Pneumatic Valve Actuator :: Variants" block is passed without modification to the "Flow :: Variants" block.
2. Stuck pneumatic actuator: in this fault mode, the position of the pneumatic valve actuator is set to a fixed value, regardless of the value of the controlling "Signal Pressure" input. When this mode is active, the calculated actuator position is ignored. A user-determined fault magnitude parameter is used instead, originating from the "Fault Magnitude1" block shown in the diagram.
3. Blockage: this fault mode models an obstruction in the fluid flow path of the valve. The pneumatic actuator position is not affected by this fault mode, so it is passed through to the "Flow :: Variants" block. To model the effects of the blockage, the nominal valve orifice area is scaled down by multiplication with the user-determined fault magnitude parameter.
4. Frozen: this fault mode represents a condition where the pneumatic actuator does not respond to its controlling input signal, similar to the "Stuck" fault described above. However, in the "Frozen" mode, the actuator position is held to its value immediately prior to the fault mode becoming active. This can be seen in Figure 2 as the "Memory" block at the bottom of the diagram.

As noted earlier, the approach we have used for fault modeling allows the addition or modification of fault modes for a given component with no impact to the normal behavior of the component or overall system. The complexity of any particular fault mode's implementation is determined by the model designer. A simple low-order approximation can be used for faults that do not require high-fidelity modeling, such as the "Stuck" and "Frozen" fault modes for the CV component. For these faults, we do not model the mechanics of a failure within any particular type of pneumatic actuator. Instead, we approximate the behavior of the actuator in a manner that minimizes the computational resources needed for the fault models. For fault modes where increased fidelity is desired, the fault can be incorporated into the physics of the affected subsystem, such as the “Blockage” fault for the CV component and the statistical wear model used for the filter component described in Section 5.4.

3.2.4. Extensibility

It is anticipated that CryoSim will be required to provide simulation data representing different configurations of the CTB system. For example, a valve replacement or rerouting of pipes in some segment of the system would constitute a modified configuration that would require corresponding changes to the model. The component-based design is well-suited to this requirement, as changes to a particular subsystem can be made by adding or removing that subsystem’s components and connecting them to match the new system topology.

Additionally, the component libraries allow the rapid creation of models of other systems that utilize the same component types. The modular nature of the CryoSim system architecture, which separates the model from the initialization, control and external interfaces, allows virtually all of the supporting code to be reused for a new system model without modification.

3.3. Initialization Module

The initialization module is responsible for setting up the simulation environment and external interfaces before a simulation is run. To support the multiple interfaces and use cases for CryoSim, the initialization module must provide a number of entry points while enforcing consistent behavior throughout the simulation process. That is, a simulation controlled through the GUI must accept the same inputs and provide the same outputs as a standalone simulation run or one controlled via the message bus. Additionally, the initialization module must validate user-supplied input signals and parameters, and ensure that the set of parameters and signals presented to the simulation module is complete and well-defined. CryoSim uses the base MATLAB workspace to store and process initialization and simulation data, allowing both the initialization and control modules to interact with the system model (via internal MATLAB/Simulink system calls), and the external interfaces (via MATLAB/Java interaction, described later).

3.3.1. Input Data

To ensure that all signals and parameters are defined prior to a simulation run, the initialization module loads a default configuration file, which contains all of the required values. If desired, a user-defined input file can be loaded after the default data file. The initialization module first validates the user-specified data for type, range and dimensionality. The validated data is then merged into the default configuration, with user-specified values always taking precedence over the default values. To avoid unexpected simulation results, the user is notified if the data in their input file is either incomplete or contains invalid entries.

3.3.2. Batch Mode

One important use-case of the CryoSim package is running sets of simulations to generate data for parametric and Monte Carlo analysis. To spare a user from the effort needed to generate unique input files for each set of desired parameter values, the initialization module includes an interface to allow a user-supplied calling function to execute the full initialization procedure once and then run a set of simulations. For each simulation run within the set, the calling function passes an arbitrary set of parameter and signal values which are used to override the default values for that run. This enables a simple user-supplied script to run a batch of simulations with a unique combination of parameter and signal values for each run.

3.4. Control Module

The control module is implemented as an “Interpreted MATLAB function” block inside of the Simulink system model. This function is executed at a predetermined rate as part of the simulation process, and has access to the model’s input and output signal values during the simulation. Additionally, because it executes as a MATLAB function, the control function has access to both the base MATLAB workspace and the external GUI and message bus Java interfaces.

During each iteration of the control function, new values for the model’s control signals and fault status are read from tables in the base MATLAB workspace. The module then queries the GUI and message bus interfaces (if present) for any user-generated control signals or fault commands, which are merged into the default tables. Values received from these interfaces always take precedence over the default values generated by the initialization module using the input file. The updated control signals and faults are then sent to the model, and the model’s current input and output values are published to the GUI and message bus interfaces.

Another important function of the control module is the ability to control the real-world execution speed of the simulation by comparing the simulation clock to a system clock, and introducing appropriate delays if necessary. This functionality allows the CryoSim module to act as a substitute data source for the actual CTB system, which publishes measurements and system input values at one second intervals. Correct behavior in this mode requires that the host computer can execute the simulation at a rate of at least one simulation time step per real-world time step. For simulation runs used for offline data generation and model development, the execution speed control can be disabled, allowing the simulation to run as fast as possible in a given computing environment.

Upon termination of the model's execution, a cleanup function is executed. This function creates an output file which can be used for offline analysis and to meet data retention requirements. The output file contains all of the data needed to reproduce the simulation run (model parameters, input signals and injected faults), as well as the simulation outputs and message log generated by the CryoSim module. With the exception of the simulation outputs and message log, the data structure of the output file is identical to the input file. Thus, a particular simulation can be re-run simply by stripping these tables from an output file, then using it as the input file for a new simulation.

4. External Interfaces

4.1. Overview

We have developed two external interfaces to integrate CryoSim into our SHM system architecture. First, we created an interactive GUI that enables full control of the simulation, from starting and stopping a simulation, to injecting faults and changing model parameters and signals. The second interface consists of an adapter used to connect to the message bus interface used by the SHM system. This adapter allows CryoSim to publish simulation data onto the bus in a manner identical to the physical system it models, functioning as a virtual test bed for the SHM system.

Both interfaces were implemented in Java®, making use of MATLAB's ability to directly access Java objects and methods. The use of Java provides two key advantages. First, by developing Java classes to handle the computations needed for the GUI and message bus, we reduce the computational burden on MATLAB, enabling faster simulation speeds. This improvement in performance is primarily due to the greater control over threading available in Java. The second advantage is the availability of commercial GUI toolkits for industrial controls and systems. We used the GLG Toolkit (www.genlogic.com) for CryoSim, which minimized the effort needed to produce the GUI.

4.2. Graphical User Interface

The CryoSim GUI was designed to meet the following objectives:

- The GUI can control simulation execution, including start/stop/pause commands and control of execution speed.
- The GUI displays the current values of the simulation's output signals, and also includes the ability to produce time-series plots of past values of these signals.
- The GUI allows the user to specify input values (control signals), inject faults and modify other simulation parameters, both before a simulation run and during its execution.
- The GUI has two operational modes: an interactive mode which runs a simulation and generates new output data, and a playback mode which replays data recorded from a prior simulation run, and thus does not require the use of the MATLAB/Simulink software.

4.2.1. Java-MATLAB Interface

As shown in Figure 1, the interface between CryoSim and the GUI includes a change from the MATLAB environment used in CryoSim to a Java-based GUI. To enable full interactivity, user inputs to the GUI must be passed to the CryoSim initialization and control modules, and model outputs must be sent from the control module to the GUI, all without stalling or otherwise interrupting the simulation. As noted above, the use of Java for the GUI's internal computations provides significantly more control over the threading and scheduling of these computations, compared to the single-threaded Simulink environment of the simulation. This is important because the CryoSim control module is part of the Simulink model, and any blocking or delay due to interaction with the GUI has the potential to significantly reduce simulation speed.

Communication between MATLAB and Java can be implemented in MATLAB via the built-in javaMethod() functionality, and in Java using the third-party matlabcontrol API (http://code.google.com/p/matlabcontrol). MATLAB access to Java objects and methods is well-documented and supported by Mathworks, Inc., and serves as the basis of most of our interface. On the other hand, Java access to the MATLAB environment via matlabcontrol makes use of an undocumented interface, although some information is available through third parties (Altman, 2013 and Altman, 2011). Additionally, the matlabcontrol API provides only a limited feature set compared to the use of javaMethod(), providing further weight to our decision to use javaMethod() calls whenever possible.

As shown in Figure 5, the data flow between the GUI and CryoSim can be classified into initialization procedures, which take place before a simulation run, and interactive control during the simulation run. The data exchanged between the GUI and CryoSim during initialization includes model parameters and input signals as well as a table containing faults to be injected during the simulation. During the initialization stage, CryoSim loads a user-specified input file which provides the parameters and signals used for a simulation run. CryoSim then pushes this data to the GUI, where it is used to initialize the information presented to the user. The user can then modify signals, parameter values, and the fault injection table. When the user has created the desired simulation scenario, the simulation can be started via the simulation controls available in the GUI.

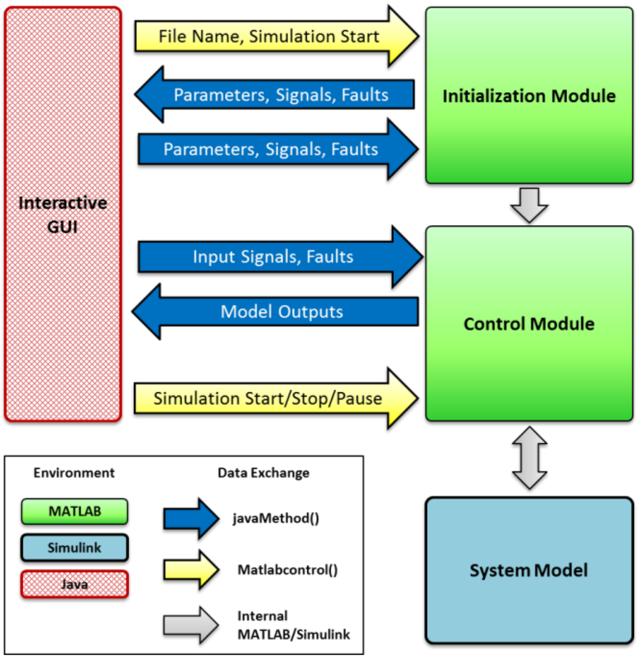


Figure 5. CryoSim-GUI interface

During a simulation, the interaction between CryoSim and the GUI is handled through the use of javaMethod() calls in CryoSim's control module. The control module pushes a vector containing the current output signal values to the GUI, where they are used to update the GUI's display of the model's state. The control module then queries the GUI for any updated input signals or faults that the user might have supplied, and provides them to the model. In this implementation, all user inputs to the GUI are applied to the model at the next time step after the GUI has made them available. This mode of interaction, when coupled with the flexible threading available in Java, allows the simulation to run independently and without risk of blocking from the GUI. The only direct control the GUI has over CryoSim is the set of simulation commands (start, stop and close simulation), which use the matlabcontrol API and can thus execute as soon as activated by the user.

4.2.2. CryoSim-GUI Interaction

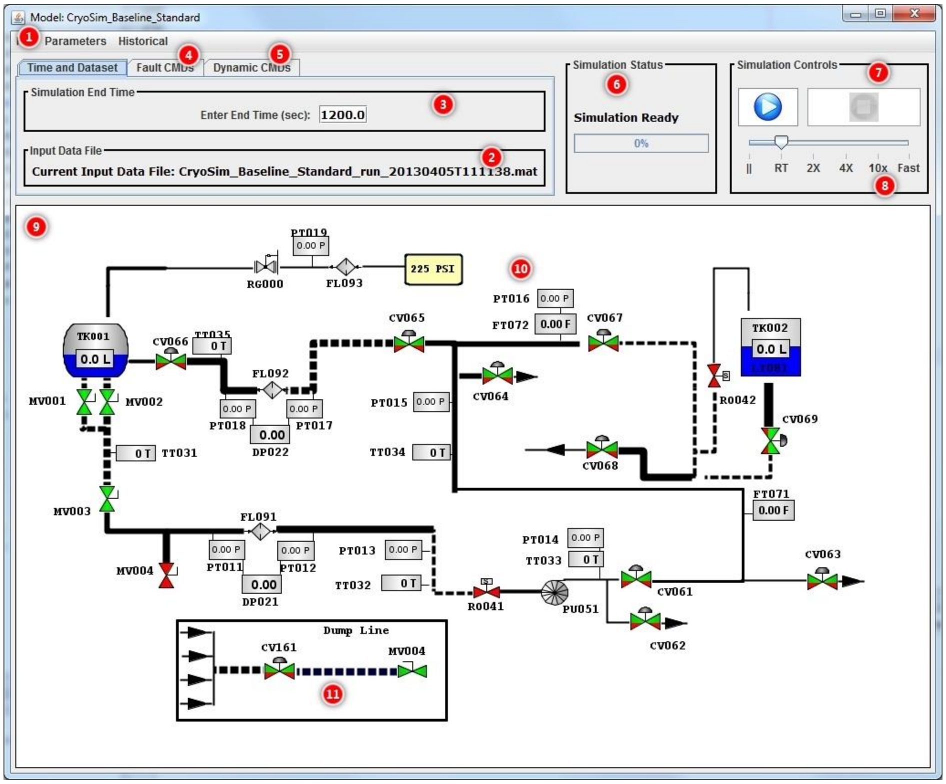
The GUI provides full interactive control of CryoSim using the controls shown in Figure 6. This diagram shows the main GUI window, which consists of a number of panels and controls, numbered here for reference in the text. The system shown has been simplified from the full CryoSim model in order to reduce its visual complexity for illustration purposes.

As discussed earlier, CryoSim makes use of an input file to provide parameters, input signals and faults for a particular simulation run. The GUI menu bar (#1) allows the user to select an input file, which is shown in the configuration panel (#2). This panel also allows the user to specify the simulation length (#3), inject faults (#4) and modify input signals (#5). The Simulation Status panel (#6) displays status messages, simulation progress and the simulation clock. The Simulation Controls panel allows the user to start, stop or pause the simulation (#7) and control the simulation speed (#8).

The System Schematic Panel (#9) contains a graphical representation of the system model's components and topology. Sensor components are included in the model in locations corresponding to CTB sensor locations. Their outputs are shown using text boxes in the GUI (#10) and are updated during each iteration of the control module. Additional 'virtual sensor' outputs in the model provide simulation data for locations in which there is no corresponding sensor in the actual system. The GUI makes use of these additional signals to provide more data to the user for exploratory data analysis, allowing for more detailed understanding of the system's behavior.

As mentioned earlier, the GUI graphics were developed using the GLG Toolkit, which provides an interface whereby the visual appearance of an element can dynamically change based on a state variable's value. We use this functionality to display valve positions (indicated by the color of the valve's body, #11), pipe flow rates and system temperatures.

In addition to serving as a visual representation of the model's state, the GUI allows interactive control of the model's inputs and fault injection status. The user can click on the text label for any component and bring up a component detail window. The contents of this window vary depending on the nature of the component, but can include a time-series plot of the component's input and output signals, controls to allow the user to override the default input signals with a user-determined value, and the ability to inject a fault into the component, either immediately or at some future time during the simulation. Additionally, clicking on a valve body in the main GUI window will toggle the valve's position between fully open and fully closed. All user-supplied input signals are implemented as overrides to the signal values contained in the input file. Thus, if a user sets a valve position using either the toggle functionality or the component detail window, the valve will remain under user control for the remainder of the simulation run, rather than respond to any pre-scheduled changes in the input file.



Model: CryoSim_Baseline_Standard

Simulation End Time

Enter End Time (sec): 1200.0

Input Data File

Current Input Data File: CryoSim_Baseline_Standard_run_20130405T111138.mat

Simulation Status

Simulation Ready

0%

Figure 6. CryoSim GUI

4.3. Message Bus Interface

The primary objective of the CryoSim package is to act as a data source substitute for the CTB system. This architecture makes use of a publish/subscribe message bus based on the Internet Communications Engine (Ice), a suite of communications middleware developed by ZeroC (www.zeroc.com). Using Ice, a message bus based on the publish/subscribe paradigm was developed as the core of the integrated health system. In this configuration, a system being monitored will publish sensor readings indicating its current state to the message bus. Health management modules then subscribe to this published data, perform analysis, and publish messages based on this analysis. The same framework was used in (Poll et al., 2007) and (Balaban et al., 2013).

During a simulation run, CryoSim publishes the values of the model's output signals, which represent all of the available sensors and test points of the CTB system. By controlling the simulation's execution speed to match real-world time, the data published by CryoSim matches the real-world CTB data in timing and format. This enables developers of health management modules to use CryoSim as a virtual testbed without modification. Additionally, the publish/subscribe message bus architecture allows remote control over the CryoSim system through the same interface.

Within CryoSim, the message bus interface is implemented as a hybrid MATLAB/Java construct. The Ice software generates a Java class interface containing the user-defined message formats. The message classes are then combined with Java code that implements the necessary publish and subscribe functionality. The CryoSim initialization module then instantiates the Java classes, establishing communications with the message bus via calls to their methods.

5. Results

In this section we present the results of several simulation runs to demonstrate key aspects of CryoSim. We first provide some information on simulation accuracy and speed, followed by an example of component-level fault injection, followed by demonstration of the use of variant subsystems to trade simulation fidelity for speed. Finally, we show an example of a prognostic health management algorithm interacting with the simulation to produce an analysis.

5.1. Simulation Accuracy and Speed

CryoSim uses the Simulink environment for determining the time-varying solution to the set of ordinary differential equations (ODE) that constitute the system model. The choice of solver algorithm and error tolerances affects both simulation speed and accuracy. For our model, the variable step size ode45 solver works well with the "Standard" fluid variants, while the increased stiffness of the two-phase model requires the use of the ode23s solver, also a variable-step algorithm. For the standard model, a typical simulation runs at approximately 3x speed on a 3.3 GHz workstation, although the time required for a given simulation can vary significantly depending on the dynamics of the model inputs. We have obtained acceptable results using the default relative error tolerance of 1e-3, although some integration noise can be seen in the fluid flow rates. This noise can be reduced by lowering the error tolerance at the cost of decreased simulation speed. The two-phase model is still under development, but is expected to run more slowly than the standard model due to the increased complexity of the underlying physics.

We have validated the model against data recorded from the CTB in the post-chilldown state. In this scenario, the simulated pressures matched the measured data with a maximum error of 0.23%, while the fluid flow rate matched the measured rate to better than 0.1%, considerably less than the measurement noise. Actuation times of the pneumatically-controlled valves had lower fidelity, with a mean 10-90% rise-time error of 0.58 seconds in absolute terms, and 17.7% error relative to the observed actuation times. However, the accuracy of the pneumatics domain components is adequate for most of the intended uses of CryoSim, which focus on system behavior in the cryogenic fluid domain.

5.2. Fault Injection

To demonstrate the fault injection capability of CryoSim, we ran a basic simulation with a single fault injected during the run. This example used the standard flow variant rather than the two-phase flow. The model was initialized in a post-chilldown state, where virtually all of the cryogenic fluid is in the liquid phase. Figure 7(a) shows a schematic representation of a part of the system, including the control valve CV201. This valve starts in the fully open position, and at t=30 seconds, a “Stuck” fault is injected with a magnitude of 50%.

Figure 7(b) shows the outputs of four pressure sensors in the model. In the schematic PT134 is shown immediately upstream of the faulty valve, and PT147 is located at the end of the section shown in the schematic. PT112 is not shown, but is located further upstream from the valve, while PT193 is further downstream. When the fault is injected at t=30, the upstream pressures increase slightly, while the downstream pressures experience a more significant decrease due to the increased drop at the valve.

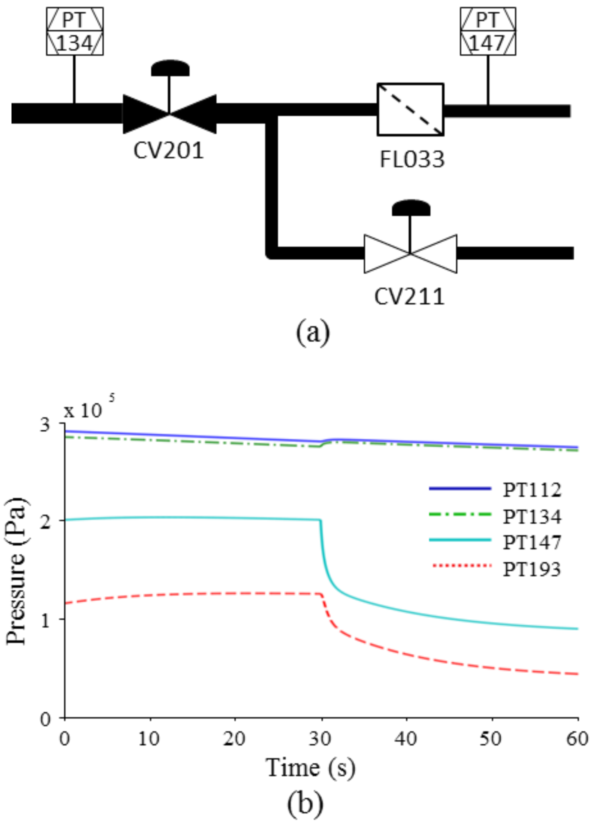


Figure 7. Fault injection example

5.3. Variable Simulation Fidelity

To show the use of variant subsystems to selectively control model fidelity in a specified domain, we simulated the actuation speed of a pneumatically-actuated control valve component, using the “Standard” pneumatics variant which computes gas pressures and flows and uses these values to determine the actuator’s position, and compared this run with the “Basic” pneumatics variant, which replaces the pneumatics-domain components with simple behavioral models. For the pneumatically-actuated control valve, the “Basic” variant replaces two nonlinear pressure-computing elements with simple first-order lowpass filters. The behavior of these two variants compared to experimental data is shown in Figure 8. Note that the “Standard” variant more accurately represents the observed data than the “Basic” variant.

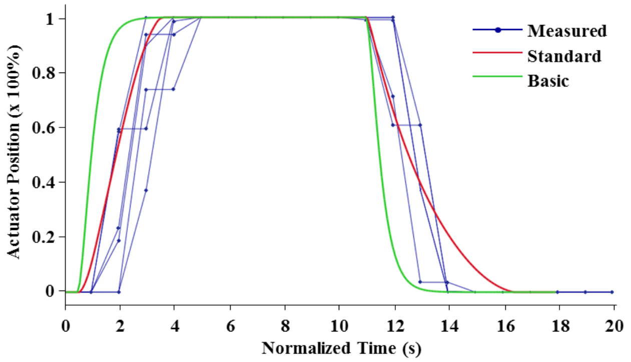


Figure 8. Behavior of pneumatic valve actuator models

The use of domain-specific variants to select appropriate levels of fidelity can significantly reduce simulation time. In this example, the use of the reduced-fidelity “Basic” variant for the pneumatics domain resulted in small differences in valve actuation speed that did not significantly affect the signals of interest in the cryogenic fluid domain. The difference in simulation speed for a run with 1000 seconds of simulation time was dramatic, with the “Standard” variant requiring 676 seconds (1.48x) and the “Basic” variant requiring 354 seconds (2.82x).

5.4. Prognosis Example

To demonstrate how the simulation interacts with a health management algorithm over the message bus, we select a prognosis example demonstrating prognostics of a cryogenic filter. Filters are often periodically replaced on a time-based maintenance schedule. Moving to a condition-based maintenance paradigm can prevent a healthy filter from being replaced and a damaged filter from being used.

The purpose of the filter is to prevent particles contaminating the fluid from moving through to other parts of the system. As fluid passes through a filter, particle matter will collect at the filter and decrease its effective area, thus increasing the pressure drop across the filter for the same flow rate. This behavior is captured in the following equations.

$$Q(t) = \rho\, c\, A(t) \sqrt{\frac{2|\Delta p(t)|}{\rho}}\, sign\big(\Delta p(t)\big) \qquad (1)$$

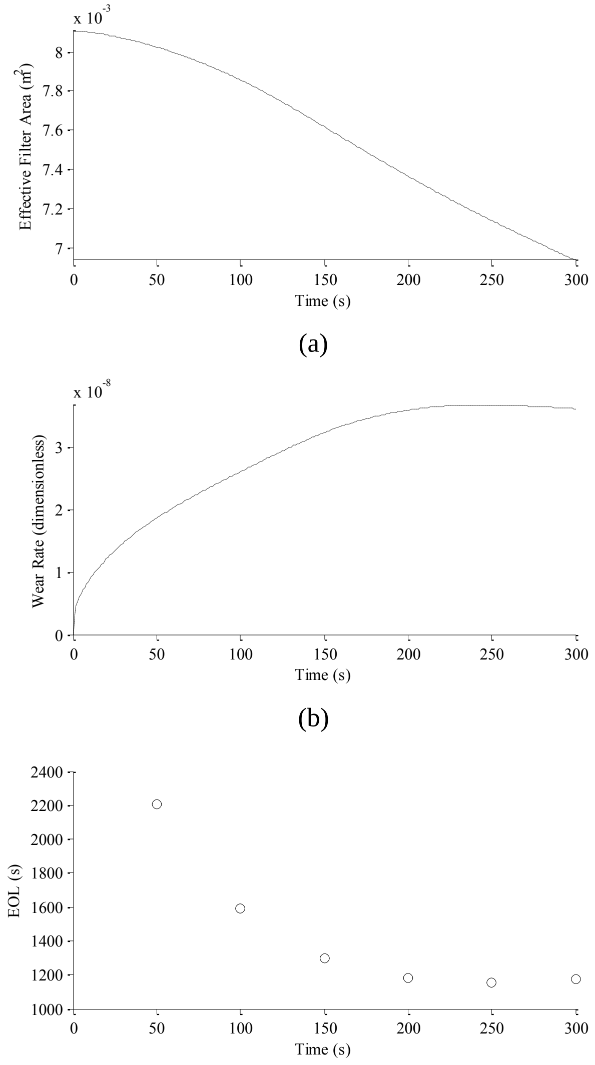
$$\dot{A}(t) = -wQ \qquad (2)$$

Here, Q is the volumetric flow, ρ is the fluid density, c is the flow coefficient, A is the effective filter area, Δp is the pressure drop across the filter, and w is a wear parameter representing the percentage contamination per unit length of fluid (which is, in general, stochastic). The effective filter area decreases as a function of the contamination and the flow rate through the filter.

In this model, the pressure difference is an input and the flow is an output. A model-based prognosis algorithm is used in which the health state of the filter (A and w) is estimated, and this estimate is then used as the initial state in predicting end of life (EOL) and remaining useful life (RUL) of the component (Daigle & Goebel, 2013). For the filter, EOL is defined as the time point at which the effective filter area drops below some specified limit, in this case, 50% of its nominal area.

The prognostics module receives over the message bus the measured values of the differential pressure and the flow, and these serve as inputs to the estimation algorithm (an unscented Kalman filter, see (Julier & Uhlmann, 2004) and (Daigle et al., 2012) for details). The module makes periodic predictions for filter EOL and RUL, and publishes these back to the message bus.

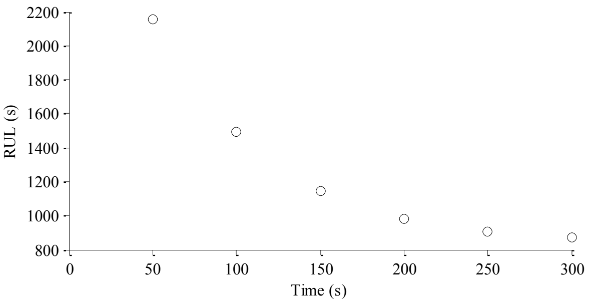
As a demonstration we consider a wear parameter value of 3×10^{-8}. Figure 9 shows the estimated filter area, wear parameter, predicted EOL, and predicted RUL. By 200 s the estimates and predictions begin to converge.



(a)

(b)

(c)



(d)

Figure 9. Filter prognostics results.

6. Conclusion

In this paper, we have discussed the design and application of CryoSim, a simulation-based virtual testbed for SHM applications. The core of CryoSim is a component-based multi-domain system model created in Simulink. Our design includes component-level fault modeling and per-simulation adjustment of model fidelity through the use of variant subsystems. To facilitate the use of CryoSim, we created a Java-based GUI which allows full interactive control of the simulation. Additionally, we have integrated an interface to an external publish/subscribe message bus, enabling CryoSim to function as a drop-in replacement for the CTB system.

We are currently in the process of converting the Simulink model to a standalone version written in C/C++, making use of the Simulink Coder™ software package. This will enable users of CryoSim to run simulations without the need for a license for MATLAB and Simulink. The two-phase cryogenic fluid flow model is under development. When complete, it will be incorporated into CryoSim, taking advantage of the capability to update library components and domain-specific variant subcomponents without impact to previously-available functionality. Because of the model-agnostic design of the CryoSim framework, we anticipate that this architecture will be used for future SHM applications with other multi-domain system models.

Acknowledgement

This work was funded in part by the NASA Automated Cryogenic Loading Operations (ACLO) project under the Office of the Chief Technologist (OCT), the Advanced Ground Systems Maintenance (AGSM) Project under the Ground Systems Development and Operations program and the Integrated Ground Operations Demonstration Unit (IG-ODU) Project under the Advanced Exploration Systems (AES) program.

References

Agusmian P.O., Sas, P. and Van Brussel, H. (2013). Modeling and simulation of the engagement dynamics of a wet friction clutch system subjected to degradation: An application to condition monitoring and prognostics. *Mechatronics*, vol. 23, no. 6, pp. 700-712.

Altman, Y. (2013). *Undocumented MATLAB*, http://undocumentedmatlab.com/

Altman, Y., (2011). Undocumented Secrets of MATLAB-Java Programming, Chapman and Hall/CRC

Balaban, E., Narasimhan, S., Daigle, M., Roychoudhury, I., Sweet, A., Bond, C., & Gorospe, G. (2013, May). Development of a Mobile Robot Test Platform and Methods for Validation of Prognostics-Enabled Decision Making Algorithms, *International Journal of Prognostics and Health Management*, *4*(1).

Biswas, G., Mahadevan, S. (2007, March) A Hierarchical Model-based approach to Systems Health Management. *Proceedings of the 2007 IEEE Aerospace Conference*.

Daigle, M., & Goebel, K. Model-based Prognostics with Concurrent Damage Progression Processes. (2013, May). *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(4), 535-546.

Daigle, M., Saha, B., & Goebel, K. (2012, March). A Comparison of Filter-based Approaches for Model-based Prognostics. In *Proc. of the 2012 IEEE Aerospace Conference*.

Goodrich, C., Narasimhan, S., Daigle, M., Hatfield, W., Johnson, R., & Brown, B. (2009, June). Applying Model-based Diagnosis to a Rapid Propellant Loading System. In *Proc. of the 20th International Workshop on Principles of Diagnosis*, 147-154.

Granger, R. A. (1995). Fluid Mechanics. New York, NY: Dover.

Julier, S. J., & Uhlmann, J. K. (2004, March). Unscented filtering and nonlinear estimation. In *Proc. of the IEEE*, *92*(3), 401–422.

Poll, S., Patterson-Hine, A., Camisa, J., Garcia, D., Hall, D., Lee, C., Mengshoel, O., Neukom, C., Nishikawa, D., Ossenfort, J., Sweet, A., Yentus, S., Roychoudhury, I., Daigle, M., Biswas, G., & Koutsoukos, X. (2007, May). Advanced Diagnostics and Prognostics Testbed. In *Proc. of the 18th International Workshop on Principles of Diagnosis*, 178-185.

Biographies

John P. Barber received the B.S. and Ph.D. degrees in Electrical Engineering in 2001 and 2006 from Brigham Young University, Provo, UT, USA. His research and professional activities have focused on process development and control, as well as modeling and simulation in a number of domains. He is currently an employee of Stinger Ghaffarian Technologies, supporting NASA via the ESC contract at Kennedy Space Center. He is a member of the IEEE.

Kyle B. Johnston received the B.S. degrees in Physics and Astrophysics in 2004 and the M.S. degree in Space Science in 2006 from the Florida Institute of Technology. He has been a primary and co-author on a number of astrophysics journals while working in the academic space. Since 2007 he has worked in the private sector, specializing in the development of scientific simulations, analytical analysis of large datasets, and intelligent passive sensor development. He is currently an employee of ENSCO, Inc. Aerospace Engineering division. Professional society memberships include the American Statistical Association and the American Astronomical Society.

Matthew Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems. He is a member of the IEEE.

Anomaly Detection in Gas Turbine Compressor of a Power Generation Plant using Similarity-based Modeling and Multivariate Analysis

Tomás Carricajo[1], Felipe Kripper[2], Marcos E. Orchard[3], Luis Yacher[4], Rodrigo Paredes[5]

[1,2,4]CONTAC Ingenieros Ltda., Avda. Américo Vespucio Sur 315, Ñuñoa, Santiago 7760005, Chile
(tcarricajo, fkripper, lyacher)@contac.cl

[3]Department of Electrical Engineering, Universidad de Chile, Santiago 8370451, Chile
morchard@ing.uchile.cl

[5]Endesa Chile, Av. Santa Rosa 76, Santiago 8330099, Chile
rapb@endesa.cl

ABSTRACT

This paper introduces advances on the implementation of anomaly detection modules based on a combination of nonparametric models and multivariate analysis of residuals. The proposed anomaly detector utilizes similarity–based modeling (SBM) techniques to represent the process behavior and principal component analysis (PCA) for the study of model residuals; while partial least squares (PLS) is used to select an optimal subset of process variables to be included in the design of the detection module. In addition, the method considers a structured algorithm for the optimal inclusion of representative samples from the data set that is used to define the normal operation of the system. The method is validated using data that characterizes the operation of a compressor in a power generation plant.

1. INTRODUCTION

An anomaly detector (Orchard & Vachtsevanos, 2007) is basically a module that intends to recognize abnormal conditions within the operation of a monitored system. In this regard, the implementation of anomaly detectors is one of the first and most important steps needed to ensure operational continuity of the process, plant safety, as well as high quality standards. Conventional anomaly detection and fault diagnosis algorithms (Isermann, 1997) are typically designed to provide a solution for the supervision of a finite number of fault modes that are believed to be severe, frequent, and "testable"; fault modes that are selected on the basis of a Failure Modes, Effects, and Criticality Analysis (FMECA). This task needs to be performed while simultaneously minimizing the probability of false alarms and the detection time (time between the initiation of a fault and its detection), given a fixed threshold that represents the maximum risk (associated to the fault condition) that is allowed in the system.

Classical fault detection and identification (FDI) methods rely on an accurate model of the system under consideration and the utility of an innovation or "discrepancy" between the actual plant output and the model output, for all possible operating conditions, to detect an unanticipated fault (Isermann, 1997; Isermann & Balle, 1997). The innovation (or residual) method captures the fault signature, and suggests which residuals are normal or which ones result from fault conditions. A variety of techniques have been proposed based on estimation theory, failure sensitive filters, multiple hypothesis filter detection, generalized likelihood ratio tests, model-based approach, statistical analysis, and information theory (Ayhan *et al.*, 2008; Khan & Rahman, 2009; Lebaroud & Cleac, 2008).

If process/system dynamics are not well understood, then verification, calibration, and validation of parametric models may represent a difficult challenge. In contrast, nonparametric models offer a direct representation of nonlinear systems that requires the availability of historical data and a minimal comprehension of the relationships that exist between process variables. The definition of "normal" operation is done only by selecting an appropriate number of data samples that could illustrate moments where the process behaved accordingly to a particular set of requirements or standards; the need of a particular structure or linear/Gaussian assumptions is thus avoided.

In this regard, this article shows the implementation of a monitoring scheme that identifies abnormal operating conditions in a compressor of a power generator plant,

Carricajo et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

utilizing a nonparametric modeling approach known as *Similarity-based Modeling* (SBM). Provided that the plant represents a multivariate nonlinear system, the use of SBM allows generating estimates of the system output that can be used to compute residuals, when compared with actual measurements. Partial least squares (PLS) is used to select an optimal subset of process variables to be included in the design of the detection module, considering to this end the impact that those variables may have in terms of the mean-squared error of model residuals for data associated to "normal" operation.

The method also considers a structured algorithm for the optimal inclusion of representative samples from the data set that is used to define the normal operation of the system. This feature is critical since it is possible that the process model may exhibit problems simply because the database that is being considered for training purposes does not represent all possible operation conditions. Furthermore, in case of implementing a fault detection scheme, the addition of new samples to the database must be done with special attention of not incorporating samples corresponding to these abnormal conditions, since if this is done, the SBM algorithm will consider faulty conditions as known, and hence, normal.

The assessment of the system behavior cannot be performed purely considering each variable residual, since the process is inherently multiple-input multiple-output; consequently, multivariate analysis techniques such as *Principal Component Analysis* (PCA) (Jackson, 1991; Fuente *et al.*, 2009) are employed in order to reduce the space dimension, while ensuring an adequate representation of the residual vector. Additionally, hypothesis testing procedures such as the Hotelling's test (Beale & Kim 2002) are also considered to ensure that the modeling errors remain in a statistically acceptable region.

This paper presents some extensions and results obtained after the implementation of the scheme that was presented in (Tobar, 2010) at facilities of a Chilean power generation company: Endesa-Chile. For confidentiality issues, process labels and time stamps have been discarded in all figures.

This article is organized as follows. Section II presents the necessary theoretical resources to understand the implementation of the proposed system monitoring scheme; i.e., the fundamentals of SBM, partial least squares, principal component analysis, and the Hotelling's test. Section III explains the considerations regarding the data preprocessing procedures, a justification for the implementation of the proposed schemes, and the results obtained for the anomaly detector when using two different sets of process variables as inputs/outputs of the SBM model for the compressor of a power generation plant. Finally, Section IV states the concluding remarks and suggests guidelines for future research work in this field.

2. Theoretical Background

2.1 Similarity-based Modeling for System Monitoring

One advantage of the nonparametric modeling techniques is that they do not require an a priori knowledge of the system, since its implementation is based on the identification of similarities and relationships between a given data set and online observations, instead of the construction of algebraic structures based upon these observed data. A particular case of such structures is the Similarity-based Model (SBM), which estimates the system output by comparing online measurements and a historical database which represents the system under study. SBM has proven to be a successful estimator when used in high dimension systems using considerably low number of training samples (Gong *et al.*, 2009).

In order to understand the SBM basic concept for systems modeling, consider the static system defined by (1):

$$y = f(x), x \in R^m, y \in R^p, \tag{1}$$

where x and y are the system input and output respectively, and $f(\bullet)$ is an unknown function.

When input and output measurements are available for the system described in (1), it is possible to define the following matrices to be used for model identification purposes (D_i and D_0 stand for input and output matrices, respectively):

$$D_i = [x_1, x_2, ..., x_n] \in R^{m\times n}, \tag{2.a}$$

$$D_0 = [y_1, y_2, ..., y_n] \in R^{p\times n}, \tag{2.b}$$

where $y_i = f(x_i)$, $\forall i = 1, \ldots, n$, and the pairs $[x_i, y_i]_{i=1\ldots n}$ accurately represent the system behavior; i.e., they span the regions containing the system operations points.

Hence, SBM assumes that for a given an input x^*, it is possible to estimate $y^* = f(x^*)$ by a linear combination of the columns of D_o denoted by $\hat{y}^*$. Consequently, the problem of estimating $y^* = f(x^*)$ can be regarded as the determination of a vector w such that $\hat{y} = D_0 w$.

This vector can be computed as in (3):

$$w = \frac{\hat{w}}{\vec{1}^T \cdot \hat{w}}, \tag{3.a}$$

$$\hat{w} = (D_i^T \Delta D_i)^{-1} (D_i^T \Delta x_i), \tag{3.b}$$

where Δ is a similarity operator (Gong *et al.*, 2009; Pivoso *et al.*, 1994). SBM is not restricted to any particular similarity operator; however, according to the literature, the selected similarity operator must hold certain properties. For two elements $A, B \in \Re^u$, $A\Delta B$ $\in \Re^+$ must be symmetric, reaching its maximum in A=B and monotonically decreasing with $\|A - B\|$.

Literature does not provide a framework for choosing a suitable similarity operator based on the available measurements. In this work, all similarity operators were based on saturated linear kernels.

SBM is a nonparametric modeling technique that is mainly used to identify static systems (or at least, systems where the dominant time constant is negligible with respect to the data sampling period). In this regard, and especially when the process exhibits noticeable dynamics, the model structure requires some adjustments before its implementation. For example, past observations (both inputs and outputs) may be incorporated as regressors to estimate the system response in time. For this particular case study, though, system dynamics were neglected (thus the process was regarded as a static one). As it was mentioned above, this concept can only be applied when the data is acquired at a very low frequency with respect to the system dominant constant.

SBM residuals can be computed simply using the difference between the model outputs (SBM estimates) and online measurements as in (4). If the estimates differ considerably from the actual measurements in the training data (w.r.t. a given criteria such as mean-squared error), it could be inferred that the associated operating point has not been incorporated yet into the SBM structure, and consequently the optimal database that ultimately defines the SBM model must be extended with samples representing the unknown condition. After the process of incorporating samples to the database is complete, i.e. once for every input x^* the estimation error given by

$$\begin{aligned} e &= y^* - \hat{y}^* \\ &= f(x^*) - D_0 \frac{(D_i^T \Delta D_i)^{-1}(D_i^T \Delta x^*)}{\vec{1}^T \cdot (D_i^T \Delta D_i)^{-1}(D_i^T \Delta x^*)} \in \Re^p \end{aligned} \tag{4}$$

is acceptable under a specified criteria, the relationships between the measured variables should be assessed to ensure consistency with the operation conditions represented in the database. Due to the large number of variables that are present in industrial systems, multivariate-processing algorithms should be implemented to verify these relationships.

2.2 Partial Least Squares

Partial least squares, also referred to as “projection to latent structures”, is a parametric modeling technique. This technique allows system modeling through a reduction of the problem dimensionality and the maximization of the covariance between projections of the input data matrix X and the output data matrix Y (Chiang *et al.*, 2001). It uses a matrix $X \in \Re^{n\times m}$ and a matrix $Y \in \Re^{n\times p}$, where m is the number of variables predictors, n is the total number of observations of data and p is the number of observed variables in Y.

First, the matrices X and Y must be centered on the mean and normalized by their variances. Then, the matrix X can be decomposed into an array called scores $T \in \Re^{n\times a}$ and a loading matrix $P \in \Re^{m\times a}$, where a is the reduced order of the data, the residue matrix $E \in \Re^{n\times m}$.

$$X = TP^T + E \tag{5}$$

The matrix $T{\cdot}P^T$ can be expressed as the sum of products of vectors scores t_j and load vectors p_j.

$$X = \sum_{j=1}^{a} t_j p_j^T + E \tag{6}$$

Similarly, the matrix Y is decomposed into matrices:

$$Y = UQ^T + F = Y = \sum_{j=1}^{a} u_j q_j^T + F \tag{7}$$

If “a” is equal to min (m,n), then E and F are zero and this technique is reduced to the ordinary least squares. Choosing “a” smaller than min (m,n) noise is reduced. The objective is to determine the loading and scores vectors which maximize the correlation between X and Y.

PLS estimates the scores vectors u_j with scores vector t_j as:

$$u_j = t_j b_j^T \text{, or equivalently } U = TB \tag{8}$$

Finally,

$$Y = TBQ^T + F \tag{9}$$

where F is the prediction error matrix. The matrix B is selected to minimize the norm of F. T and U scores matrices are calculated as to maximize the covariance between X and Y for each component “a”.

Although PLS is typically used to generate a linear parametric Multiple-Input Multiple-Output (MIMO) model for the process, as a result of an appropriate selection for the number of projection components, there are other important complementing aspects that can also be studied. Particularly, this article uses an analysis of the coefficients in matrices B, T and U to assess the impact of each of the inputs variables X on each of the output variables Y.

2.3 Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique for correlated variables; i.e. for a given a set of correlated variables, it aims at finding a set of uncorrelated indicators that can help to characterize the variability of the process in a smaller dimension. PCA performs a linear transformation of the data, which is optimal in terms of capturing its variability, and determines a new data set ordered by the level of representation of the entire process variance.

Theoretically, for the data matrix

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}, \qquad (10)$$

which comprises n observations for each one of the m variables, PCA finds a loading matrix $P \in \Re^{m \times a}$, $a \le n$ which relates X to the first principal components being contained in the score matrix:

$$T = XP. \qquad (11)$$

Denoting the i^{th} column of T by t_i, the transformation performed by PCA holds (Chiang *et al.*, 2001) the following properties:

1. $Var(t_1) \ge Var(t_2) \ge ... \ge Var(t_a)$.
2. $Mean(t_i) = 0, \forall i$.
3. $t_i^T t_k = 0, \forall i \ne k$.
4. There is no other transformation of "*a*" components that captures more variations in the data.

Additionally, the projection back on an a-dimensional space is given by (Wise *et al.*, 1990):

$$\hat{X} = TP^T, \qquad (12)$$

and hence, the difference between the original data stored at X and its projection is the residual matrix E:

$$E = X - \hat{X}, \qquad (13)$$

which captures the variations of space generated by the remaining ($m – a$) components, and has typically low signal-to-noise ratio. It has been formally justified (Golub *et al.*, 1983) that, when "*a*" is properly chosen, these remaining components represent the random noise of the measurements, whereas the first "*a*" components describe dynamic variations.

The application of PCA in our system monitoring framework is to reduce the dimension of the error vector "*e*", simplifying in that manner the anomaly detection procedure (in terms of the associated computational cost). Indeed, once the PCA linear transformation has been applied to the error vector, one can easily recognize if the system is behaving in an anomalous manner through the application of a hypothesis test formulated in terms of the main principal components.

2.4 Hotelling's Test

The one sample Hotelling's T^2 index is typically used to test H$_0$: $\mu = \mu_0$ vs. H$_A$: $\mu \ne \mu_0$ in a 2-class classification problem. However, when applied to multivariate Gaussian residual vectors, it also provides the means to compute a scalar threshold that characterizes the maximum acceptable deviation of the model residual, for a given level of significance (Gonzalez *et al.*, 2003). To properly introduce the Hotelling's T^2 test, consider the sample covariance of the data matrix X given by

$$S = \frac{1}{n-1} X^T X \cdot \qquad (14)$$

The Hotelling's T^2 test states that a particular observation $x \in \Re^m$ belongs to the same class as the data in X if the statistic

$$T^2 = x^T S^{-1} x, \qquad (15)$$

is below the threshold

$$T_\alpha^2 = \frac{m(n-1)(n+1)}{n(n-m)} F_\alpha(m, n-m), \qquad (16)$$

where Pr(Z ≤ $F_\alpha(g, k)$) = α if Z~$F_\alpha(g, k)$, an F-distribution with degrees of freedom g and k. When the data matrix X characterizes the model residuals obtained when the process is healthy, then an anomaly may be detected by analyzing the time instants when the alternative hypothesis is accepted.

3. ANOMALY DETECTION IN GAS TURBINE COMPRESSOR OF POWER GENERATION PLANT USING SIMILARITY-BASED MODELING, PLS AND PCA

A monitoring scheme for the detection of anomalies in the operation of the compressor of a Chilean natural-gas power generation plant was implemented using SBM to model the operation of the compressor at many different operating points (even including operation after the execution of maintenance procedures), and PCA for residual analysis. Selection of input/output variables within the structure of the SBM model considered the analysis of the coefficients in matrices associated to PLS models for the aforementioned plant. In this regard, all process variables that exhibited comparatively small weights in the PLS loading-plot (Chiang *et al.*, 2001) were discarded.

Training and validation data included 19,530 observations for each one of the main process variables. All measurements were acquired using OSIsoft PI software (OSIsoft 2013); including signals associated to pressures, temperatures, valves positions, voltages, speed of rotating parts, and other Boolean states that indicated if certain control loops were active. Data from all measured process variables were grouped in an "input" data matrix $X \in \Re^{19530 \times 42}$ and an "output" data matrix $Y \in \Re^{19530 \times 53}$; being 42 and 53 the number of input and output variables in the process, respectively. The i^{th} rows of the matrices X and Y was respectively denoted by $x_i \in \Re^{42}$ and $y_i \in \Re^{53}$, and the matrix containing all the acquired measurements was denoted as $M=[X,Y]\in \Re^{19530\times 95}$. For monitoring purposes, these data points were processed sequentially in order to emulate online observations, although an intermediate normalization step was used in order to avoid biased results due to the different variables magnitude. All the numerical implementations were performed in MATLAB® software.

The implementation of a nonparametric monitoring scheme requires data bases with a comprehensive representation of different process operation conditions. Thus training data considered different operating points for healthy operation, as well as post-maintenance data and abnormal system operation.

3.1 Data Pre-processing

The use of nonparametric models and SBM can only be justified if the system exhibits nonlinearities and the existence of several operating points. PCA was used to quickly identify the existence of these operating points; using only four principal components of training data for this purpose. Figure 1 shows the results of the aforementioned analysis, which captures the 87% of the data variability, where it is evidenced that there are clustered regions for the operation of the compressor. Failure to characterize all these operating points using simply a collection of linear-in-the-parameters models (Gonzalez *et al.*, 2003) inspired the use of a monitoring technique based on SBM. It must be noted that, for confidentiality reasons, data labels cannot be clearly indicated on this article. In addition it is important to mention that, for all practical purposes, the models only incorporated a static characterization of the system. The latter is based on the fact that all thermo-dynamical and mechanical subsystems were always controlled in closed loops that ensured dominant time constants smaller than the data sampling period. Although it is always possible to increase the sampling frequency to a point where the dynamics of the control loops are in evidence, the company explicitly decided to incorporate those features as part of future research activities.

Being stated that the data admits the use of SBM techniques, and assuming that the system dynamics can be neglected, a suitable similarity operator should be defined with respect to the statistical properties of the measurements. After a preliminary study, the similarity operator that best captured the data variability was the saturated triangular operator defined in (17).

$$A\Delta B=\begin{cases} d-\| A-B\|, & \| A-B\|\le d+\varepsilon \\ \varepsilon, & \| A-B\|> d+\varepsilon \end{cases} \qquad (17)$$

where $\varepsilon > 0$ is a small number that ensures $A\Delta B > 0$, and $d > 0$ is a threshold depending on the observations variance. The definition of these parameters heavily depends on the distribution of clusters and the distance between samples in the training data set.

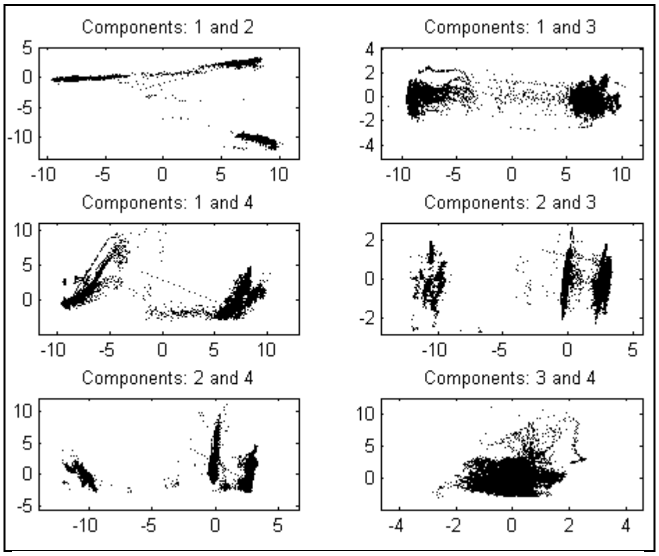


Figure 1. Principal component analysis (PCA) of data from power generation plant. Clusters are the first indication of the existence of several operating points within the data set.

3.2 Database description and a first implementation of the proposed anomaly detection scheme

A subset of data samples was selected from the acquired input/output data for purposes of SBM training and weight characterization. Training data was chosen to incorporate different modes of operation through a novel iterative method that focused on a two-objective optimization problem that minimized of the number of data samples to be included in the training set, while also minimizing the mean-squared error of the resulting SBM-based model residual. This is a critical procedure since, as Figure 2 shows, many operating points are presents within the data that was acquired to characterize the operation of the turbine power and its compressor. In fact, some of the data depicted in Figures 2a, 2b, and 2c contain two different instances of faulty operation, as well as healthy turbine operation, one maintenance procedure, and operation after maintenance. After each fault, the plant always stopped its operation and, during maintenance, the plant was shut down for extended periods of time.

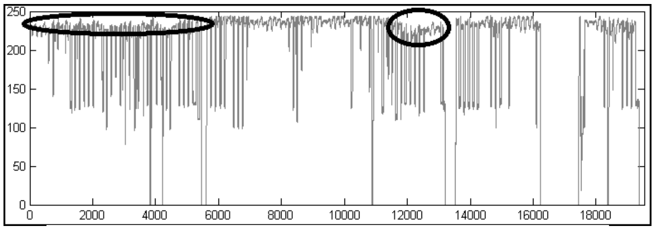


Figure 2a. Illustration of a compressor fault from the standpoint of the power turbine operation.

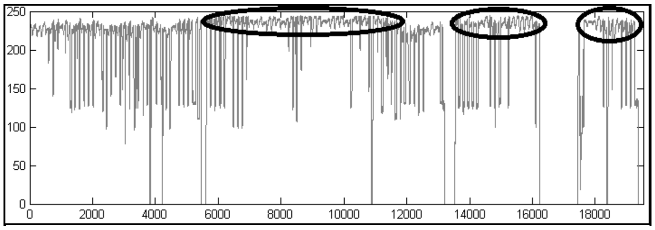


Figure 2b. Normal operation in turbine. Last operating point corresponds to post- maintenance.

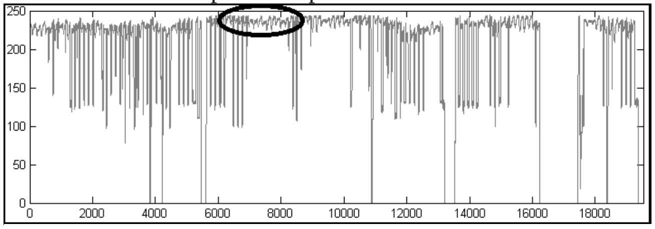


Figure 2c. Data subset used to train the SBM-based anomaly detection algorithm.

A first implementation of the proposed SBM-based anomaly detection scheme was performed for all process variables (i.e., the model had 42 input and 53 output variables), without the dimensionality reduction that some multivariate techniques such as PLS could suggest. Particularly in this case, the obtained mean-squared error (MSE) of the SBM-based normalized power output estimate is presented in Figure 3. It can be seen that using the specified database, the MSE that is related to SBM estimates remains considerably low for the region that contain the training data, other normal operating regions, and even for the post-maintenance data. As expected, the T^2 index is greater than the threshold for data associated to faulty operation. In must be noted that the principal components of the error matrix $E = Y - \hat{Y}$ (which statistically characterizes the training set, where $\hat{Y}$ represents the SBM-based estimate) were used to compute the T^2 index threshold. Additionally, Hotelling's test has been applied to find the 95% confidence ellipse; using for these purposes the software SCAN developed by the Chilean company CONTAC Engineers Ltda (SCAN 2013).

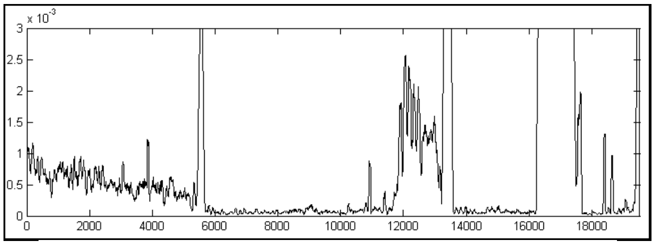


Figure 3. MSE associated to SBM-based estimates for output variables in power generation plant (MSE = 0.000594).

As Figure 4 shows, the T^2 index for the SBM-based residual of the process output variables is adequate to detect a fault in the compressor of the gas turbine.

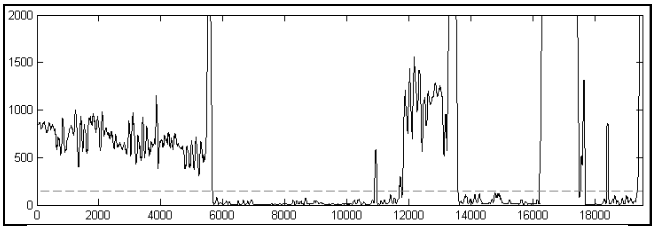


Figure 4. Anomaly detection using a detection threshold based on the Hotelling's T^2 index for training data. Hotelling's T^2 threshold is set in 150.

3.3 Selection of variables and Second Implementation

Utilizing PLS property to maximize the covariance between the input matrix *X* and the output matrix *Y*, a method of dimensionality reduction is proposed based on the analysis of correlations. A reduced set of variables is chosen in order to keep Hotelling's test detecting system faults, while maximizing the correlation and variability between inputs and outputs.

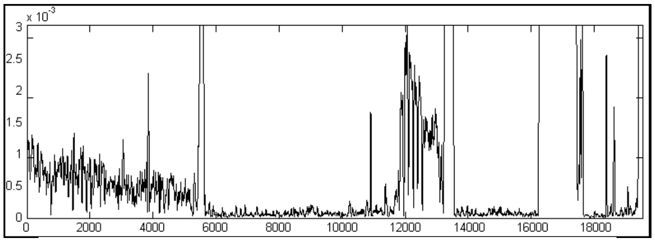


Figure 5. MSE associated to SBM-based estimates for output variables in power generation plant (MSE = 0.001586).

In this case study, and using the proposed methodology, it is discovered that only 5 input and 3 output variables are sufficient for anomaly detection purposes, thus helping to define new matrices $X \in \Re^{19530\times5}$ and $Y \in \Re^{19530\times3}$. Figure 5 shows the square error for the normalized power output from a new SBM-based structure. It is appreciated that for this new set of variables the estimate exhibits a larger MSE in general, although the dimensionality reduction associated to it allows to perform all the necessary computations in real-time. It must be noted, though, that the model still maintains its capability of discriminating normal from abnormal behavior in the plant.

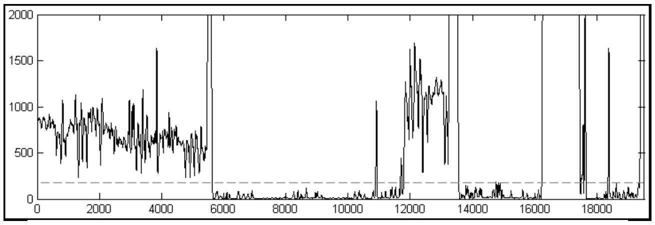


Figure 6. Anomaly detection using a detection threshold based on the Hotelling's T^2 index for training data. Hotelling's T^2 threshold is set in 150.

Using the same methodology as above, a Hotelling's T^2 index is constructed using projections on the space determined by the PCA of the model residuals, using the software SCAN. The results are depicted in Figure 6. As Figure 6 illustrates, and comparing with the results shown in Figure 4, the methodology allowed generating equivalent results for the anomaly detection module although the total number of variables included in the SBM model was reduced from 95 to 8; ensuring appropriate detection of faults in the compressor of the gas turbine. The computational cost was significantly lessened in the second implementation of the detector.

4. CONCLUSION

This article presents and validates a scheme to detect anomalies in the compressor of a gas turbine in a Chilean power generation plant, by comparing the process outputs with SBM-based estimates. The proposed scheme also provides the means to select the data samples that will be included in the training data set by an optimal procedure that minimizes the number of samples while also minimizing the MSE of the model residuals. The use of PCA and PLS techniques helped to dramatically reduce the dimension of the detection problem to a point where it was possible to build the SBM-based detector using only 8 process variables as sources of information. Once a representative training set is constructed, the proposed scheme estimate the system output, exhibiting a reduced MSE and also capturing the relationships between input and output variables even after maintenance procedures. Finally, it is important to know that the detector presented in this paper is monitoring the compressor (the same compressor with which the model was trained) online for more than a year ago. During this time, all anomalies have been confirmed by operators as true faulty conditions.

REFERENCES

Ayhan B., Trussell H., Chow M. Y. & Song M.H. (2008), "On the use of a lower sampling rate for broken rotor bar detection with DTFT and AR-based spectrum methods," *IEEE T. Industrial Electronics*, vol. 55, no. 3, pp. 1421-1434.

Beale G. O. & Kim J. H. (2002), "Fisher discriminant analysis and the T2 statistic for process fault detection and classification," *Industrial Electronics Society, IEEE 2002 28th Annual Conference*, vol. 3, pp. 1995- 2000, 5-8.

Chiang L. H., Russell E. L. & Braatz R. D. (2001), "Fault Detection and Diagnosis in Industrial Systems," *Springer Verlag London Limited*.

Fuente M. J., Garcia-Alvarez D., Sainz-Palmero G. I. & Villegas T. (2009), "Fault detection and identification method based on multivariate statistical techniques," *Emerging Technologies & Factory Automation 2009, IEEE Conference*, pp.1-6, 22-25.

Golub G. H. & van Loan C. F. (1983), "Matrix Computations," *Johns Hopkins University Press*, Baltimore, Maryland.

Gonzalez G. D., M. Orchard, J.L. Cerda, A. Casali & G. Vallebuona (2003), "Local models for soft-sensors in a rougher flotation bank," *Minerals Engineering*, vol. 16, no.5, pp. 441-453.

Gong L. and D. Schonfeld (2009), "Space kernel Analysis," *IEEE International Conference on Acoustics, Speech, and Signal Processing* ICASSP'09, Taipei, Taiwan. *Mediterranean Conference on Control and Automation* MED'07, Athens, Greece.

Isermann R. (1997), "Supervision, fault detection and fault diagnosis methods – an introduction," *Journal of Control Engineering Practice*, vol. 5, no. 5, pp. 639-652.

Isermann R. & Balle P. (1997), "Trends in the Application of Model-Based Fault Detection and Diagnosis of Technical Processes," *Control Eng. Practice*, vol. 5, no. 5, pp. 709-719.

Jackson J. E. (1991), "A users guide to principal components," Wiley.

Khan M. & Rahman M. (2009), "Development and implementation of a novel fault diagnostic and protection technique for IPM motor drivers," *IEEE T. Industrial Electronics*, Vol. 56, No. 1, pp. 85-92.

Lebaroud A. & Cleac G. (2008), "Classification of induction machine faults by optimal time-frequency representations," *IEEE T. Industrial Electronics*, vol. 55, no. 12, pp. 4290-4298.

Orchard M. & Vachtsevanos G. (2007), "A particle filtering-based framework for real-time fault diagnosis and failure prognosis in a turbine engine."

OSIsoft (2013), http://www.osisoft.com/.

Pivoso M. J. & Kosanovich K. A. (1994), "Applications of multivariate statistical methods to process monitoring and controller design," *Int. J. of Control*, vol. 59, pp. 743-765.

SCAN (2013), software developed by the Chilean company CONTAC Engineers Ltda., http://www.contac.cl/scan/.

Tobar F.A., Yacher L., Paredes R. & Orchard M. (2010), "Anomaly Detection in Power Generation Plants using Similarity-based Modeling and Multivariate Analysis," *American Control Conference*.

Tracy N. D., Young J. C. & Mason R. L. (1992), "Multivariate control charts for individual observations," *J. of Quality Control*, vol. 24, pp. 88-95.

Wegerich S. (2008), "Condition Based Monitoring using Nonparametric Similarity Based Modeling," Technology Development, SmartSignal Corporation.

Wise B. M., Ricker N. L., Velkamp D. J. & Kowalski B. R. (1990), "A theoretical basis for the use of principal component models for monitoring multivariate process," *Processes Control and Quality*, vol. 1, pp. 41-51.

I/P Transducer Application of Model-Based Wear Detection and Estimation using Steady State Conditions

Christopher Teubert[1] and Matthew Daigle[2]

[1] *SGT, Inc. NASA Ames Research Center, Moffett Field, CA, 94035, USA*
christopher.a.teubert@nasa.gov

[2] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
matthew.j.daigle@nasa.gov

Abstract

For modern systems, wear estimation plays an important role in preventing failure, scheduling maintenance, and improving utility. Wear estimation relies on a series of sensors, measuring the state of the system. In some components, the sensors used to estimate wear may not be fast enough to capture brief transient states that are indicative of wear. For this reason it is beneficial to be capable of detecting and estimating the extent of component wear using steady-state measurements. This paper details a method for estimating component wear using steady-state measurements, and describes a case study of a current/pressure (I/P) transducer. I/P Transducer nominal and off-nominal behavior are characterized using a physics-based model, and validated against expected component behavior. This model is used to determine steady state responses to many common I/P Transducer wear modes, isolate the active wear mode, and estimate its magnitude.

1. Introduction

As systems are becoming more complex, more expensive, and are being sent to increasingly unreachable places, such as space or the bottom of the ocean, wear detection and estimation become increasingly important. Wear detection and estimation play a critical role in preventing failure, scheduling maintenance, and improving system utility.

Many modern wear estimation techniques rely on measurement of the system's transient states (Daigle & Goebel, 2013; Orchard & Vachtsevanos, 2009; Saha & Goebel, 2009; Luo, Pattipati, Qiao, & Chigusa, 2008). However, in some components, the available sensors may not be fast enough to capture brief transient states that are indicative of wear. This can either be a result of sensor technological limits, or budgetary constraints on sensor selection (as sensors with higher resolution and higher sampling frequencies are generally more expensive). For this reason, it is beneficial to be capable of detecting and estimating the extent of component wear using only steady-state measurements. Previous work in prognostics does not address this need, and a new methodology is required.

This paper describes a method for estimating component wear from steady-state conditions. This is accomplished utilizing a physics-based model that accounts for system behavior in both nominal and degraded conditions, and that is tuned utilizing physical specifications and knowledge of system behavior. This model is then used to map the effect of various modes of wear on steady-state behavior. Combined with a particle filter-based estimation scheme, this model can be used for prognostics, as described in previous work by (Daigle & Goebel, 2011; Orchard & Vachtsevanos, 2009; Saha & Goebel, 2009; Zio & Peloni, 2011).

As a case study, this method is applied to a current/pressure transducer, henceforth referred to as an I/P Transducer or IPT. I/P Transducers are effectively pressure regulators that vary the output pressure depending on the supplied electrical current signal. They operate by throttling a nozzle to create a pressure difference across a diaphragm, which controls the throttling of a valve. These are often used for supplying precise pressures to control pneumatic actuators and valves.

The paper is organized as follows. The development of the IPT model is described in Section 2. Section 3 details the process of mapping IPT wear from steady state conditions, and using that mapping to detect and estimate wear in physical systems. The paper concludes with a discussion of the implications of this research and a description of future work in Section 4.

Christopher Teubert et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

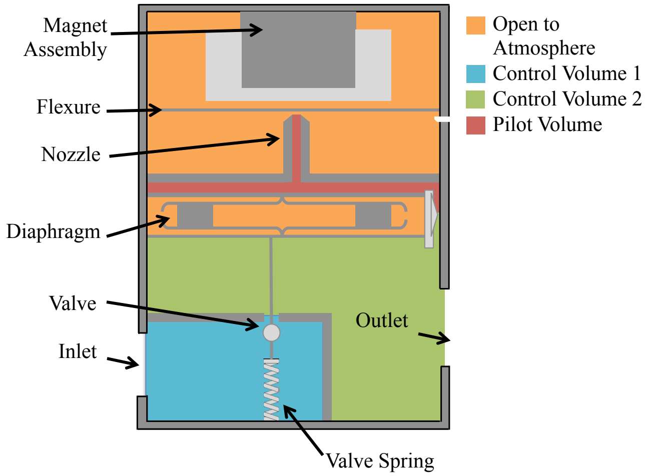


Figure 1. Current/Pressure Transducer Schematic

Figure 2. Current/Pressure Transducer

Table 1. IPT Specifications

Name	Type 1000 IPT
Manufacturer	Marsh Bellofram
Supply Pressure Range	18-100 psig
Input Signal Range	4-20 mA
Output Pressure Range	3-15 psig

2. I/P Transducer Modeling

In this section, we develop a physics-based model of both healthy and faulty IPT behavior. This is used to identify how faulty behavior affects performance for development of the wear detection and estimation methodology, and for possible future prognostic applications. This model was created using domain knowledge of the system's behavior and physical make-up.

As a case study, we use a Marsh Bellofram Type 1000 IPT, illustrated in Figures 1 and 2. This model was chosen because of its use in the pneumatic valve testbed at NASA Ames Research Center (Kulkarni, Daigle, & Goebel, 2013). The IPT is divided into three distinct control volumes (CVs), each marked in a different color in the image. The IPT output pressure varies with the current supplied to the magnet assembly. When the current is high, the magnet assembly throttles the flow out of the pilot nozzle, allowing less air to escape. With a low input current more gas escapes from the nozzle, lowering the pilot pressure. The pressure difference across the diaphragm moves the valve, which adjusts the gas flow between CV1 and CV2. Adjusting this flow changes the pressure in CV2 and in the outlet. Some specifications for this IPT are included in Table 1 (Marsh Bellofram, n.d.).

In this section we will describe development, configuration, validation, and use of the IPT model for both the nominal and wear conditions.

2.1. Problem Formulation

We assume the system may be described by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{v}(t)) \tag{1}$$

$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{n}(t)) \tag{2}$$

where $t \in \mathbb{R}$ is the continuous time variable, $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the state vector, $\theta(\mathbf{t}) \in \mathbb{R}^{n_\theta}$ is the parameter vector, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(t) \in \mathbb{R}^{n_v}$ is the process noise vector, $\mathbf{f}$ is the state equation, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ is the output vector, $\mathbf{n}(t) \in \mathbb{R}^{n_n}$ is the measurement noise vector, and $\mathbf{h}$ is the output equation.

Given a system defined in this way, the objective is to estimate the wear parameter, $\theta_w \in \boldsymbol{\theta}$, given a measured steady state output, $\mathbf{y}_{SS}$, and a known input, $\mathbf{u}$. For this architecture it is assumed that only one mode of wear is occurring at a time (single fault assumption). It may be possible to estimate wear in the case of multiple simultaneous modes of wear, given additional steady state output measurements at other input currents. This is not under the scope of the current research, but will be explored in future research.

2.2. Nominal Model

The IPT model was developed using mass and energy balances. Each CV contains gas at a specific pressure, changing depending on the gas in-flow and out-flow. The system's state is signified by the vector $\mathbf{x}(t)$, consisting of the pressures at each control volume $(p_1(t), p_2(t), p_{pilot}(t))$, the position and velocity of the valve ($x_V(t)$ and $v_V(t)$, respectively), and the flexure position and velocity ($x_F(t)$ and $v_F(t)$, respectively).

The IPT performance is dependent on the supply pressure provided at the inlet, $p_i(t)$, and the signal current sent to the magnet assembly, $i(t)$. These two values make up the input vector, $\mathbf{u}(t)$. For the IPT being modeled, the signal current is between 4 and 20 mA, which varies the outlet pressure, $p_{out}(t)$, between 3–15 PSIG. Outlet pressure is considered to be the only value in the output vector, $\mathbf{y}(t)$.

The input ($\mathbf{u}(t)$), state ($\mathbf{x}(t)$), state derivative ($\dot{\mathbf{x}}(t)$) and output ($\mathbf{y}(t)$) vectors are summarized below:

$$\mathbf{u}(t) = \begin{bmatrix} p_i(t) \\ i(t) \end{bmatrix} \tag{3}$$

$$\mathbf{x}(t) = \begin{bmatrix} p_1(t) \\ p_2(t) \\ p_{pilot}(t) \\ p_{out}(t) \\ x_V(t) \\ v_V(t) \\ x_F(t) \\ v_F(t) \end{bmatrix} \tag{4}$$

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{p}_1(t) \\ \dot{p}_2(t) \\ \dot{p}_{pilot}(t) \\ \dot{p}_{out}(t) \\ v_V(t) \\ a_V(t) \\ v_F(t) \\ a_F(t) \end{bmatrix} \tag{5}$$

$$\mathbf{y}(t) = p_{out}(t) \tag{6}$$

Here velocity, v, and acceleration, a, are defined as the derivative of position, x, and velocity, respectively. Additionally gas flow into a control volume from a bordering control volume is represented by $\dot{q}_{ij}$, where the subscript i represents the first control volume and j the bordering one and $\dot{q}_{ij}$ is the fluid flow into i from j. The flow, $\dot{q}_{ij}$, is a function of the pressure in the control volume, p_i, pressure in the second control volume, p_j, and the area of the opening between them, A_{ij}. These equations are summarized below:

$$\dot{x} = v \tag{7}$$

$$\dot{v} = a \tag{8}$$

$$\dot{q}_{ij} = A_{ij}\sqrt{|p_i - p_j|} * sgn(p_i - p_j) \tag{9}$$

Each of the $\dot{p}$ terms are dependent on the bordering control volumes. The sum over all the interactions with a given control volume gives the total pressure flux. Accounting for all the bordering CVs the $\dot{p}$ equations become

$$\dot{p}_1 = (\dot{q}_{12} + \dot{q}_{10})\frac{R * T_1}{V_1} \tag{10}$$

$$\dot{p}_2 = (\dot{q}_{21} + \dot{q}_{2p} + \dot{q}_{2Out})\frac{R * T_2}{V_2} \tag{11}$$

$$\dot{p}_p = (\dot{q}_{p2} + \dot{q}_{pNozzle})\frac{R * T_p}{V_p} \tag{12}$$

$$\dot{p}_{out} = \dot{q}_{out2}\frac{R * T_{out}}{V_{out}} \tag{13}$$

where R represents the gas constant, and T the temperature in that control volume.

The signal current is supplied to the magnet assembly, which reacts, applying pressure on the flexure. This pressure is greater for greater signal currents. As the flexure stretches, it throttles the airflow out of the nozzle. For low input signals, the flexure flexes less, allowing more air to escape from the pilot volume, decreasing its pressure. The pilot volume is supplied from CV_2 by a small entry to the right of the diaphragm as seen in Figure 1. The net force on the flexure is the sum of the magnet assembly force (F_{Mag}), the resistive force of the Flexure (F_{Flex}), and friction ($F_{Friction}$):

$$F_F = F_{Mag} + F_{Flex} + F_{Friction}. \tag{14}$$

where the individual forces are

$$F_{Mag} = \frac{i^2}{2}(C_{mag} - C_{mag2} * r_{mag}) \tag{15}$$

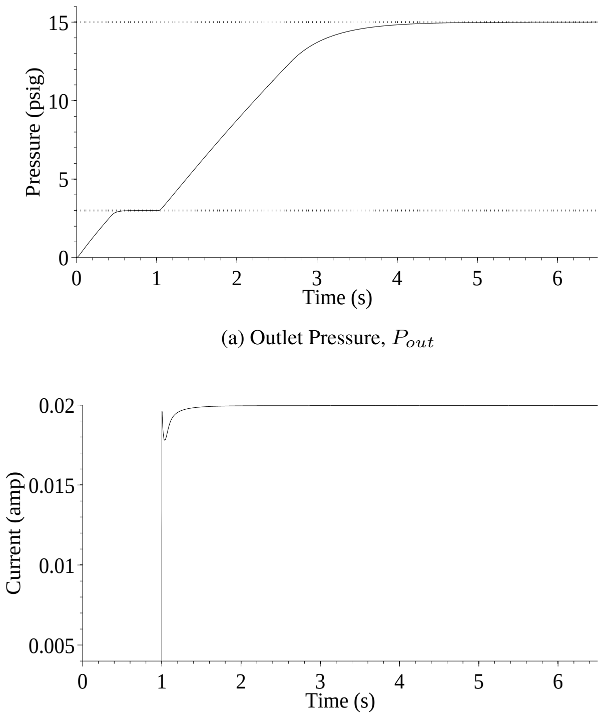
$$F_{Flex} = -k_{Flex}(x_F - x_{F0}) \tag{16}$$

$$F_{Friction} = C_f v_F \tag{17}$$

Here the lumped parameters C_{mag} and C_{mag2} include the gap between the coils and the metal, the area of the metal, the number of turns of the coil, and the magnetic constant. The coil resistance is represented by r_{mag}. Here the value C_f is the coefficient of friction.

The pressure difference between CV_2 and CV_{pilot} produces a closing force on the valve. The lower the input signal the greater the closing force. The net force on the valve (F_V) is the sum of the forces of the Valve Spring (F_{VS}), the force from the pressure difference across the valve (F_{PD}), the force created by the pressure difference across the Diaphragm ($F_{Diaphragm}$), and the force of friction ($F_{Friction}$). The throttling of this valve changes the flow rate between CV_1 and CV_2, affecting the output pressure P_{out}.

$$F_V = F_{VS} + F_{PD} + F_{Diaphragm} + F_{Friction} \tag{18}$$



(a) Outlet Pressure, P_{out}

(b) Current, i

Figure 3. Outlet pressure for different currents

where the individual forces are

$$F_{VS} = -k_V * (x_V - x_{V0}) \tag{19}$$

$$F_{FD} = (p_2 - p_1) * A_V \tag{20}$$

$$F_{Diaphragm} = (p_{Pilot} - p_2) * A_D \tag{21}$$

$$F_{Friction} = C_f v_V \tag{22}$$

where A_V and A_D are the areas of the valve and the diaphragm, respectively, and k_V is the valve spring coefficient.

Each of these relationships were then converted to be in terms of acceleration using the following relationship

$$a_V = \frac{1}{m_V} F_V \tag{23}$$

$$a_F = \frac{1}{m_F} F_F \tag{24}$$

where m_V and m_F are the mass of the valve and flexure, respectively.

The movement of both the flexure and the valve are derived by integrating their respective acceleration equations. The nominal output pressure response is illustrated in Figure 3a, with its respective signal current in Figure 3b.

This IPT model was qualitatively validated by comparing the simulated behavior with known behavior. This domain knowledge was gathered from system documentation, conversations with the manufacturing company, and observations of actual behavior.

2.3. Wear Model

Through discussions with the manufacturers and with users of I/P transducers and similar components, five possible wear modes were indicated. These wear modes are described below:

1. **Inlet Leak** A leak where the supply pressure is provided to CV_1. Modeled by adding a leak of area A_{in} for fluid flow between CV_1 and the surrounding environment (at 1 atm). The resulting fluid flow is represented by

$$\dot{q}_{in} = A_{in}\sqrt{|p_1 - p_{atm}|} * sgn(p_1 - p_{atm}) \tag{25}$$

2. **Valve Seat Leak** A leak between CV_1 and CV_2. Modeled by adding a leak of area A_{VS} for fluid flow between CV_1 and CV_2. A negative A_{VS} models clogging of the valve.

$$\dot{q}_{VS} = A_{VS}\sqrt{|p_1 - p_2|} * sgn(p_1 - p_2) \tag{26}$$

3. **Outlet Leak** A leak at the outlet. Modeled by adding a leak of area A_{out} for fluid flow between CV_2 and the surrounding environment (at 1 atm).

$$\dot{q}_{out} = A_{out}\sqrt{|p_2 - p_{atm}|} * sgn(p_2 - p_{atm}) \tag{27}$$

4. **Valve Spring Weakening** A weakening of the valve spring. Modeled by decreasing the spring coefficient, k_V.

5. **Magnet Assembly Weakening** A weakening of the magnet assembly. Modeled by increasing the resistance in the magnet coils, r_{mag}.

The wear parameters vector, $\boldsymbol{\theta_w}$, consisting of values representing the state of wear for each of the five wear modes, is shown in the below equation

$$\boldsymbol{\theta_w} = \begin{bmatrix} A_{in} \\ A_{VS} \\ A_{out} \\ k_V \\ r_{mag} \end{bmatrix} \tag{28}$$

3. Wear Estimation

Wear estimation is the process of estimating the current extent of wear on a system. This is important for prognostics (predicting failure), scheduling maintenance, and triggering automated mitigation actions. This is often done using methods such as a Kalman Filter or Particle Filter (Arulampalam, Maskell, Gordon, & Clapp, 2002; Daigle, Saha, & Goebel, 2013).

A lookup table method was used for fault estimation. This method was chosen because of its fast, efficient nature and its ability to be applied to both linear and non-linear systems. To define this lookup table the I/P transducer was simulated at various states of each wear mode and various input currents. The steady state output pressure was recorded for each case. The result was used as a reverse lookup table to estimate the wear given a specific observed steady state output pressure for a given input current. Values between data points were linearly interpolated. This was found to be sufficiently accurate given a high granularity lookup table. The granularity of the lookup table can be adjusted to increase accuracy as needed.

The resulting outlet pressure for each fault mode given a high and low input current can be seen in Figure 4. Here the outlet pressure given a high input current is indicated by the green dashed line, while that based on a low current is indicated by the blue solid line. All possible values for the IPT at a given fault level fall between these two points. In this case it was found that monitoring the steady state output pressure does not allow for the estimation of the damage state in the case of an inlet leak. This leak results in a decrease in the pressure in CV_1, which does not result in a change in the output pressure until a much larger leak (around 0.2 m^2). For this reason the Inlet Leak case has not been included in the figures.

By contrast, the valve seat leak has a definite increasing effect on the outlet pressure (Figure 4a). This change in output pressure is a result of additional gas coming into CV_2 from CV_1 through the leak opening. For a leak of 0.005 m^2 the outlet pressure increased by 0.11 psig for a high signal current and 0.022 psig for a low current.

The outlet leak also has a definite and measurable effect on the outlet pressure. As the leak grows in size, more gas escapes from CV_2, resulting in a lower outlet pressure as seen in Figure 4b. For a leak of 0.01 m^2 the outlet pressure decreases by 0.05 psig for a high signal current and 0.0045 psig for a low current.

The valve spring exerts a force on the valve system countering that of the diaphragm. As the spring wears, the spring coefficient, k, decreases. This results in a lower counter force against the diaphragm, causing an increased output pressure as the spring coefficient decreases, as seen in Figure 4c. The effect of this is much more prominent for high input current, where the force of the diaphragm is higher. For a weakening of 0.005 to a k of 3.212 N/m the outlet pressure increased by 0.29 psig for a high signal current and remained the same for a low current.

Finally, wear in the magnet-coil assembly is simulated here by increasing the coil resistance. This, in turn, reduces the force of the magnet on the flexure proportionally with input current. The decreased force results in a greater pressure difference across the diaphragm. This closes the valve, and results in a decreased output pressure as seen in Figure 4d. This effect is much almost unseen for the low input current as a result of how the effect scales with current. For an increase of $0.1\ \Omega$ to a r_{Mag} of $180.1\ \Omega$ the outlet pressure decreased by 0.045 psig for a high signal current and remained the same for a low current.

Each of these four wear modes resulted in a change in outlet pressure. The results for single point wear have been summarized in Table 2.

Table 2. Affect of Wear Modes on Outlet Pressure

Wear Mode	Effect
Inlet Leak	None
Valve Seat Leak	Increased Outlet Pressure
Outlet Leak	Decreased Outlet Pressure
Worn Spring	Increased Outlet Pressure
Work Coil	Decreased Outlet Pressure

Once the relationship between the fault parameter (θ), input current (i), and the measured steady state output ($\mathbf{y}_{SS}$) has been determined the resulting knowledge base can be used for wear isolation and estimation.

Two measurements with two different input current levels are required to completely isolate the fault cause. This is to differentiate between two faults that result in the same effect on output pressure. For example, if the outlet pressure is measured to be higher than it should be, that could either be indicative of a worn spring or a valve seat leak. Each of these leaks has a different relationship with input current. The second measurement allows for isolation between similar such faults. For systems with additional fault modes additional measurements may be needed to isolate between similar faults.

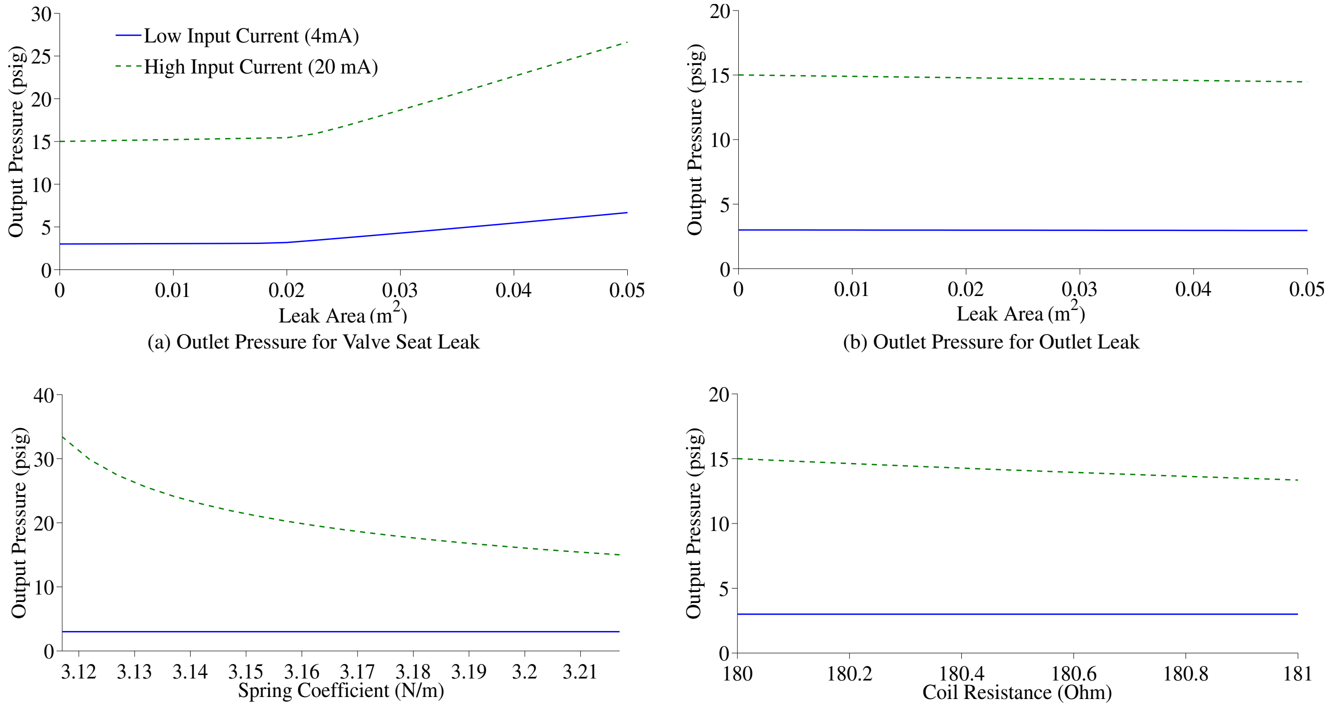
The following section details an example of this method.

3.1. Example

For this example let us assume we have a leak in the valve seat of 0.001 m^2.

The first measurement of steady state outlet pressure is 15.0215 psig at the maximum signal current of 20 mA. Using the reverse lookup table there are two possible options: a leak in the valve seat of 0.0098 m^2, or worn spring with a spring constant of 3.2167 N/m (down 0.0003).

To definitely isolate the wear mode a second measurement is taken, this time with the minimum signal current of 4 mA. The outlet pressure is measured to be 3.0044 psig. This can either correspond to a leak in the valve seat of 0.0105 m^2, or worn spring with a spring constant of 3.173 N/m (down 0.044).



(a) Outlet Pressure for Valve Seat Leak

(b) Outlet Pressure for Outlet Leak

(c) Outlet Pressure for Worn Spring

(d) Outlet Pressure for Worn Coil

Figure 4. Damaged Outlet Pressure

Both measurements result in a valve seat leak estimation that is fairly similar, allowing the user to estimate that there is a leak in the valve seat of around 0.01015 m^2. This is calculated by taking the average of the two estimates. The difference in these measurements is due to measurement noise (**n**). Additional measurements at different input currents could be used to further refine the damage estimate, and filter out system noise.

4. Conclusion

This paper details the development of a model-based wear estimation approach using steady state measurements of the outlet pressure of a current/pressure transducer. This approach was then applied for the wear modes of Inlet leaks, Valve Seat Leaks, Outlet Leaks, Spring Wear, and Coil Wear, which were determined to be the most likely modes of failure.

This method was shown to be effective in identifying wear in simulations for a worn coil, worn spring, outlet leak, and leak in the valve seat. With each of these wear modes the resulting effect on the outlet pressure was different when considering two different input currents. Measuring the outlet pressure at two different input currents allows for the identification of the failure mode. The lookup table created in this study can then be used to estimate the severity of the wear. The results here demonstrate the effectiveness of steady state wear estimation for an I/P transducer.

This approach to wear estimation allows for wear estimation for components where sensors may not be fast enough to capture brief transient states that are indicative of wear. The results from wear estimation routines such as this one can then be used to create a prognostic model, schedule maintenance, or trigger automated mitigation action.

This study relied on physics-based simulations of IPT behavior validated against observations of actual system behavior. We are currently in the process of constructing a testbed that will include the IPT modeled for this study (Kulkarni et al., 2013). Future work includes testing this method of wear estimation in this testbed. Additionally, future work includes the application of prognostics utilizing wear estimation, estimation of multiple simultaneous wear modes, and uncertainty in wear estimation from steady state conditions.

Acknowledgment

This work was funded in part by the NASA Automated Cryogenic Loading Operations (ACLO) project under the Office of the Chief Technologist (OCT), the Advanced Ground Systems Maintenance (AGSM) Project under the Ground Systems Development and Operations program and the Integrated Ground Operations Demonstration Unit (IGODU) Project under the Advanced Exploration Systems (AES) pro-

gram.

Nomenclature

a	Acceleration
A	Area
C_f	Coefficient of Friction
C_{mag}	Magnetic Lumped Parameter
$\mathbf{f}$	State Equation
F	Force
$\mathbf{h}$	Output Equation
i	Current
k	Spring Constant
m	Mass
$\mathbf{n}$	Sensor Noise Vector
p	Pressure
r_{mag}	Magnetic Coil Resistance
R	Gas Constant
t	Time, continuous
T	Temperature
$\mathbf{u}$	Input Vector
w	Wear Parameter
$\mathbf{w}$	Wear Parameter Vector
x	Position
$\mathbf{x}$	State Vector
v	velocity
$\mathbf{v}$	Process Noise Vector
$\mathbf{y}$	Output Vector
$\boldsymbol{\theta}$	Parameter Vector

References

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.

Daigle, M., & Goebel, K. (2011, August). A model-based prognostics approach applied to pneumatic valves. *International Journal of Prognostics and Health Management*, *2*(2).

Daigle, M., & Goebel, K. (2013, May). Model-based prognostics with concurrent damage progression processes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(4), 535-546.

Daigle, M., Saha, B., & Goebel, K. (2012, March). A comparison of filter-based approaches for model-based prognostics. In *Proceedings of the ieee aerospace conference.*

Kulkarni, C., Daigle, M., & Goebel, K. (2013, September). Implementation of prognostic methodologies to cryogenic propellant loading testbed. *Proceedings of 2013 IEEE Autotestcon.*

Luo, J., Pattipati, K. R., Qiao, L., & Chigusa, S. (2008, September). Model-based prognostic techniques applied to a suspension system. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *38*(5), 1156 -1168.

Marsh Bellofram. (n.d.). Type 1000 i/p & e/p transducers [Computer software manual].

Orchard, M., & Vachtsevanos, G. (2009, June). A particle filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*(3-4), 221-246.

Saha, B., & Goebel, K. (2009, September). Modeling Li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the annual conference of the prognostics and health management society 2009.*

Zio, E., & Peloni, G. (2011). Particle filtering prognostic estimation of the remaining useful life of nonlinear components. *Reliability Engineering & System Safety*, *96*(3), 403-409.

Biographies

Christopher Teubert received his B.S. in Aerospace Engineering from Iowa State University in 2012. While at Iowa State University, he conducted research on asteroid deflection mission design and asteroid fragment propagation for Iowa State University's Asteroid Deflection Research Center (ADRC). Previous to his current position he worked as a spacecraft systems engineer for a Mars sample return mission as part of the NASA Academy Program. He is currently researching systems and algorithms for diagnostics, prognostics, and system health management for Stinger Ghaffarian Technologies, Inc. at NASA Ames Research Center's Prognostic Center of Excellence (PCoE). He plans to begin pursuing a M.S. in 2014.

Matthew Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems.

Probabilistic delamination diagnosis of composite materials using a novel Bayesian Imaging Method

Tishun Peng[1], Abhinav Saxena [2], Kai Goebel[3], Shankar Sankararaman[4], Yibing Xiang[5], and Yongming Liu[6]

[1,5,6]*School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ, 85281, USA*
tpeng7@asu.edu
yibing.xiang@asu.edu
yongming.liu@asu.edu

[2,4]*SGT,NASA Ames Research Center, Moffett Field, CA, 94035, USA*
abhinav.saxena@nasa.gov
shankar.sankararaman@nasa.gov

[3]*NASA Ames Research Center, Moffett Field, CA, 94035, USA*
kai.goebel@nasa.gov

ABSTRACT

In this paper, a framework for probabilistic delamination location and size detection is proposed. A delamination probability image using Lamb wave-based damage detection is constructed using the Bayesian updating technique. First, the algorithm for the probabilistic delamination detection framework using Bayesian updating (Bayesian Imaging Method - BIM) is presented. Following this, a fatigue testing setup for carbon-carbon composite coupons is introduced and the corresponding lamb wave based diagnostic signal is collected and interpreted. Next, the obtained signal features are incorporated in the Bayesian Imaging Method to detect delamination size and location, as along with corresponding uncertainty bounds. The damage detection results using the proposed methodology are compared with X-ray images for verification and validation. Finally, some conclusions and future works are drawn based on the proposed study.

1. INTRODUCTION

Composite materials are widely used in many applications, such as rotorcraft, aerospace, automobiles, and civil engineering structures because of their low weight and high strength properties. Delamination damage may happen within the composite plate due to impact loading or cyclic loadings. Because of the embedded nature of delamination damage, visual inspection is not able to easily detect it and, therefore, nondestructive evaluation (NDE) techniques are generally used and extensively investigated for this type of diagnosis problems in composites.

Currently, there are many deterministic non-destructive techniques (NDT) available for the delamination diagnosis, such as thermography (Koruk and Kilic 2009; Mielozyk, Krawczuk, Malinowski, Wandowski and Ostachowicz 2012), ultrasonic (Kazys and Svilainis 1997), X-ray (Nicolleto and Hola 2010), and eddy currents (Grimberg, Premel, Savin, Le Bihan and Placko 2001; Sophian, Tian, Taylor and Rudlin 2001). A comprehensive review of these methods for delamination location and size detection is presented in (Cheng and Tian 2012). Most NDE methods require that the specimens to be maintained under certain strict experiment condition for ex-situ damage detection. Alternatively, with the development of Lamb wave-based damage detection methods, piezoelectric sensors have been widely used (Lemistre and Balageas 2001; Giurgiutiu, Zagrai and Bao 2002) for structural health monitoring because of their low cost and high efficiency (Constantin, Sorohan and Gavan 2011). Lamb waves can propagate in thin plate without significant dispersion in certain modes of wave propagation (Scalea, Francesco, Robinson, Tuzzeo and Bonomo 2002). Using proper mode selection, piezoelectric sensor networks can be used for damage inspection of composite plate structures (Wang, Rose and Chang 2004). The advantage of this method lies in that embedded or surface mounted PZTs can be used for in-situ monitoring of structure's health condition. In this technique, damage features are extracted from the received Lamb wave signal using signal-processing algorithms. Several characteristics of the received signal (e.g., the attenuation,

Tishun Peng et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

phase shift, energy etc.) indicate changes from baseline i.e. damage features (Raghavan and Cesnik 2007).

For Lamb wave detection methods, there are two common approaches (Raghavan and Cesnik 2007)- (a) pulse-echo, and (b) pitch-catch. Using pulse-echo based approach, damage position can be detected by methods like triangulation , (Su and Ye 2009; Zhou, Su and Cheng 2011) or using cumulative coefficient change (Zhao, Gao, Zhang, Ayhan, Yan, Chiman and Joseph 2007). Furthermore, it also allows estimating detect the damage intensity if damage location is known (Peng, He, Liu, Saxena, Celaya and Goebel 2012). However, most existing damage detection techniques based on Lamb waves are deterministic and cannot systematically include the uncertainties, such as measurement uncertainty and model parameter uncertainty in the damage diagnosis. In view of the above mentioned difficulty, a probabilistic damage detection method for in-situ applications is proposed in this study. This novel method is presented here to simultaneously detect the damage location and size, then provide their confidence information. The proposed method combines the Lamb wave-based damage detection technique and a novel Bayesian Imaging Method (BIM) to achieve this goal.

This paper is organized as follows. First, the Bayesian theorem is introduced and a probabilistic delamination detection, localization, and size estimation framework (Bayesian Imaging Method) is developed. Then, a Lamb wave based test and signal analysis setup is presented for diagnostic feature extraction. Next, an example is presented to validate the proposed Bayesian Imaging method for delamination size and location detection. Damage diagnosis uncertainty bounds are simultaneously generated. Finally, some conclusions and future works are drawn based on the proposed study.

2. Damage diagnosis algorithm using Bayesian imaging method

Bayes' theorem is widely used in image processing, which combines the prior distribution of a realistic image and utilizes new measurement data to improve image resolution or segmentation (Li, Dong, Guan, Li and Zhou 2007; Pickup, Capel, Roberts and Zisserman 2009). It is also used to compress the information needed to reconstruct the image by optimizing the basis-function weights, which is a powerful tool to deal with the problem of limited measurements (Ji, Xue and Carin 2008). These methods are explained briefly next.

2.1. Bayes' Theorem

Bayes' theorem is commonly used for probabilistic inference or learning process. It can combine the prior belief about parameters and current system response to provide a reasonable prediction of parameters distribution. Let $p(\theta)$ denote the prior distribution of parameter θ. According to the Bayes' theorem, the posterior distribution of θ is given as

$$q(\theta) \propto p(\theta)p(x'|\theta) \qquad (1)$$

where $p(x'|\theta)$ is the likelihood function, which reflects the current system response x' and $q(\theta)$ is the posterior distribution of updated parameters. Let x' be an in-situ measurement, $M(\theta)$ is the prediction value based on a model. If there is no measurement noise or model uncertainty, the measurement would be identical with the model prediction, i.e. $x' = M(\theta)$. However, this is usually never the case. Therefore, it is necessary to account for measurement noise ϵ and model uncertainties τ and the relationship between x' and $M(\theta)$ can be expressed as

$$x' = M(\theta) + \epsilon + \tau \qquad (2)$$

Assuming that the two error term ϵ and τ are independent zero mean normal variables (Bell 2001; Adam 2002), the sum of them can be expressed as a new random variable $e = (\epsilon + \tau) \sim N(0, \sigma_e)$. Therefore the likelihood function $p(x'|\theta)$ can be expressed as

$$p(x_1', x_2', \dots x_n'|\theta) = \frac{1}{\left(\sqrt{2\pi}\sigma_e\right)^n} exp\left(-\frac{1}{2}\sum_{i=1}^{n}\left(\frac{(x_i' - M(\theta))}{\sigma_e}\right)^2\right) \qquad (3)$$

where n is the number of available measurements. Substituting Eq. (3) into Eq.(1), the posterior distribution of parameter θ is expressed as

$$p(\theta|x_1', x_2', \dots x_n') \propto$$
$$p(\theta)\frac{1}{\left(\sqrt{2\pi}\sigma_e\right)^n} exp\left(-\frac{1}{2}\sum_{i=1}^{n}\left(\frac{(x_i' - M(\theta))}{\sigma_e}\right)^2\right) \qquad (4)$$

where the posterior distribution of the parameter θ can be approximated by the samples drawn by the Markov chain Monte Carlo simulation.

2.2. Damage Diagnosis and Bayesian Imaging Development

In the damage detection problem, the sensor signal can be periodically accessed during the servicing operations of structures and signal features can be extracted using appropriate signal processing techniques. To predict the damage size and location, a physics model $M(\theta)$ describing the relationship between the signal features and damage information should be developed first, which is developed using information extracted from available testing datasets. Next, a likelihood function for the updated parameters, i.e., delamination size and location is built considering the measurement and model uncertainties. Now, the posterior belief about damage can be estimated from the posterior distribution of the updated parameters. In this case the parameter vector θ comprises of delamination geometric center coordinate (x_0,y_0) and delamination size a. Since no prior belief is available for any of these three parameters, the prior distribution x_0, y_0 are assumed to be uniform distribution cover the whole possible region where the delamination may appear. The delamination size distribution is uniformly distributed from zero to a large size (e.g, the physical length of the specimen as the largest possible delamination size). A very small quantity (i.e., 0.001 mm) for the delamination distribution lower bound is used to avoid numerical difficulties. The overall diagnosis framework is shown in Fig. 1.

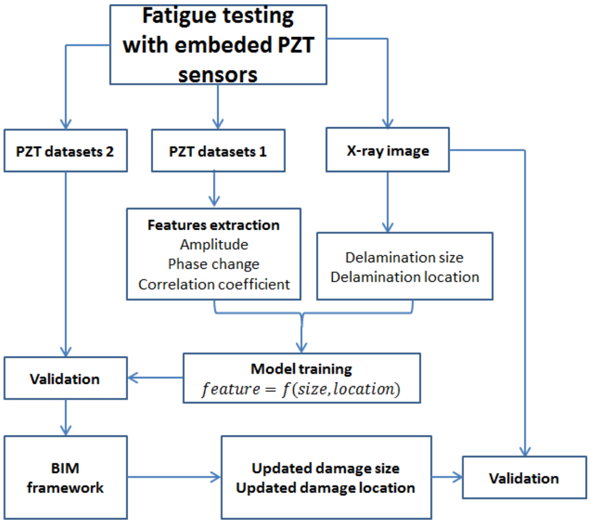


Figure 1. Flowchart for the damage diagnosis method

In Bayesian updating, it is nontrivial to derive an analytical solution if the posterior distribution is non-parametric or very complex, which is the case in this problem. Therefore, the Markov-Chain Monte-Carlo (MCMC) method is used to draw samples. Detailed discussion on MCMC method can be found in several references (Hasting 1970; Peskun 1973; Cowles and Carlin 1996; Fort, Moulines and Priouret 2012) and is, therefore, not discussed here.

The key idea behind the proposed BIM is that the entire specimen is discretized into many small cells (e.g., with size 1mm ×1mm in the current study) and each cell is assigned an associated probability of damage. The probability of damage is updated based on measured signal features using the Bayesian technique. The updated posterior distribution at each cell can be used to construct an image that directly represents the damage location and size.

3. Delamination Diagnosis Experiment Using Piezoelectric Sensor

In this section, the proposed BIM is demonstrated using an experimental study. The testing datasets are used to train the model $M(\theta)$ to get the likelihood function. The Bayesian updating results are compared with X-ray images for model verification and validation. Details are presented next.

3.1. Fatigue Cycling Experiment Setup

In a separate effort run-to-failure fatigue experiments were conducted using composite coupons with 12 plies. Torayca T700G uni-directional carbon-prepreg material was used for this 15.24cm × 35.56cm dog bone geometry coupons and a notch (5.08 mm × 19.3mm) is introduced to induce stress concentration, as shown in Fig. 2.

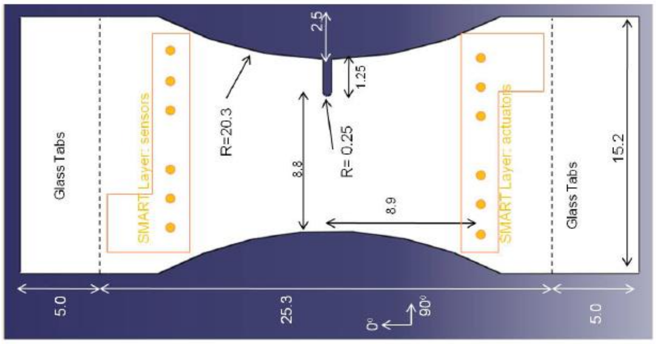


Figure 2. The geometry of the dog bone coupon (unit: cm)

These experiments served several objectives - (i) collection of run-to-failure data with periodic system health data using PZT sensors, (ii) collection of ground-truth data for the delamination to validate the sensor measurement analysis, (iii) accounting for variations between samples of same internal structure (layup), and (iv) characterizing variations between samples of different internal structures. Three symmetric layup configurations were chosen to account for the effect of the ply orientation: Layup 1: $[0_2/90_4]_2$, Layup 2: $[0/90_2/45/-45/90]_2$, and Layup 3: $[90_2/45/-45]_2$. Two six-PZT sensor SMART Layers from Acellent Technologies, Inc (Fig. 3(a)) were attached to the surface of each sample. In Fig. 3(a), actuator 1 to 6 were used to actuate the PZT signal and sensors 7-12 collect the corresponding signal propagation through the plate. Each actuator and sensor acted as a diagnosis path to interrogate the damage information. A more detailed description about these experiments is given in (Saxena, Goebel, Larrosa, Janapati, Roy and Chang 2011). Fig. 3(a) shows such a path form actuator 5 to sensor 8, which is represented as path 5⟶8. The other paths follow the same rule as above in the following sections.

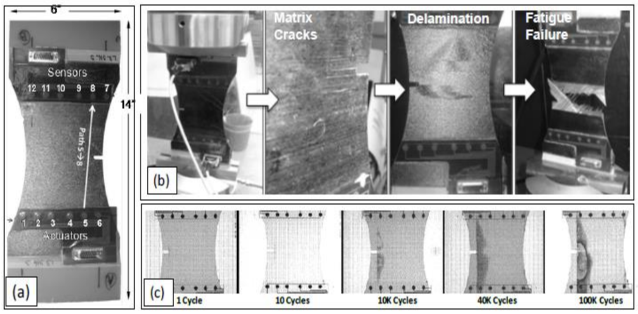


Figure 3. (a) Coupon specimen, SMART layers location, and diagnostic path from actuator 5 to sensor 8,
(b) Development of matrix cracks and delamination leading to fatigue failure,
(c) Growth in delamination area during the increased loading cycles.

Using this configuration of sensor network, all PZTS are used one by one as actuator to actuate the Lamb wave, which is received by other acting as sensors. It is expected that the growth in delamination size will be captured in the received signals from a particular diagnosis paths that cover delamination area (e.g. path 5⟶8), which was validated by the comparison between features and delamination size in literature (Saxena, Goebel, Larrosa, Janapati, Roy and Chang 2011). For diagnosis path 5⟶8, the signal received by sensor 8 at different loading cycles is plotted in Fig. 4. As illustrated in Fig. 4, an increase in delamination size can be captured by monotonic trends in features (amplitude, correlation coefficient, and phase change). Conceptually, a change (decrease) in normalized amplitude reflects the energy dissipation due to the damage. The phase angle change is attributed to the increased wave traveling distance induced by the damage. The correlation coefficient change reflects the signal perturbation due to the new waves generated at the damage surfaces due to reflections (Raghavan and Cesnik 2007). All of these features are computed by comparing the received signal from a pristine coupon, called baseline and the signals from damaged coupons.

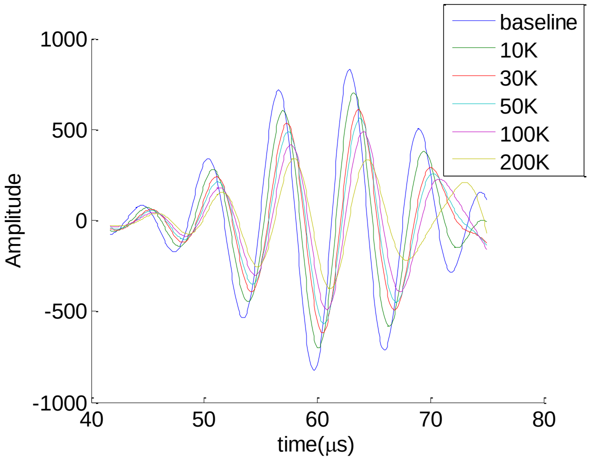


Figure 4. Changes in signal received at sensor 8 as a function increasing fatigue cycles.

3.2. Data Processing

Observation from x-ray images of the damaged coupons reveal that damage grows from the tip of the slit in a characteristic way extending as a half elliptical lobes. Therefore, damage shapes are modeled as half elliptical lobes. Using the sensor network and the analysis method described above, there are two parameters describing these half elliptical shapes that would possibly affect the received signal, which is shown in Fig. 5. The green ellipse is the delamination area observed from the x-rays and the red envelope is introduced to cover the entire area, whose radius is used as a proxy for delamination size a. The distance from the delamination center to the diagnosis path is denoted by d. Features extracted from measurements can be related to these two parameters. For instance, corresponding features for actuator 5 are given in Fig. 6.

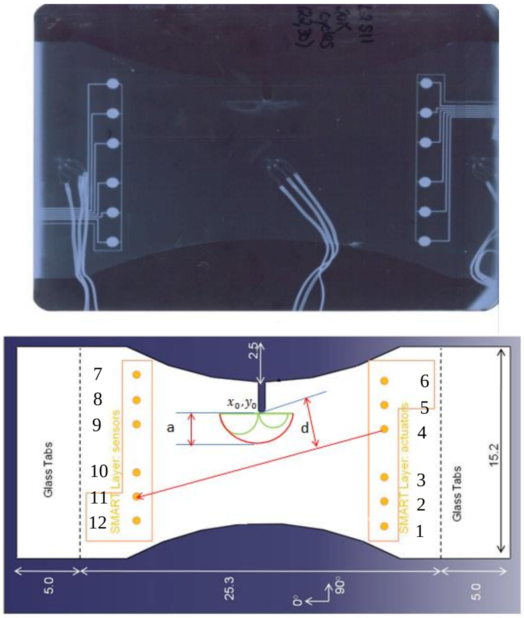
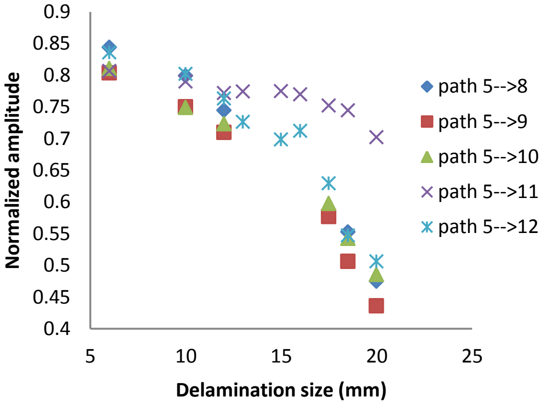
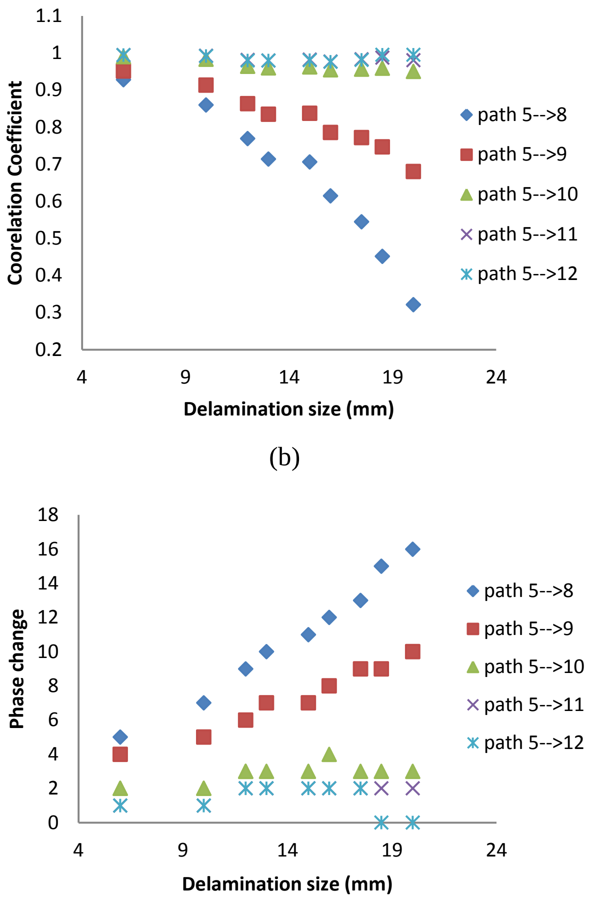


Figure 5. X-ray image and schematic representation delamination



(a)



(b)

(c)

Figure 6. Features related to actuator 5 for different delamination size and distance. (a). Normalized amplitude, (b). Correlation coefficient, (c). Phase change

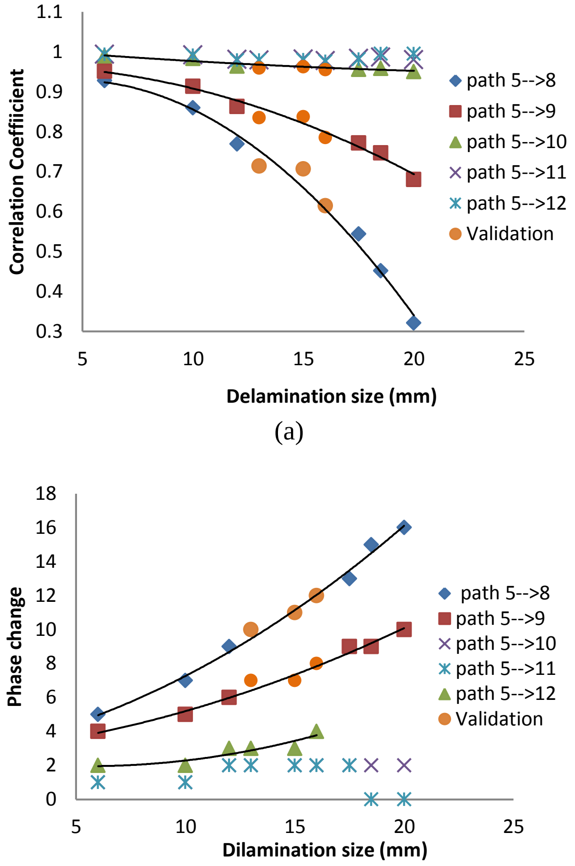
As shown in Fig. 6, the correlation coefficient and phase change features are more sensitive to the distance compared to the normalized amplitude. For example for a fixed distance, these two features have monotonic relationship with the delamination size, which is consistent with the trend in Fig. 4. In order to use these two features in the BIM proposed earlier, a model is introduced to express the relationship between the features with the delamination size and position. A generic expression can be written as

$$feature = f(a, d) \qquad (5)$$

where a is the delamination size, d is the distance from the delamination center to the direct diagnosis path. It should be noted that Eq. (5) is a generic expression and does not limit to a specific function type. In this study, a polynomial regression model is used. Using the trend in the datasets, the model used for these two features is given as

$$feature = f_1(d) \times a^2 + f_2(d) \times a + f_3(d) \quad (6)$$

where $f_i(d) = \beta_{i1} \times \ln(d) + \beta_{i2}$ for correlation coefficient; β_{ij} is regression coefficients, which can be obtained by learning from the training datasets. After tuning these coefficients, the testing and fitted results for features are shown in Fig. 7. The yellow dots are the validation data and the rest are used for the training. It is can be seen that the simple regression model above gives satisfactory results except for the regions where the delamination is far away from the diagnostic path.



(a)

(b)

Figure 7. The testing data and curve fitting
(a). Correlation coefficient, (b). Phase change

The data analysis presented above is from a single actuator (actuator 5), but a similar trend is observed for data from other actuators. For a given delamination defect, damage information from different actuators and wave paths can be combined to provide a better estimation of the delamination size and location. The following section, presents an example of the BIM method application.

4. Demonstration Example

As described in section 2, the physics model M(θ) is needed to show the relationship between damage information and signal features, which can be substituted by the fitting model shown in Eq. (6). The posterior belief about the damage is estimated by the posterior distribution of the updated parameters. Parameter θ represents the delamination center coordinate (x_0, y_0) and delamination size a. Since no prior belief is available for these three parameters, the prior distributions of location is assumed to be $x_0 \sim U(93.5,259.5)$, $y_0 \sim U(25,125)$, which covers all the possible location where delamination may appear. The coordinate definition of this specimen is given in Fig. 8. The delamination size distribution is assumed as $a \sim U(0.001,18)$, where U means uniform distribution. The likelihood function is built based on the difference between fitting model and real experimental data. The measurements from actuator 5 and 6 are utilized in Bayesian updating, as given in Table 1. It should be noted that each updating iteration incorporates one measurement in the BIM framework. The aspect ratio for these two dimensions are assumed to be 2.5 based on the experimental X-ray image.

Table 1. The sensor measurements for given delamination

Actuator	5		6	
Sensor	Correlation Coefficient	Phase change	Correlation Coefficient	Phase change
8	0.7142	10	0.6858	10
9	0.8351	7	0.8279	7
10	0.9595	3	0.9476	4

The posterior distribution of (x_0, y_0, a) can be estimated by the samples drawn using the MCMC, which updates the belief about the delamination location and delamination size at each updating iteration. At the same time, corresponding median and uncertainty bound predictions are computed to describe the accuracy of each updating result. The delamination location estimates after each updating iteration are shown in Fig. 9.

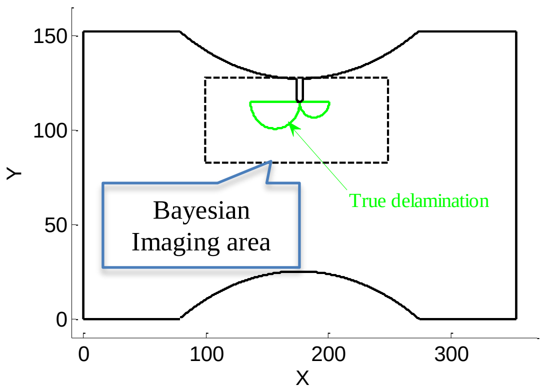


Figure 8. The definition of the sample coordination and specific area to show the Bayesian image

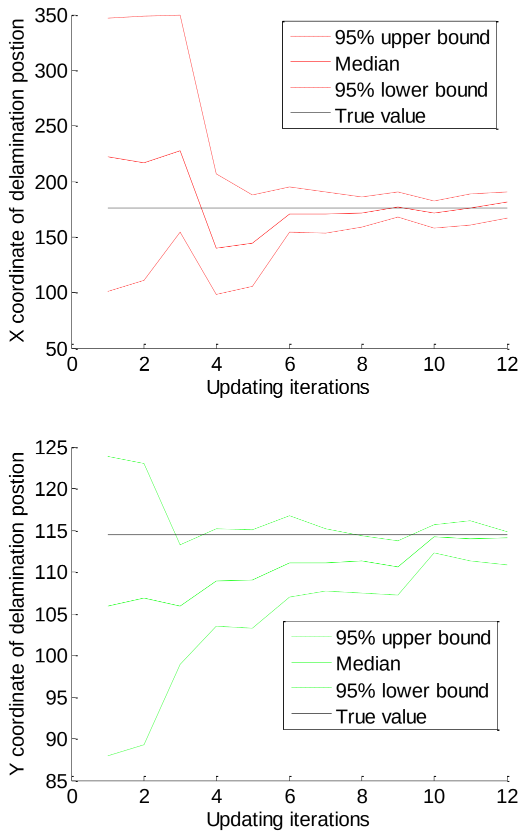
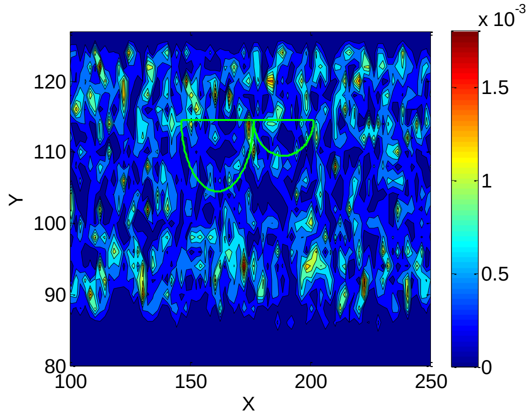
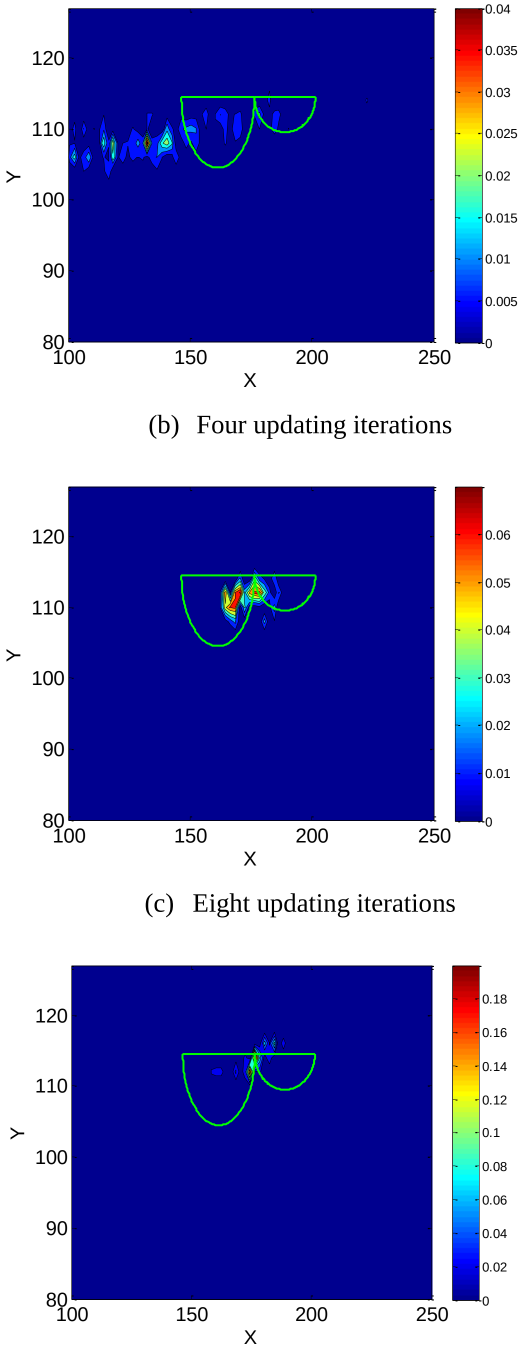


Figure 9. The delamination location updating

As shown here, the estimated delamination location is approaching the true location as more data are used for the updating. Additionally, the uncertainty bounds narrow down as more measurements become available. Fig. 10 illustrates the Bayesian Imaging of the damage probability at each cell of the specimen. It is obvious that the possible delamination area is narrowed down and the probability is increasing as applied more updating iterations. At last, the location with the highest probability is considered as the most probable delamination centers, which is almost the same with the true value as show in Fig. 10(d).



(a) One updating iteration



(b) Four updating iterations

(c) Eight updating iterations

(d) Twelve updating iterations

Figure 10. The damage location probability updating at each cell of the plate.

Simultaneously, delamination size is updated gradually, as shown in Fig. 11. By incorporating the location and size information, the estimated delamination area can be calculated. Fig. 12 gives the comparison between the true delamination from the X-ray images and the updated results using the proposed BIM. Satisfactory agreement is observed.

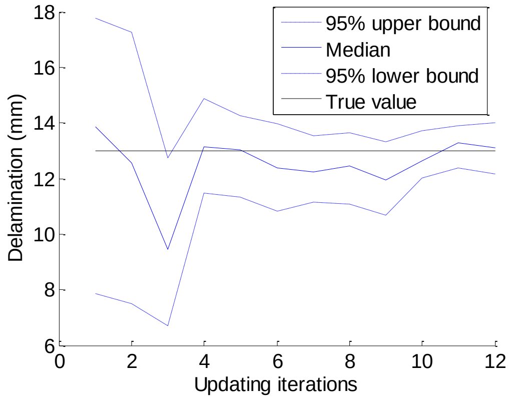
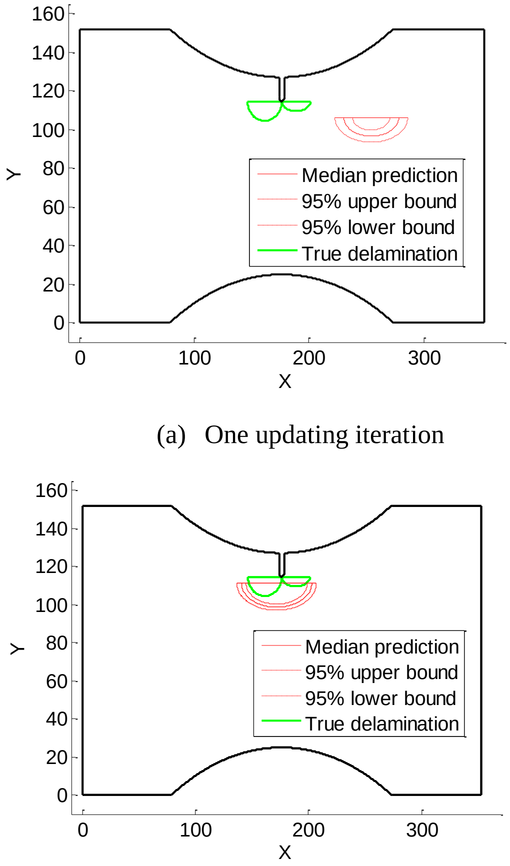
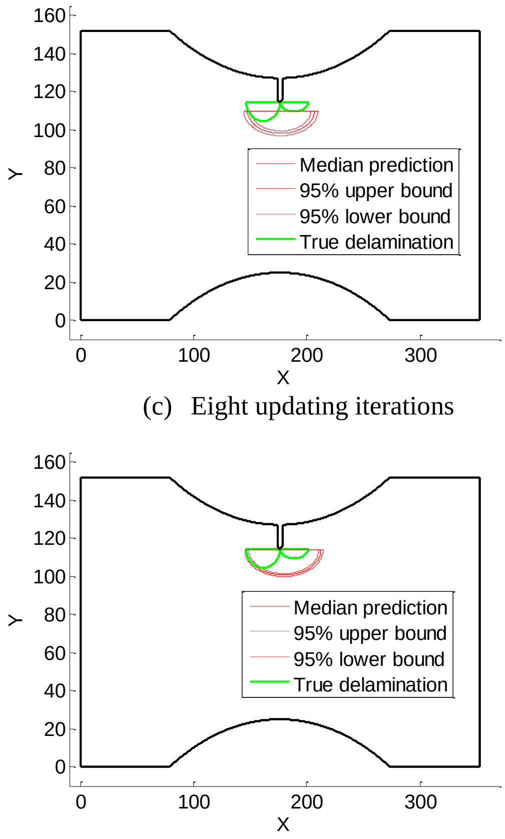


Figure 11. The delamination size updating



(a) One updating iteration

(b) Four updating iterations



(c) Eight updating iterations

(d) Twelve updating iterations

Figure 12. The comparison between the true value and updating result

As given in Fig. 12, the predicted delamination is reconstructed based on the location and size prediction after each updating iteration. More updating iterations means more information is incorporated in the Bayesian updating. At the same time, the uncertainty bound is decreased as applied more updating iterations, which is consistent with the result given in Fig. 9 and Fig. 11.

5. Conclusion

In this paper, a probabilistic damage size and location updating algorithm is proposed, which incorporates the Lamb wave based signal features into the Bayesian updating framework. The proposed method is validated by experimental measurements from X-ray images. Based on the results obtained above, several conclusions are drawn:

1. The Lamb wave propagation based SHM method is able to capture the delamination size and location information
2. The correlation coefficient and the phase change in the received signal are more sensitive to the damage location, but they are not very sensitive when the damage is far away from the sensor path.
3. Bayesian updating can represent and manage the damage detection uncertainties, including both modeling uncertainty and measurement uncertainty. Probabilistic estimation of damage size and location can be obtained and the Bayesian image is constructed based on the probability of each cell.

Currently, only two feature are utilized in the Bayesian updating. Further efforts are required to incorporate more features to give a better detection performance. Simultaneous, More parameters need to be found to characterize the delamination besides size and location. Other possible regression models needs further investigation to cancel the uncertainties between different specimens. Additionally, an irregular delamination shape needs further investigation, which can be done by attaching more sensors on the target region.

Acknowledgement

The research reported in this paper was partially supported by the NASA through Global Engineering and Materials, Inc. (GEM) under the project NNX12CA86C. The support is gratefully acknowledged.

Nomenclature

a	delamination size
x'	Measurement
x_0	x coordinate of the delamination center
y_0	y coordinate of the delamination center
β_{ij}	Regression coefficients
ϵ	Measurement noise
θ	Updating parameters
τ	Model uncertainty

References

Adam, M. T. (2002). "G104-A2L Guide for estimation of measurement uncertainty in testing." American Association of Laboratory Accreditation Manual: 10-18.

Bell, S. (2001). "A Beginner's Guide to Uncertainty of Measurement." The National Physical Laboratory 2: 9-16.

Cheng, L. and Tian, G. Y. (2012). "Comparison of Nondestructive TestingMethods on Detection of Delaminations in Composites." Journal of Sensors 2012(2012): 7.

Constantin, N., Sorohan, S. and Gavan, M. (2011). "Efficient and low cost PZT network for detection and localizaiton of damage in low curvature panels." Journal of Theoretical and Applied Mehanics 49(3): 685-704.

Cowles, M. K. and Carlin, B. P. (1996). "Markov Chain Monte Carlo Convergence Diagnostics: A Comparative

Review." Journal of the American Statistical Association 91(434).

Fort, G., Moulines, E. and Priouret, P. (2012). "Convergence of adaptive and interacting Markov chain Monte Carlo algorithms." Annals of Statistics 39(6): 3262-3289.

Giurgiutiu, V., Zagrai, A. and Bao, J. J. (2002). "Piezoelectric wafer embedded active sensors for aging aircraft structural health monitoring." Structural Health Monitoring 1(1): 41-61.

Grimberg, R., Premel, D., Savin, A., Le Bihan, Y. and Placko, D. (2001). "Eddy current hologaphy evaluation of delamination in carbon-epoxy composites." Insight 43(4): 260-264.

Hasting, W. K. (1970). "Monte Carlo sampling methods using Markov Chain and their applications." Biometrika 57: 97-109.

Ji, S., Xue, Y. and Carin, L. (2008). "Bayesian Compressive Sensing." IEEE transaction on signal processing 56(6).

Kazys, R. and Svilainis, L. (1997). "Ultrasonic detection and characterization of delaminations in thin composite plates using signal processing technique." Ultrasonics 35: 367-383.

Koruk, M. and Kilic, M. (2009). "The usage of IR thermography for the temprature measurements inside an automobile cabin." International Communication in Heat and Mass Tansfer 36: 872-877.

Lemistre, M. and Balageas, D. (2001). "Structural health monitoring system based on diffracted Lamb wave analysis by multiresolution processing." Smart materials and structures 10: 504.

Li, Y.-l., Dong, L.-y., Guan, W.-z., Li, Z. and Zhou, L.-y. (2007). "The Application of Bayesian Method in Image Segmentation." ICICIC '07 Proceedings of the Second International Conference on Innovative Computing, Informatio and Control 490.

Mielozyk, M., Krawczuk, M., Malinowski, P., Wandowski, T. and Ostachowicz, W. (2012). "Active therography method for delamination detection and localisation in composite structures." 6th European Workshop on Structural Health Monitoring

Nicolleto, A. and Hola, K. (2010). "X-ray computed tomography vs. metallography for pore sizing and fatigue of cast Al-alloys." Pocedia Engineering 2(1): 8.

Peng, T., He, J., Liu, Y., Saxena, A., Celaya, J. and Goebel, K. (2012). "Integrated fatigue damage diagnosis and prognosis under uncertainties." Annual Conference of Prognostics and Health Management Society.

Peskun, P. H. (1973). "Aptimum Monte Carlo sampling using Markov chains." Biometrika 57: 97-109.

Pickup, L. C., Capel, D. P., Roberts, S. J. and Zisserman, A. (2009). "Bayesian Methods for Image Super-Resolution." THe computer Journal 52(1): 101-113.

Raghavan, A. and Cesnik, C. E. S. (2007). "Review of guided-wave structural health monitoring." Shock and Vibration Digest 39(2): 91-116.

Saxena, A., Goebel, K., Larrosa, C. C., Janapati, V., Roy, S. and Chang, F.-K. (2011). "Accelerated Aging Experiments for Prognostics of Damage Growth in Composite Materials." Proceedings of the 8th International Workshop on Structural Health Monitoring 1: 1139-1149.

Scalea, d., Francesco, L., Robinson, J. S., Tuzzeo, D. and Bonomo, M. (2002). "Guided wave ultrasonics for NDE of aging aircraft components " Proc. SPIE 4704: 123-132.

Sophian, A., Tian, G. Y., Taylor, D. and Rudlin, J. (2001). "Electromag-netic and eddy current NDT: a review." Insight 43(5): 302-306.

Su, Z. and Ye, L. (2009). "Identification of damage using Lamb waves: From fundamentals to applications." London: Springer-Verlag GmbH & Co.: 346.

Wang, C. H., Rose, J. T. and Chang, F.-K. (2004). "A synthetic time-reversal imaging method for structural health monitoring." J. of smart mater. Struct. 13: 413-423.

Zhao, X., Gao, H., zhang, G. f., Ayhan, B., Yan, F., Chiman, K. and Joseph, L. R. (2007). "Active health monitoring of an aircraft wing with embedded piezoelctric sensor/actuator network:I.Defect detection, localization and growth monitoring " Smart MATER. STRUCT. 16: 1208-1217.

Zhou, C., Su, Z. and Cheng, L. (2011). "Probability-based diagnostic imaging using hybrid features extracted from ultrasonic Lamb wave signals." Smart materials and structures 20(12): 125005.

Heat Exchanger Fouling and Estimation of Remaining Useful Life

Tutpol Ardsomang[1], J. Wesley Hines[2], and Belle R. Upadhyaya[2]

[1] Department of Industrial and Systems Engineering, University of Tennessee, Knoxville, TN 37996, USA
tardsoma@utk.edu

[2]Department of Nuclear Engineering, University of Tennessee, Knoxville, TN 37996-2300, USA
jhines@utk.edu
bupadhya@utk.edu

ABSTRACT

One of the challenges in data-driven prognostics is the availability of degradation data for application to prognostic methods. In real process management settings, failure data are not often available due to the high costs of unplanned breakdowns. This research presents a data-driven (empirical) modeling approach for characterizing the degradation of a heat exchanger (HX) and to estimate the Remaining Useful Life (RUL) of its design operation. The Autoassociative Kernel Regression (AAKR) modeling was applied to predict the effect of fouling on the heat transfer resistance. The result indicates that AAKR model is an effective method to capture the HX fouling in the dynamic process. The AAKR residuals were fused to develop a prognostic parameter which was used to develop a General Path Model (GPM) with Bayesian updating. The results demonstrate the successful application of this approach for the heat exchanger RUL prediction.

1. INTRODUCTION

Fouling is defined as the formation of unwanted material deposition on the heat transfer surface. It occurs naturally in heat transfer processes across several types of industries. The fouling problem exists in more than 90 percent (Steinhagen R, 1993) of heat exchangers. The consequences of fouling accumulation include not only the economic loss, but heat transfer efficiency degradation, high flow resistance and pressure drops, and increased safety hazards, such as overheating of the HX surface.

The fouling problem has been the topic of intensive research by several groups. Several methods on the fouling prediction of heat exchanger (HX) have been continuously proposed. Since the fouling progression is a complicated process, influenced by many parameters and not well understood, the available mathematical models do not cover all forms of fouling mechanisms. The lack of physical understanding of the fouling process dynamics is still a crucial issue that needs to be further investigated.

In recent years, empirical modeling approaches such as the Artificial Neural Network (ANN), Support Vector Machine (SVM) and other statistical techniques have been widely applied in several types of industrial processes. With the ability to learn the different operating conditions of the data and to develop the model without any physical understanding of the process, the empirical modeling approaches have become more interesting and promising for the complicated fouling problem. Several publications, that address the issue of heat exchanger fouling prediction, are available (Vasilios, 2012; Lingfang, 2008; Upadhyaya et al., 2004; Ingimundardóttir, 2009). In this project, an AAKR model is selected to develop the fouling prediction model. AAKR is a non-linear, non-parametric model that uses kernel regression to interpolate historical data stored in a memory matrix. The prediction results shows that an AAKR model can be applied appropriately to both dynamic and static heat exchanger processes.

The fouling prediction using various types of empirical modeling has been successfully studied. However, calculating the Remaining Useful Life (RUL) of the HX when the fouling is first noticeable has not yet been fully explored. In this project, the General Path Model (GPM) (Hines and Coble, 2010) is introduced for the heat exchanger's RUL prediction.

The GPM is developed using historical degradation measurements. The degradation path is expressed as the measure of degradation from the time when a fault first occurs to the end of life, which is usually indicated by exceeding a predetermined critical threshold. This threshold

Tutpol Ardsomang et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

can be defined by a performance specification or engineering judgment. It is assumed that component degradation path can be described by some underlying parametric model and there is a unique degradation path for each individual component. The model parameters can be updated using new individual observations and applying Bayesian updating techniques, which allows both current observation and past knowledge to be considered in the model fitting.

The RUL prediction of early-detected fouling in a heat exchanger is performed using the GPM with Bayesian updating techniques. The historical failed degradation paths which were simulated based on the asymptotic fouling behavior were used to develop the model. The unfailed degradation data were also simulated to test the performance and accuracy of the GPM predictions. The results show that the GPM with Bayesian updating gives an effective model to predict the RUL of the heat exchanger.

All analysis in this study was performed using the MATLAB^{TM} software. The AAKR models were developed, tested and validated using the Process and Equipment Monitoring (PEM) toolbox (Hines and Garvey, 2006), developed by the University of Tennessee. The RUL predictions of heat exchanger were developed, using the MATLAB-based Process and Equipment Prognostic (PEP) toolbox (Hines and Garvey, 2011).

2. BACKGROND

This section introduces the basic mechanisms of heat excahnger fouling, HX physical model, AAKR and GPM with Bayesian updating.

2.1 Heat Exchanger Fouling

Fouling is the terminology generally used to describe the accumulation and formation of unwanted material on the material phase interface, which is the cause of the heat transfer capacity deterioration. Fouling is one of the most important problems of the heat transfer equipment such as heat exchangers, boilers, and steam generators. There is considerable interest in this subject because of the detrimental impact on the economy in various industries.

The fouling process is a result of two processes: a deposition of contamination onto the tube walls and a removal of deposition from the tube walls. The rate of fouling deposition growth (fouling resistance or fouling factor, R_f) can be calculated as the difference between the deposition and removal rates as shown in Equation (1).

$$R_f = \Phi_d - \Phi_r \tag{1}$$

The fouling behavior is classified into four categories (Epstein, 1988), as indicated in Figure 1. The delay time t_d is the initial time period where no fouling occurs. The t_d is unpredictable since it appears to be random in nature. However, t_d is generally shorter after the first fouled surface is cleaned. The plots in Figure 1 represent four different types of fouling mechanisms.

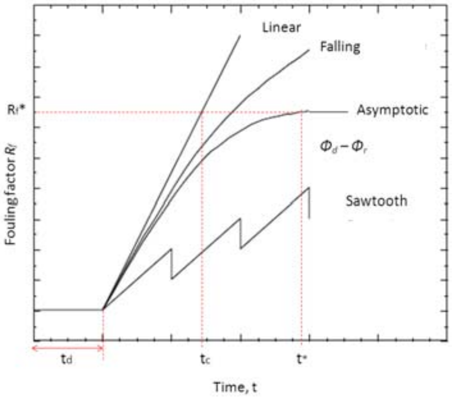


Figure 1. Fouling Curve (Epstein, 1988)

- Linear fouling indicates the constant deposition rates which the difference between Φd and is Φr is constant.
- Falling fouling curve indicates the mass of deposit increase with time but not linearly and does not reach the steady state.
- Asymptotic fouling curve is the most common in the industrial process. The pure particulate fouling also falls into this mode. It indicates the rate of fouling gradually decreasing over time and reaching a steady state eventually, when Φ_d is equal to Φr. The fluid velocity causes a shear stress at the fouling layer that removes some of the particulates. As the fouling layer becomes thick, the fluid velocity increases, thus increasing the rate of removal of the deposit. The thickness of the final steady-state fouling layer is inversely proportional to the original velocity (Nesta, 2004).
- Sawtooth indicates generally increasing trend punctuated by short periods of decreasing trend due to periodic shedding of fouling deposits

For the asymptotic fouling behavior, the fouling resistance, R_f, can be expressed by

$$R_f = R_f^* (1- e^{-\beta_c\Theta}) \tag{2}$$

where R_f* is the asymptotic fouling resistance which Φ_d is equal to Φr, βc is a decay rate and θ is the time or usage parameter.

2.2 Heat Exchanger Physical Model

The shell and tube design is one of the simplest and popular HX types, which can be found in most process plants. It consists of a series of tubes, called the tube bundle, which contain the fluid that must be either heated or cooled. The second fluid flows over the tube in the shell side which can either supply or remove the heat. Figure 2 is a diagram of a simple shell and tube heat exchanger which was used in this study.

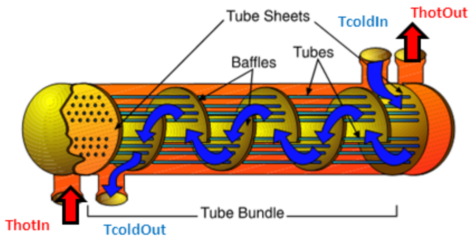


Figure 2. Shell and Tube Heat Exchanger

The traditional heat balance model is utilized based on the assumption that the amount of heat given up by the hot fluid is equal to the amount of heat received by the cold fluid, as shown in the Equation (3).

$$\dot{Q} = \dot{m}_h C_h \Delta T_h = \dot{m}_c C_c \Delta T_c \qquad (3)$$

where ṁ is mass flow rate, C_p is specific heat capacity and ΔT is temperature difference. In general, there are two physical models which represent the heat exchanger: the Log Mean Temperature Difference (LMTD) method and the Effectiveness-NTU method.

The heat transfer can also be expressed by

$$\dot{Q} = U\,A\,\Delta T_{LMTD} \qquad (4)$$

Where U is the overall heat transfer coefficient, A is the heat transfer area, ΔT_{LMTD} is the log-mean temperature difference for the heat exchanger, and is given by (Holman, 1981)

$$LMTD = \Delta T_{LMTD} = \frac{\Delta T_1 - \Delta T_2}{ln(\frac{\Delta T_1}{\Delta T_2})} \qquad (5)$$

If the heat exchanger is the concurrent flow type:

$$\Delta T_1 = T_{h_{in}} - T_{c_{in}} \;,\; \Delta T_2 = T_{h_{out}} - T_{c_{out}} \qquad (6)$$

For the counter current flow heat exchanger:

$$\Delta T_1 = T_{h_{in}} - T_{c_{out}} \;,\; \Delta T_2 = T_{h_{out}} - T_{c_{in}} \qquad (7)$$

The reciprocal of UA is called the overall thermal resistance, which increases in proportion to the fouling deposition amount. From Equation (4), the overall thermal resistance can be derived as

$$\frac{1}{UA} = \frac{\Delta T_{LMTD}}{\dot{m}_h C_h \Delta T_h} = \frac{\Delta T_{LMTD}}{\dot{m}_c C_c \Delta T_c} \qquad (8)$$

The effectiveness-NTU (Holman 1981) method is used when the information is insufficient to calculate the log-mean temperature (LMTD); for instance, when the fluid output temperatures are unknown. Based on the effectiveness of the heat exchanger, it can be defined as the rate between the actual heat transfer and the maximum possible heat transfer which can be hypothetically obtained in a counter-flow heat exchanger.

$$e = \frac{Q}{Q_{max}} \qquad (9)$$

It is noted that the fluid will experience the maximum possible temperature difference, which is the difference between inlet of primary side and the inlet of secondary side ($T_{h_{in}} - T_{c_{in}}$).

$$\dot{Q}_{max} = (\dot{m} \times c_p)_{min}\left(T_{h_{in}} - T_{c_{in}}\right)$$

$$= \dot{m}C_{\min}\left(T_{h_{in}} - T_{c_{in}}\right) \qquad (10)$$

$C_{\min}$ is the heat capacity rate computed by the multiplication of mass flow rate and minimum specific heat, which is either of the hot or the cold fluid. From equations (5), (8) and (9), the effectiveness equation is

$$e = \frac{C_h(T_{h_{in}} - T_{h_{out}})}{C_{min}(T_{h_{in}} - T_{c_{out}})} \quad \text{or} \quad e = \frac{C_c(T_{c_{in}} - T_{c_{out}})}{C_{min}(T_{h_{out}} - T_{c_{in}})} \qquad (11)$$

Where C_h and C_c are the heat capacity rate of hot and cold side, respectively, and are computed from the multiplication of mass flow rate and specific heat capacity. The Number of Thermal Units (NTU) is expressed as

$$NTU = \frac{UA}{C_{min}} \qquad (12)$$

2.3 Autoassociative Kernel Regression

AAKR is a type of kernel regression modeling which is a non-parametric, empirical modeling technique that uses historical, fault-free observations to correct any errors present in current observations. The term "Autoassociative" means that the model inputs and outputs are the same variables. Autoassociative is the common empirical architecture, which all variables input to the model are estimated. It is useful for monitoring equipment or system with a high degree of correlation between variables.

Basically, an AAKR is constructed by putting the non-faulty data into the memory matrix. An AAKR model will learn the relationships from all the variables. When the new data come in, the model will interpolate between those new data and the memory matrix using kernel regression. If the inputs contain the error either by noise, instrument drift or process drift, the model is expected to predict what the normal value should be. The residual between the predicted and measured values can be monitored to detect anomalies in instrument channels and the process.

In AAKR a set of new measurements and a set of prototypical measurements are compared based on distance operator. First, the exemplar or memory vectors used to develop the empirical model are stored in a matrix *X*, where X_{ij} is the i^{th} observation of the j^{th} variable. For n_m observations of *p* process variables, this matrix is expressed as

$$\mathbf{X} = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,p} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n_m,1} & X_{n_m,2} & \cdots & X_{n_m,p} \end{bmatrix} \tag{13}$$

A query vector is the vector of process variable measurements represented by a *1xp* vector

$$\text{x} = [\text{x}_1 \ \ \text{x}_2 \ \ldots . \ \text{x}_\text{p}] \tag{14}$$

The distance between a query vector and each of the memory vectors is computed. The most common function is the Euclidian distance

$$d_j = \sqrt{\sum_{i-1}^{n}(X_{j,i} - x_i)^2} \tag{15}$$

This calculation is repeated for each of the n_m memory vectors, resulting in an n_m *x 1* matrix of distances (*d*). The distances are transformed to similarity measures used to determine weights by evaluating the Gaussian kernel with a bandwidth, *h*. The bandwidth is optimized using a cross validation technique to minimize the prediction error. This regularization is necessary for ill-posed problems to minimize prediction noise and improve repeatability [Hines 2005].

$$w_i = K_j(d,h) = \frac{1}{\sqrt{2\pi h^2}}\exp\left(\frac{-d_j^{\,2}}{2h^2}\right) \tag{16}$$

After achieving the optimal *h* bandwidth, the prediction $\hat{y}_p$ is obtained by a weighted linear (W_i) combination of the similar memory vectors expressed by the equation

$$y_p = \frac{\sum_{i=1}^{n_m} w_i \cdot x_i}{\sum_{i=1}^{n_m} w_i} \tag{17}$$

2.4 General Path Model with Bayesian Updating

Since the degradation of the heat exchanger performance is measurable, the condition based prognostics is the most appropriate approach. The GPM is an example of degradation modeling developed by Lu and Meeker in 1993. GPM is based on the available degradation measurements to estimate Time to Failure (TTF) distribution. In fact, useful information underlying in degradation measurements may result in a better reliability prediction. The GPM analysis begins with the assumption that individual equipment will fall into the same underlying functional form of the degradation path under the specific failure mode.

Basically, degradation measurements show the degradation paths (or degradation signals) to the end of life which is usually indicated by the crossing of a predetermined critical threshold. However, it is not necessary that all units have to be run to failure, unfailed or right censored data also contain useful information for GPM prediction. Another assumption for GPM is that component degradation can be described by some underlying parametric model, based upon physical models or from historical degradation data, and there is a unique degradation path for each individual component. To estimate failure times using a GPM, degradation paths need to be extrapolated to the failure threshold. Then the observed degradation paths will form the TTF distribution. The degradation of the i^{th} unit at time t_j is expressed by equation 18.

$$y_{ij} = \eta(t_j,\Phi,\Theta_i) + \varepsilon_{ij} \qquad i = 1,2,3,....n \tag{18}$$

where η is a GPM function, t_j is time of the j^{th} measurement or inspection, Φ is a vector of fixed-effect parameters, θ_i is a vector of random effects parameters for individual i^{th} component, and ε_{ij} is assumed the normal distribution with mean zero and the standard measurement error term $\text{N}(0,\sigma=\varepsilon^{\wedge 2})$. The model parameters are estimated from the historical data. This degradation path model, y_i, can be extrapolated to the failure threshold to estimate the component's time of failure.

The Bayesian technique can be combined with a GPM model to predict the RUL. Bayesian updating is a method that allows the prior information to combines with the new observations to update model parameter predictions. This method will allow both current observation and past knowledge to be considered in model fitting

A linear regression model, $Y = \beta X$, is considered to be the simplest approach to generate the model parameters. β is the vector of parameters:

$$\beta = (X^T \textstyle\sum_y^{-1} X)^{-1} X^T \textstyle\sum_y^{-1} Y \tag{19}$$

Where X is matrix of time, Y is vector of degradation measure and $\sum_y$ is the variance-covariance noise matrix, which represents the accuracy of each entry in the Y-vector.

In fact, the linear regression model is linear-in-parameters which can be populated with any function of degradation measures such as quadratic, exponential, sinusoidal, etc.

To include Bayesian updating, the prior information (model parameter) is treated as one additional data point to the Ordinary Least Square (OLS) solution, matrix X is appended with an identity matrix I_k and matrix Y is appended with a priori value of the model parameter. The variance-covariance matrix is also added with a final row and column of zeros, with the variance of a priori information in the diagonal element, see the equation 20

$$\mathrm{y}_* = \begin{bmatrix} \mathrm{y} \\ \mathrm{b}_0 \end{bmatrix}, X_* = \begin{bmatrix} X \\ I_k \end{bmatrix}, \Sigma_* = \begin{bmatrix} \Sigma_\mathrm{y} & 0 \\ 0 & \Sigma_\mathrm{b} \end{bmatrix} \qquad (20)$$

It is assumed that the noise in the degradation measurements is constant and uncorrelated across observations of y. This allows the variance-covariance matrix to be a diagonal matrix consisting of noise variance estimates and a priori knowledge variance estimates, which simplified computer implementation. After a priori knowledge is used in conjunction with n current data observation, the posterior estimated parameters become the new estimated parameters if more new data were obtained. The variance of the new data is estimated as

$$\sigma_{post}^2 = \left(\frac{1}{\sigma_{prior}^2} + \frac{n}{\sigma_y^2} \right)^{-1} \qquad (21)$$

There are two pieces of information, the prior and the data, used to form the posterior estimation. The weighting of these information depend on the variance of the prior such as the variance or uncertainty of the data and the amount of data. In other words, the prior b_0 will have large weighting if the variance of the prior is small. However, the data will be weighted more heavily when new data are collected.

3. Experimental setup

The current experimental setup was designed based on the previous experiment (Upadhyaya et al., 2004). The heat exchanger test bed is a simple two sided loop system. The exchanger, API Basco HT, is a shell-and-tube type exchanger with brass shell, internal brass tube sheets, and 64 x ¼"copper tubes. The hot fluid passes through the tube while cold fluid passes through the shell. A 15-gallon tank holds the contaminated water used to accelerate the fouling process. A drain system utilizing a ball valve is placed between the tank and pump inlet to facilitate draining the system for cleaning. Three 1500 watt immersion heaters are inserted in the tank to generate heat up to 4,000 Watt into the water. A 0.5 HP centrifugal pump is used to pump water through the closed loop with the maximum flow rate of 40 GPM at head pressure 20 ft and 5 GPM at 80 ft as the minimum. Figure 3 is a photograph of the heat exchanger test bed.

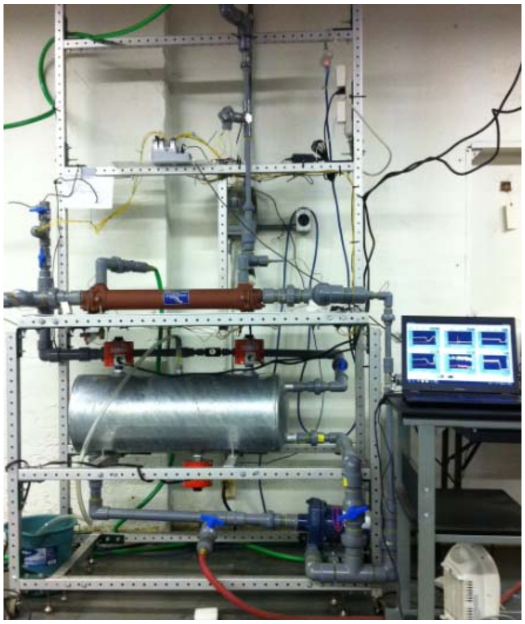


Figure 3. Heat Exchanger Test Bed.

Referring to the Piping and Instrument diagram (P&ID) in Figure 4, the hot water is first pumped to a T-connector to split the water flow. One is the bypass pathway. The main function is to control pressure by XV-2. The second pathway is the main line to the heat exchanger and return to the tank. The flow control valve FV-1, needle type, is installed to control the flow in the main line where the flow rate is measured by the turbine flow meter with F to V converter (Fhot). The water flow rate ranges between 0 to 7.5 gallons per minute (GPM). Two thermocouples, type T, (Thot-1, Thot-2) and two pressure sensors (Phot-1, Phot-2) are installed at inlet and outlet of the heat exchanger to measure the temperature and pressure. XV-1 and XV-3 are utilized as an air vent to relieve any air bubbles initially developed in the system. The four thermocouples, type K, (TZ-1 to TZ-4) are placed along one of the tubes of the heat exchanger to extract more valuable information.

The shell side of the system is an open loop system with the water supply varied between 0 to 7.5 GPM from the facility internal plumbing. A hose transports the water to the shell side of the HX. This water flow rate (Fcold) is measured by an identical flow meter. The inlet and outlet temperatures are measured by thermocouples Tcold-1 and Tcold-2, respectively, on the shell side of the heat exchanger.

To increase the fouling rate naturally, Kaolin clay is added to the water in the tube side of the exchanger. Kaolin, produced by Thiele Kaolin Company, is appropriated for the small-scale shell and tube heat exchanger. The particle size is less than 2 micro millimeters with 98 percent of dry clay tested at 20 percent solids and pH 6.8. The concentration of Kaolin in the heat exchanger is approximately 2,800 ppm.

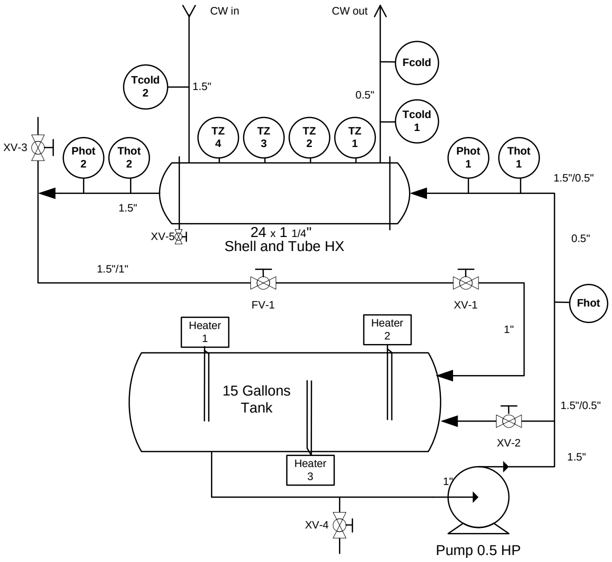


Figure 4. P&ID of Heat Exchanger Test Bed

4. RESULTS AND ANALYSIS

The first part of this section presents the fouling prediction using AAKR models. The fouling data are then used to establish the HX degradation parameters in the GPM to predict the RUL

4.1 Fouling prediction

The heat exchanger was operated under accelerated fouling (due to Kaolin) until a steady-state is reached. The first experiment was run for 250 hours. After cleaning, the second experiment lasted 380 hours. In the second experiment, the process was disturbed by turning one heater off at the end of the data. The purpose is to test the ability of the model to detect any changes in the dynamic process. The fouling prediction is expected not to have this disturbance under normal operation. Figure 5 shows the plots of some of the process variables during accelerated fouling.

Several models based on different variables were investigated. The best prediction was from the group of variables based on the physical model, the overall thermal resistance (Equation 8), which consists of 5 variables, dtcold, dthot, LMTD, Fhot and Fcold. Fcold was removed because of constant value which is useless for the model. Figure 6 illustrates fouling data for this variable group.

In Figure 6, only one variable that is affected by the fouling is the LMTD which is used for fouling prediction. The process disturbance also has an influence on the LMTD.

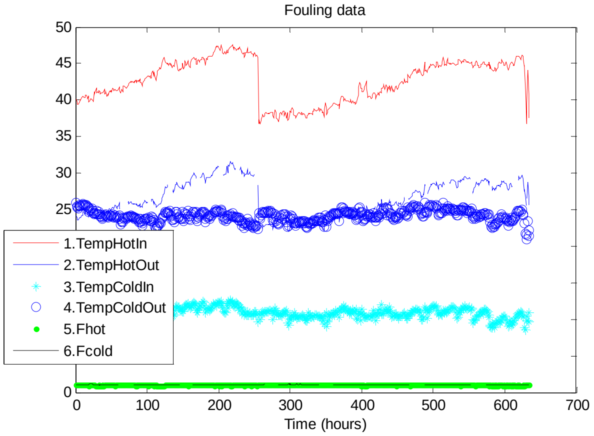


Figure 5. Fouling data

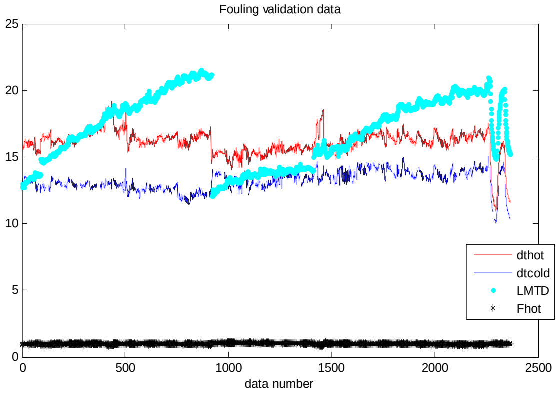


Figure 6. Fouling data for AAKR model development.

The results shown in Figure 7 indicate that the model can predict the fouling and has ability to detect the process disturbance at the end of fouling validation data. However, the prediction is not perfect since the fouling prediction is still under the influence of the process disturbance. The same experiment was run several times under the different operating condition and the prediction data will be then used in the next section.

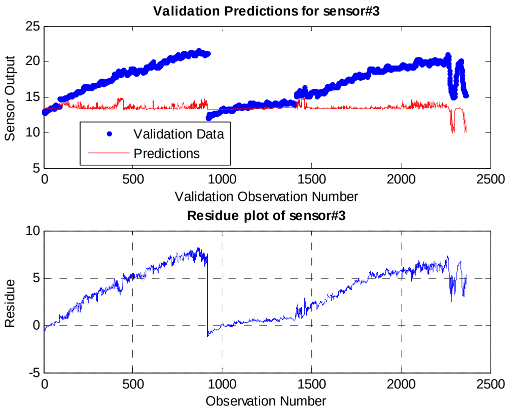


Figure 7. Prediction and Residual plot

4.2 Remaining Useful Life Prediction by GPM

The fouling prediction data from the previous section are used as the degradation path in GPM. The degradation path is defined as the measure of fouling from the beginning to the steady state until the asymptotic behavior. In this study, degradation paths which were from the AAKR model prediction and also simulation were scaled from zero to one hundred percent. Figure 8 presents the degradation paths used in the GPM model for two different conditions.

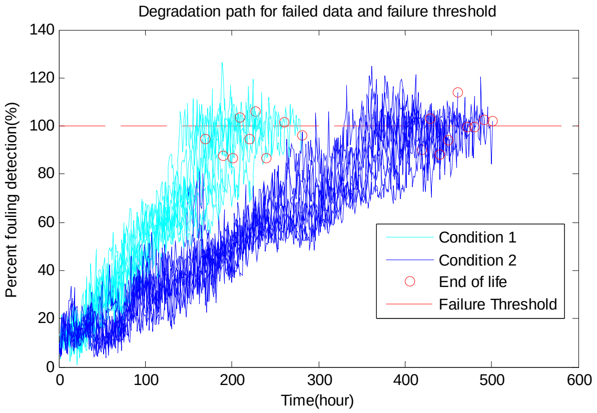


Figure 8. Degradation path from fouling prediction

Figure 8 shows that there are two groups of degradation paths. The first group consisting of 9 paths was operated with 1 GPM flow rate at both primary and secondary side. Nine degradation paths in the second group were operated with 3 GPM flow rate at primary side and 1 GPM at secondary side. The failure threshold is defined as the steady state of the fouling deposition or at 100%. In fact, the experiments were not stopped immediately after fouling reached 100%, but were continued for a period of time to ensure that the fouling reached the steady state.

The degradation measures were fit with the parametric models through the linear regression process. In this research, various models such as linear, exponential, quadratic were investigated. The model accuracy is defined by the average of Mean Square Error (MSE) of the model prediction for those degradation paths. Table 1 presents the result of the model fitting from both fouling condition.

Table 1. Model Fitting and Mean Square Error

Model	**MSE**	
	Condition 1	**Condition 2**
Linear	48.9	67.5
Quadratic	49.0	**67.1**
Cubic	**47.5**	72.0
Exponential	168.3	179.1

Next, the GPM with Bayesian updating was applied for unfailed fouling data to determine the RUL. The critical failure threshold and essential parameters such as regression coefficient and noise estimation from the failed data were used to determine the time to failure (TTF). The TTF is the time that the degradation measure reaches the critical threshold and the RUL is the period between the current time to TTF.

It is assumed that unfailed fouling data act like the prior degradation model. Those data will be used to generate the path and extrapolate to the failure threshold. Since the actual failure times for these units are known, the accuracy of the model can be obtained. See Figure 9 for the example of unfailed fouling degradation paths from both conditions.

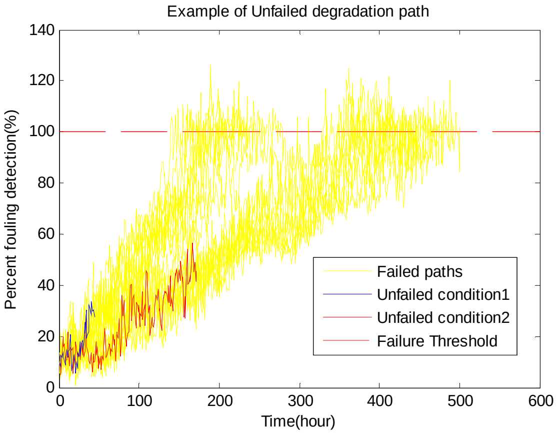


Figure 9. Example of unfailed fouling degradation paths

The RUL prediction was calculated using different parametric models such as linear, quadratic for each operating condition. The result was compared with the RUL prediction without reference to the operating condition. See Table 2 for a summary of results.

Table 2. TTF Prediction and Error

Unfail Path number	Current time	Actual TTF	Predicted TTF [hours]			
			With Stress concern	Error	W/O Stress concern	Error
1	45	150	129.8	13.5	200	33
2	70	188	153.8	18.6	229	21
3	83	206	193.7	5.9	262	27
4	135	400	464.3	16.1	500	25
5	172	414	404.5	2.3	425	3
6	210	459	445.2	3.1	473	3
Average Error				9.8		18.6

The results in Table 2 show that the GPM with Bayesian updating works well for this application. The average error is approximately 9.8 hours. However, it shows that knowledge of the stress or operating condition is important for improved prediction accuracy. The average error from the GPM with a separate operating condition half that of the GPM without knowledge of the operating condition. This example shows that understanding future operational stresses is extremely important for the accurate RUL predictive capabilities. Additional, understanding the relationship between the stresses and RUL estimates could be used to guide operations in order to survive to a maintenance opportunity.

4. CONCLUDING REMARKS

This research consists of two major parts: 1). monitoring the HX fouling using an AAKR model, and 2). predicting the RUL using a GPM with Bayesian updating. The data used in the analysis were collected from an experimental HX test bed which was operated with various operating conditions leading to different fouling rates.

The AAKR model developments were used for fouling prediction. The process disturbance was put into the validation data in order to test its effect on fouling prediction performance. Several models based on different measured and calculated variables were tested. The best model, which had the least influence from the process change, was from the model developed by the group of variables based on the physical model using overall thermal resistance.

A GPM with Bayesian updating, was developed and applied for the RUL prediction of fouling in the HX using the degradation data from the AAKR model. The result shows that this method is appropriate for use in the HX RUL application, which provides usable accuracy and errors of less than 10%. Furthermore, this research shows the importance of understanding future stress conditions to achieve high accuracy prognostic predictions.

ACKNOWLEDGMENTS

This research is being performed using funding received from the DOE Office of Nuclear Energy's Nuclear Energy University Programs under grant DE-AC07-05IDPS07-07ID14517 with the University of Tennessee.

NOMENCLATURE

AAKR	Autoassociative Kernel Regression
GPM	General Path Model
HX	heat exchanger
LMTD	Log Mean Temperature Difference
R_f	the rate of fouling, fouling resistance or fouling factor
RUL	Remaining Useful Life
TTF	Time to Failure
U	Overall heat transfer coefficient

REFERENCES

Lu C. J., & W. Q. Meeker, *Using degradation measures to estimate a Time-to-Failure distribution*, Department of Statistics and Center for Nondestructive Evaluation, Iowa State University, May 1983.

Epstein, N.,"General Thermal Fouling Models," L.F. Melo, T.R. Bott and C.A. Bernardo, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 15-30, 1988.

Holman, J. P., *Heat Transfer*, 5th Edition, McGraw-Hill, New York, 1981.

Ingimundardóttir H., & S. Lalot, "Detection of fouling in a cross-flow heat exchanger using wavelets" International Conference on Heat exchanger Fouling and Cleaning, 2009.

Hines J. W., & J. Coble, "Applying the General Path Model to Estimation of Remaining Useful Life," International Journal of Prognostics and Health Monitoring (IJPHM), 2010.

Hines, J. W., & J. Coble, *Process and Equipment Prognostics Toolbox*, Tutorial version 1.0, University of Tennessee, 2011.

Hines J. W. & D. Garvey, *Process and Equipment Monitoring Toolbox Tutorial*, Nuclear Engineering Department, University of Tennessee, 2006.

Hines J. W. & A. Usynin, "MSET Performance Optimization Through Regularization", *Nuclear Engineering and Technology*, Vol. 37, No. 2, April 2005, pp 177-184.

Hines J. W., D. Garvey, & R. Seibert, "On-Line Sensor Calibration Monitoring Challenges and Effective Monte Carlo Based Uncertainty Estimation," The University of Tennessee, 2003.

Steinhagen, R, H.M. Steinhagen, & K. Maani, "Problems and Costs due to Heat Exchanger Fouling in New Zealand Industry," Heat Transfer Engineering, Vol. 14, No. 1, pp. 19-30, 1993.

Lingfang S., *et al.*, "Research on the fouling prediction of heat exchanger based on support vector machine", International Conference on Intelligent Computation Technology and Automation (ICICTA), 2008.

Katsikis, V. N., "MATLAB- a fundamental tool for scientific computing and engineering application," Vol. 3, Chapter 3, Intech, September 2012.

Upadhyaya, B. R., J. W. Hines, *et al.*, *On-Line Monitoring and Diagnostics of the Integrity of Nuclear Plant Steam Generators and Heat Exchangers*, Final Report: Volume 1, *Experimental and Hybrid Modeling Approach for Monitoring Heat Exchanger System Performance,* prepared for the DOE-NEER Program by the University of Tennessee, Knoxville, Report No. DE-FG07-01ID14114/UTNE-07, September 2004.

Nesta, J., “Reduce fouling in shell-and-tube heat exchangers,” Hydrocarbon Processing, pp. 77-82, July 2004.

BIOGRAPHIES

Tutpol Ardsomang is pursuing his MS degree in Reliability and Maintainability Engineering with Industrial and Information engineering concentration from the University of Tennessee, Knoxville. He received his Bachelors in Instrumentation Engineering from the King Mongkut's Institute of Technology Ladkrabang and worked for 6 years with SCG chemical Co., Ltd in Thailand. He is now conducting research about the heat exchanger fouling prediction using the empirical modeling and prognostic techniques under Dr. Hines’s.

Dr. J. Wesley Hines currently the Nuclear Engineering Department Head at the University of Tennessee. He is a Professor of Nuclear Engineering and the director of the Reliability and Maintainability Engineering program. He is a past nuclear qualified submarine officer and received an MBA and both an MS and Ph.D. in Nuclear Engineering from The Ohio State University. He has been with the University for 17 years. Dr. Hines teaches and conducts research in artificial intelligence and advanced statistical techniques applied to process diagnostics, condition based maintenance, and prognostics. He has authored over 250 papers and has several patents in the area of advanced process monitoring and prognostics techniques.

Dr. Belle R. Upadhyaya received the Ph.D. degree in Engineering Sciences (Systems Science) from the University of California at San Diego. He is currently a professor of Nuclear Engineering at The University of Tennessee, Knoxville. His research and teaching interests include instrumentation and controls, reactor dynamics, advanced digital signal processing, power and process plant monitoring and diagnosis, autonomous and fault-tolerant control of small modular reactors, nuclear desalination, sensor placement strategies, accelerated aging of detectors and equipment, and reliability and maintainability engineering. He has published over 320 articles in scientific journals and conference proceedings, chapters in handbooks, and is the author or co-author of over 130 research reports. Dr. Upadhyaya is an elected Fellow of the American Nuclear Society (ANS) and an elected Fellow of the International Society of Automation (ISA)

Practical Use of Accelerated Test Data for the Prognostics Methods

Dawn An[1], Joo-Ho Choi[2], and Nam Ho Kim[3]

[1,2]*Korea Aerospace University, Goyang-si, Gyeonggi-do, 412-791, Korea*
dawnan@ufl.edu
jhchoi@kau.ac.kr

[1,3]*University of Florida, Gainesville, FL, 32611, USA*
dawnan@ufl.edu
nkim@ufl.edu

ABSTRACT

Prognostics is to predict future damage/degradation of in-service systems and the remaining useful life based on the damage data obtained at previous usage. The damage data is of great importance regardless of prognostics methods used, while it is very expensive to obtain the data because of time and cost. Instead, companies frequently use accelerated test data for the purpose of design, which is obtained under much severe operating conditions. This paper presents a method of utilizing accelerated test data for the purpose of prognostics. The uncertainty caused by mapping between nominal and accelerated operating conditions is taken into account using the Bayesian framework. As an example, crack growth data are synthetically generated under over-loaded conditions, which are utilized for both of data-driven and physics-based approaches under different conditions. Using accelerated test data increases prediction accuracy in early stage of physics-based prognostics as well as it covers insufficient data problem of data-driven prognostics.

1. INTRODUCTION

Prognostics is to predict future damage/degradation of in-service systems and the remaining useful life (RUL) based on the damage data obtained at previous usage, whose process is illustrated in Figure 1. Once damage model (black solid curve) is determined based on damage data at previous times under a given usage condition (black dots) or under the various usage conditions (grey dots), RUL which is remaining time/cycles before required maintenance can be predicted by progressing the damage state until it reaches the threshold.

Even though prognostics facilitates condition-based maintenance (Jardine, Lin, & Banjevic, 2006) that is considered as a cost-effective maintenance strategy compared to periodical preventive maintenance, there are several challenges to be viable in practice. One of them to be considered in this paper is a limited number of damage data in-service systems since it is very expensive to obtain the data because of time and cost. Instead, it is easy to obtain accelerated test data since companies frequently use them for the purpose of design. Actually, most studies on prognostics have been conducted using accelerated testing data by considering accelerated conditions as field operation conditions even though the data are obtained under much severe operating conditions. This issue has been recognized by researchers in prognostics field (Celaya, Saxena, Saha, & Goebel, 2011).

Even though the study on life estimation using accelerated test data is mature (Nelson, 1990; Park & Bae, 2010), it is different from prognostics viewpoint since in-service usage conditions are not reflected in their study. Therefore, if accelerated test data are utilized for the purpose of prognostics, more accurate prognostic results can be achieved. This paper presents a method of utilizing accelerated test data to compensate for the insufficient data problem based on a crack growth example. Different scenarios are considered according to presumable four different cases in terms of given information.

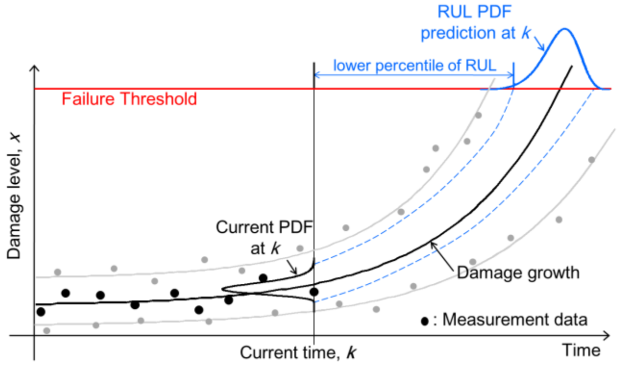


Figure 1. Illustration of prognostics

Dawn An et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The paper is organized as follows: in Sections 2, two prognostics approaches are introduced; and in Section 3 and Section 4, a crack growth example is addressed to suggest how accelerated condition data can be used under different scenarios, followed by conclusions in Section 4.

2. Prognostics Approaches

In general, prognostics methods can be categorized into physics-based (Luo, Pattipati, Qiao, & Chigusa, 2008), data-driven (Schwabacher, 2005), and hybrid (Yan & Lee, 2007) approaches, based on the usage of information. The last approach combines the other two methods to improve the prediction performance. It, however, is not mature yet, and will not be considered in this paper. There are two main differences between data-driven and physics-based approaches; (1) availability of a physical model and (2) usage of training data to identify the characteristic of the damage state. More specific explanation for two approaches is introduced in the following subsections.

2.1. Physics-Based Approaches

Physics-based approaches assume that a physical model describing the behavior of damage is available, and combine the physical model with measured data to predict future behavior of damage and RUL. The process of physics-based prognostics is illustrated in Figure 2, in which the degradation model is expressed as a function of usage conditions U, elapsed cycle or time t, and model parameters $\boldsymbol{\theta}$. The usage conditions and time are given, while the model parameters should be identified. Since behavior of damage depends on model parameters, identifying them is the most important issue to predict future damage state and RUL. In fact, parameter estimation algorithms become criteria to classify physics-based approaches. There are several algorithms such as Kalman filter (KF) (Kalman, 1960), extended Kalman filter (EKF) (Julier & Uhlmann, 2004), particle filter (PF) (Doucet, Freitas, & Gordon, 2001), and Bayesian method (BM) (Kramer & Sorenson, 1988), which are based on the Bayesian inference (Bayes, 1763).

Bayesian inference is a statistical method in which observations are used to estimate and update unknown model parameters in the form of a probability density function (PDF). Bayesian inference is based on the following Bayes' theorem (Bayes, 1763) :

$$p(\boldsymbol{\theta}\,|\,\mathbf{z}) \propto L(\mathbf{z}\,|\,\boldsymbol{\theta})\,p(\boldsymbol{\theta}) \tag{1}$$

where $\boldsymbol{\theta}$ is a vector of model parameters, $\mathbf{z}$ a vector of observed data, $L(\mathbf{z}\,|\,\boldsymbol{\theta})$ the likelihood, $p(\boldsymbol{\theta})$ the prior PDF of $\boldsymbol{\theta}$, and $p(\boldsymbol{\theta}\,|\,\mathbf{z})$ the posterior PDF of $\boldsymbol{\theta}$ conditional on $\mathbf{z}$. The likelihood is the PDF value of $\mathbf{z}$ conditional on given $\boldsymbol{\theta}$, which depends on the observed data. The prior information can be given, assumed, or not considered. The posterior PDF is obtained by multiplying the likelihood and the prior PDF. The posterior PDF at previous time becomes prior PDF at current time as a new data is added, which is called Bayesian updating. The distribution of posterior PDF get narrow through Bayesian updating as shown in Figure 2, which gives more accurate and precise prediction results of damage and RUL.

Among several algorithms, PF is commonly used for prognostics and is considered in this study. PF (a.k.a. sequential Monte Carlo method) estimates and sequentially updates the parameters, in which the posterior distribution is expressed as a number of particles and their weights. More specific explanations can be found in the literature (Doucet et al., 2001; Orchard & Vachtsevanos, 2007; DeCastro, Tang, Loparo, Goebel, & Vachtsevanos, 2009; An, Choi, & Kim, 2013).

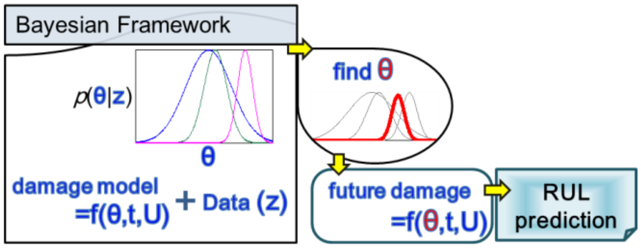


Figure 2. Illustration of physics-based prognostics

2.2. Data-Driven Approaches

Data-driven approaches use information from collected data (training data) to identify the characteristics of damage state and predict the future state without using any specific physical model. There are many different types of algorithm for data-driven approaches, and they are divided into two categories: (1) the artificial intelligence approaches that include neural network (NN) (Chakraborty, Mehrotra, Mohan, & Ranka, 1992; Yao, 1999) and fuzzy logic (Zio & Maio, 2010), and (2) the statistical approaches that include gamma process (Dickson & Waters, 1993), hidden Markov model (HMM) (Rabiner, 1989), and regression-based model such as Gaussian process (GP) regression (Seeger, 2004; Mohanty, Teale, Chattopadhyay, Peralta, & Willhauck, 2007), relevance vector machine (RVM) (Tipping, 2001), least square (LS) regression (Tran & Yang, 2009), etc.

The process of data-driven approaches is illustrated in Figure 3, which are similar to extrapolation based on

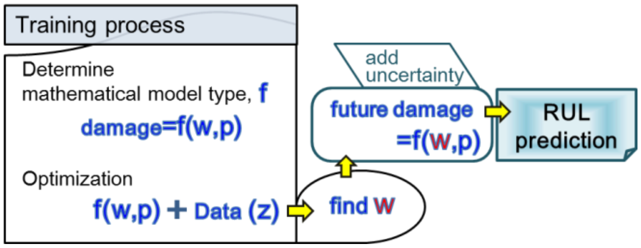


Figure 3. Illustration of data-driven prognostics

regression that contains mathematical model (f), input variables (p), and parameters (w). First of all, mathematical models that represent the relation between input variables and damage state as the output should be determined instead of a physical model. The input variables are not only the parameters shown on x -axis, but also previous damage states. Using damage data themselves as input variables makes a relation between damage data without any other information such as physical model and loading conditions. The usage of input variables, however, is flexible, and loading information can be also added for input variables if it is available, which increase accuracy of prediction with a less number of damage data. Once the mathematical model is determined including input variables, mathematical parameters associated with the model are identified by combining damage data via optimization process. The damage data used for identifying the parameters are called training data, and are obtained under the various usage conditions as shown in grey dotted markers in Figure 1 or at previous times under a given usage condition as shown in black dotted markers in Figure 1. Lastly, the future damage state and RUL are predicted based on the identified parameters and the mathematical model. Uncertainty of prediction can be considered by providing a confidence bound based on the mean square error between training outputs and training data or adding Gaussian noise.

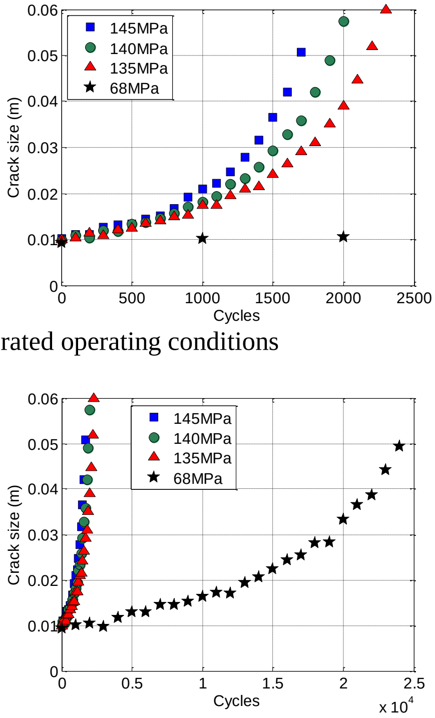
Among several algorithms, NN is most commonly used and is considered in this study. Also, Bayesian NN is employed, which determines distributions of parameters based on Bayes' theorem instead of deterministic values given by optimization process (Freitas, 2003; Neal, 1995). Also, the previous two damage data are used for input variables. In this paper, two types of prediction are considered: short-term prediction and long-term prediction. Short term prediction is to predict just one step ahead; that is, damage cannot be progressed without currently measured data since the previous damage data are required for input variables. On the other hand, long term prediction is to predict multi-step ahead by using the simulation results for the input instead of real measurement data.

3. Crack Growth Example

A crack growth problem in a fuselage panel under repeated pressurization loading is employed, which is based on Paris model (Paris & Erdogan, 1963):

$$\frac{da}{dN} = C\left(\Delta K\right)^{m}, \quad \Delta K = \Delta\sigma\sqrt{\pi a} \tag{2}$$

where a is the half crack size, N is the number of cycles, m and C are damage growth parameters, ΔK is the range of stress intensity factor, and $\Delta\sigma$ is the stress range due to the pressure differential. The synthetic measurement data are generated using Eq. (2) with true damage growth parameters $m_{\text{true}} = 3.5$, $C_{\text{true}} = 6.4\times10^{-11}$, and the initial half crack size $a_0 = 10\text{mm}$. These parameters are only used to generate synthetic data. Three over-loaded conditions, $\Delta\sigma$ =145MPa, 140MPa, and 135MPa and one nominal condition, $\Delta\sigma$ =68MPa are, respectively, considered as accelerated and field operating conditions. Also, random noise is uniformly distributed between $-u$ mm and $+u$ mm, and different levels of noise, 0.7mm and 1.5mm are, respectively, added to accelerated and field operating conditions. Figure 4 shows synthetic data, and three sets of accelerated test data in Figure 4(a) are considered as training data for data-driven approach or additional information for physics-based approach. In other words, these data are used to predict future damage growth of in-service system shown as star markers in Figure 4(b). The way to utilize accelerated test data can be roughly categorized into two. The first case is when other information such as physical model or field operating conditions are available in addition to accelerated test data, and another case is when only accelerated test data are available. These two cases are considered in the following subsections.



a) accelerated operating conditions

b) field operating conditions

Figure 4. Synthetic data under different loading conditions

4. Utilizing Accelerated Test Data

Methods of utilizing accelerated test data are presented in the following subsections. There are four different scenarios according to the availability of physical model and future loading conditions, as listed in Table 1.

Table 1. Four scenarios

	Case 1	Case 2	Case 3	Case 4
Physical model	O	O	X	X
Future loading	O	X	O	X
Available method	Physics-based (PF)		Data-driven (NN)	

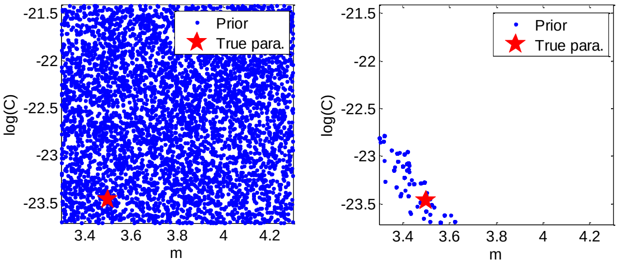
4.1. Case 1: Physical Model and Loading Conditions are Given

When a physical model and future operating conditions are available, the case is the same as physic-based prognostics using Paris model given in Eq. (2). In such a case, damage data under over-loaded loading conditions are not necessary for prognosis. The accelerated test data, however, can be used to improve prognostics results in terms of accuracy and reducing uncertainty in the early stage because the additional data can give prior information for the damage parameters.

In order to illustrate the effect of accelerated test data, the initial distribution of damage parameters of the Paris model in Eq. (2) for generic Al 7075-T651 material (Newman, Phillips, & Swain, 1999) are assumed to be uniformly distributed; that is, $m \sim U(3.3, 4.3)$ and $\log(C) \sim U\left(\log\left(5\times10^{-11}\right), \log\left(5\times10^{-10}\right)\right)$, which are shown in

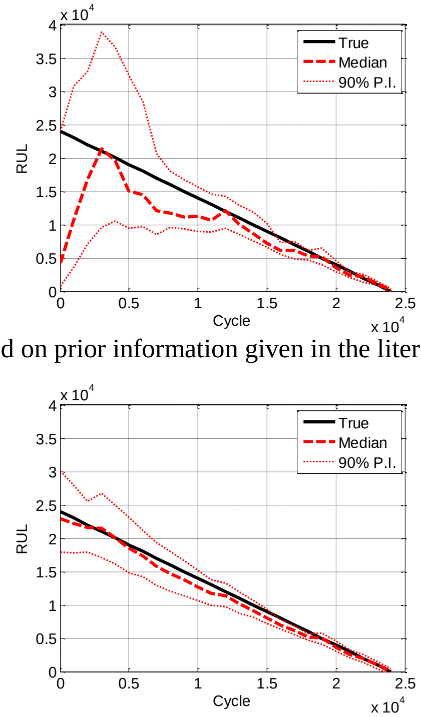
Figure 5(a). The star mark in the figure represents the true value of the parameters, which needs to be found using crack growth measurement data. When accelerated test data are available, this distribution can be narrowed using the Particle Filter method based on Eq. (1). Figure 5(b) shows the narrowed distribution using the three accelerated test data in Figure 4(a). It is noted that the scatter of distribution is much narrower than that of the initial distribution. It is also noted that due to strong correlation between the two Paris parameters, the Bayesian process cannot converge to the true value of the parameters. However, as pointed out by An et al. (An, Choi, & Kim, 2012), the crack propagation behavior will be similar for different combinations of the two Paris parameters. The advantage of accelerated test data is to start with much better initial distribution as in Figure 5(b).

For the purpose of comparison, the two distributions in Figure 5 are used as an initial distribution in PF, and respective prediction results of RUL are shown in Figure 6(a) and (b). Figure 6(b) is the case when the prior information from accelerated test data applied, which shows fairly accurate and precise results even from an early stage. Also, Table 2 shows the difference between Figure 6(a) and (b) quantitatively based on prognostics metrics (Saxena, Celaya, Saha, Saha, & Goebel, 2009). Prognostics horizon (PH) is related to the time when the predicted RUL satisfies given accuracy band, α-λ accuracy is that whether the prediction result satisfies the accuracy zone at given cycle or not, and relative accuracy (RA) and cumulative relative accuracy (CRA) mean that how accurate the prediction results are, in which the performance is better when the values of prognostics metrics are larger.



a) from the literature b) from accelerated test data

Figure 5. Prior distribution



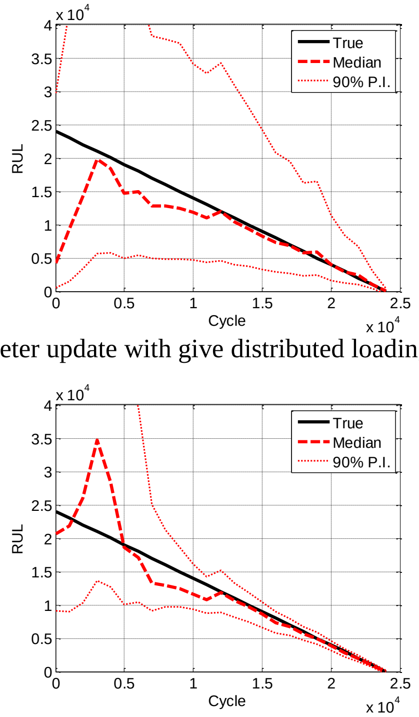
a) based on prior information given in the literature

b) based on prior information from accelerated test data

Figure 6. RUL prediction of Case 1

Table 2. Prognostics metrics

	PH (α =10%)	α-λ accuracy (α =10%, λ =0.5,)	RA (λ =0.5)	CRA
prior from literature	11000	false (0.2504)	0.9161	0.8479
prior from test data	**22000**	**True (0.6046)**	**0.9240**	**0.9285**



a) parameter update with give distributed loading

b) loading update with given parameter from accelerated test data

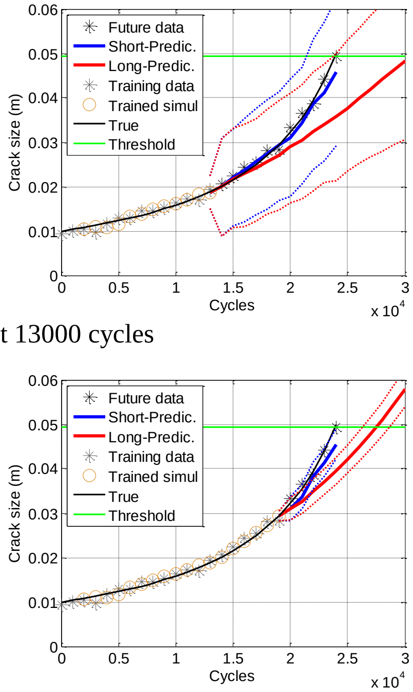
Figure 7. RUL prediction of Case 2

4.2. Case 2: Physical Model is Given but Loading Conditions are Not

When future loading is uncertain, it will make sense to assume its range roughly; let us assume uncertain loading condition is uniformly distributed between 50 MPa and 90 MPa against true loading condition, 68 MPa, while a physical model is available. There are two ways to predict RUL in this case, updating parameters with given uncertain loading condition and updating loading condition with given distributions of parameters.

Updated parameters converge to different distributions from true ones due to different loading conditions between uncertain and true conditions, but prediction results of damage and RUL can be accurate (An, Kim, & Choi, 2013). In this case, it is better not to utilize accelerated test data because the prior information from accelerated test data are not matched with field condition, and wrong prior information does not have a good effect on the prediction results (An, Choi, Schmitz, & Kim, 2011). RUL prediction shown in Figure 7(a) is obtained by updating parameters from the information given in the literature with given loading condition, whose median becomes close to the true RUL after 12000 cycles, but whose prediction interval is very wide due to wide distribution of loading condition.

For the case of updating loading condition, accelerated test data are utilized to obtain prior distribution of parameters, as shown in Figure 5(b). The result is shown in Figure 7(b), in which convergence rate of median is similar to one of Figure 7(a), but prediction interval is much smaller because loading condition is updated as well as updated distributions of parameters are employed. Also, since prior distributions of parameters are obtained from correct conditions (accelerated test), true loading condition can be identified.



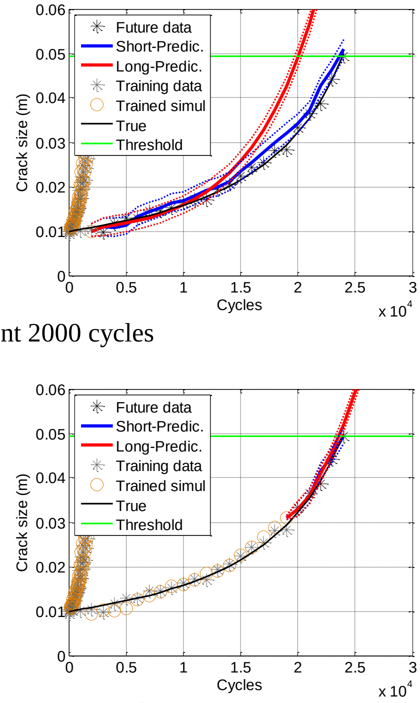
a) current 13000 cycles

b) current 19000 cycles

Figure 8. Damage prediction without accelerated test data

4.3. Case 3: Physical Model is Not Given but Loading Conditions Are

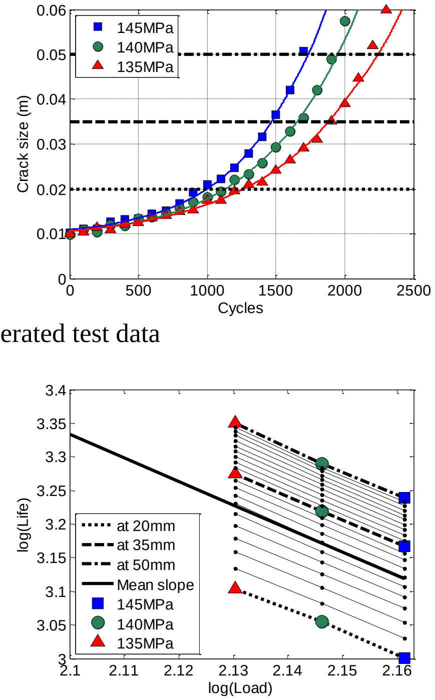
As the reverse of the previous condition, it is considered that the physical model is not available, but future loading conditions are. This case can be handled with NN by adding loading conditions to input variable. First of all, Figure 8 shows damage prediction results without accelerated test data, that is, training data are available only from field operating condition. In this case, short-term prediction becomes valid after 13000 cycles, as shown in Figure 8(a), but long-term prediction are not effective even if data up to 19000 cycles are used, as shown in Figure 8(b). In contrast, the case when the accelerated test data are used as training data gives good results in terms of short-term prediction at the very early cycles, as shown in Figure 9(a). Also, long-term prediction becomes close to the true damage growth with narrow distribution at 19000 cycles, as shown in Figure 9(b) compared to Figure 8(b). This results show that accelerated test data can be used as training data even though the difference between nominal and accelerated operating conditions is big.



a) current 2000 cycles

b) current 19000 cycles

Figure 9. Damage prediction with accelerated test data



a) accelerated test data

b) inverse power model

Figure 10. Linear relation between logarithm of life and load

4.4. Case 4: Physical Model and Loading Conditions are Not Given

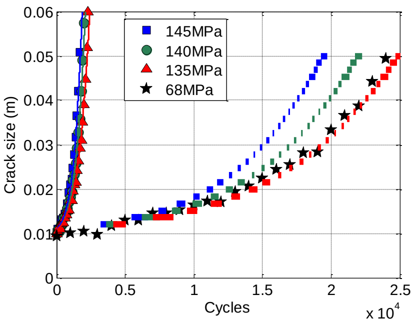
This case is also data-driven prognostics in common with Case 3, but loading condition is not available and cannot be updated since there is no physical model. Many training data under similar usage conditions to field operating conditions are required, but it is very expensive to obtain the data in terms of time and cost. In this context, the way of mapping damage data at accelerated operating conditions to nominal conditions can be a resolution.

Inverse power model is employed for mapping between accelerated conditions data and field operation data, which is widely used to define the relation between loading conditions and system's life in the problems such as an electric insulator, bearing, metal fatigue and has a linear relation between logarithm of life and load (Nelson, 1990):

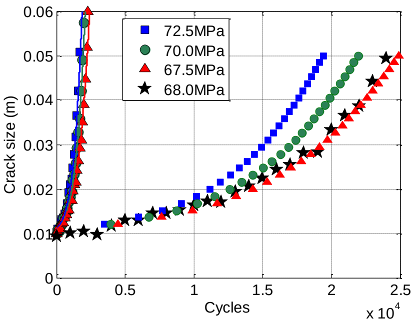
$$Life = \frac{\alpha}{Load^{\beta}}\ ,\ \log\left(Life\right) = \log\left(\alpha\right) - \beta \log\left(Load\right) \qquad (3)$$

where $Life$ is lifespan of a system, and α, β are coefficients, which is be found using life data under at least two different over-loaded conditions. Once coefficients are identified, lifespan under field operating conditions can be calculated by using linear equation in Eq. (3). To apply inverse power model for prognosis, lifespan should be determined according to damage threshold, which is calculated based on a regression model shown as solid curves in Figure 10(a) that are obtained based on NN. In Figure 10(a), dashed-horizontal line illustrates threshold 35mm, and lifespans of system under 145 MPa, 140 MPa, and 135 MPa are, respectively, 1460 cycles, 1650 cycles, and 1880 cycles. Coefficients in Eq. (3) are found by using these three data, and the linear relation is shown as dashed-diagonal line with three different makers in Figure 10(b). Three damage data under similar conditions to field operating conditions are obtained by extrapolating the dashed-diagonal line to half loading of each overloaded conditions. Mapping is to repeat this process according to different thresholds, as many diagonal lines in Figure 10(b) to obtain damage growth data that are available as training data.

By the way, noise in measured data, as shown in Figure 10(a) has an effect on identifying coefficients, and very small changes of the coefficients yield tremendous difference in the mapping results due to logarithm equation. Therefore, uncertainty of coefficients is identified based on Bayesian approach like identifying model parameters and loading condition in Sections 4.1. and 4.2. Since the uncertainty of mapping grows serious when both coefficients are considered as uncertain parameters, only the uncertainty of intercept is taken into account. This is reasonable from two grounds; (1) the slope is the almost same shown in Figure 10(b), and (2) the slope and the intercept have strong linear relation. The deterministic slope is -3.4844 that is the mean value of total 21 slopes from the



a) mapping uncertainty



b) mapping into half-loads

Figure 11. Mapping between nominal and accelerated conditions data

threshold 20 mm to 50 mm, which is illustrated as solid-diagonal line in Figure 10(b) at arbitrary intercept.

Mapping into half magnitudes of three overloaded cases are shown in Figure 11, which is obtained with fixed slope and different intercepts according to different thresholds. The width of horizontal lines in Figure 11(a) represents 90% confidence intervals caused by uncertainties of intercepts. The intervals are very small at the crack size more than 20mm due to deterministic slope and low noise level in accelerated test data. Figure 11(b) shows the median plot of Figure 11(a), and the mapping results at 67.5MPa are very close to the true damage data at 68MPa load. This results show that accelerated condition data can roughly show damage growth and RUL under field operation condition and can be utilized as the training data for nominal condition through proper mapping method.

5. Conclusions

This paper presents the way to utilize accelerated conditions data for the purpose of prognostics by paying attention that very limited number of damage data in-service is available. Four different cases are considered: (1) both physical model and loading conditions are given, (2 and 3) either physical model or loading conditions is given, and (4) neither physical model nor loading conditions is given. Using accelerated test data increases prediction accuracy in early stage in any cases. Especially, proper way for mapping between nominal and accelerated conditions data can cover insufficient data for the last case which is the most practical situation. This approach will be employed for real test data in the near future.

Acknowledgement

This work was supported by a grant from the International Collaborative R&D Program (0420-2011-0161) of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), funded by the Korean government's Ministry of Knowledge Economy.

References

An, D., Choi, J. H., Schmitz, T. L., & Kim, N. H. (2011). In-situ monitoring and prediction of progressive joint wear using Bayesian statistics. *Wear*, vol. 270(11-12), pp. 828-838.

An, D., Choi, J. H., & Kim, N. H. (2012). Identification of correlated damage parameters under noise and bias using Bayesian inference. *Structural Health Monitoring*, vol. 11(3), pp. 293-303.

An, D., Choi, J. H., & Kim, N. H. (2013). Prognostics 101: A tutorial for particle filter-based prognostics algorithm using Matlab. *Reliability Engineering and System Safety*, vol. 115, pp.161-169.

An, D., Kim, N. H., & Choi, J. H. (2013). Options for prognostics methods: A review of data-driven and physics-based prognostics. *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, April 8-11, Boston, MA.

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions*, vol. 53, pp. 370-418.

Celaya, J. R., Saxena, A., Saha, S., & Goebel, K. F. (2011). Prognostics of power MOSFETs under thermal stress accelerated aging using data-driven and model-based methodologies. *Annual Conference of the Prognostics and Health Management Society*, September 25-29, Montreal, Quebec, Canada.

Chakraborty, K., Mehrotra, K., Mohan, C. K., & Ranka, S. (1992) Forecasting the behavior of multivariate time series using neural networks. *Neural Networks*, vol. 5, pp. 961-970.

DeCastro, J. A., Tang, L., Loparo, K. A., Goebel, K., & Vachtsevanos, G. (2009). Exact nonlinear filtering and prediction in process model-based prognostics. *Annual Conference of the Prognostics and Health Management Society*, September 27 - October 1, San Diego, CA.

Dickson, D. C. M., & Waters, H. R. (1993). Gamma processes and finite time survival probabilities. *Astin Bulletin*, vol. 23(2), pp. 259-272.

Doucet, A., Freitas, N. D., & Gordon, N. J. (2001). *Sequential Monte Carlo methods in practice*. NY: Springer-Verlag.

Freitas, J. F. G. (2003). *Bayesian methods for neural networks*. University of Cambridge, Cambridge, United Kingdom.

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, vol. 20(7), pp. 1483-1510.

Julier, S. J., & Uhlmann, J. K., (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, vol. 92(3), pp. 401-422.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transaction of the ASME-Journal of Basic Engineering*, vol. 82, pp. 35-45.

Kramer, S. C., & Sorenson, H. W. (1988). Bayesian parameter estimation. *IEEE Transactions on Automatic Control*, vol. 33(2), pp. 217-222.

Luo, J., Pattipati, K. R., Qiao, L., & Chigusa, S. (2008). Model-based prognostic techniques applied to a suspension system. *IEEE Transactions on System, Man and Cybernetics*, vol. 38(5), pp. 1156-1168.

Mohanty, S., Teale, R., Chattopadhyay, A., Peralta, P., & Willhauck, C. (2007). Mixed Gaussian process and state-space approach for fatigue crack growth prediction. *International Workshop on Structural Heath Monitoring*, vol. 2, pp. 1108-1115.

Neal, R. M. (1995). *Bayesian learning for neural networks*. University of Toronto, Toronto, Ontario, Canada.

Nelson, W. (1990). *Accelerated testing: Statistical models, test plans, and data analysis*. NY: John Wiley & Sons.

Newman, J. C. Jr, Phillips, E. P., & Swain, M. H. (1999). Fatigue-life prediction methodology using small-crack theory. *International Journal of Fatigue*, vol. 21, pp. 109-119.

Orchard, M. E., & Vachtsevanos, G. J. (2007). A particle filtering approach for on-line failure prognosis in a planetary carrier plate. *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 7(4), pp. 221-7.

Paris, P. C., & Erdogan, F. (1963). A critical analysis of crack propagation laws. *ASME Journal of Basic Engineering*, vol. 85, pp. 528–534.

Park, J. I., & Bae, S. J. (2010). Direct prediction methods on lifetime distribution of organic light-emitting diodes from accelerated degradation tests. *IEEE Transactions on Reliability*, vol. 59(1), pp. 74-90.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, vol. 77(2), pp. 257-286.

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009). On applying the prognostic performance metrics. *Annual Conference of the Prognostics and Health Management Society*. September 27 - October 1, San Diego, CA.

Schwabacher, M. A. (2005). A survey of data-driven prognostics. *AIAA Infotech@Aerospace Conference*, September 26-29, Arlington, VA.

Seeger, M. (2004). Gaussian processes for machine learning. *International Journal of Neural Systems*, vol. 14(2), pp. 69-106.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, vol. 1, pp. 211-244.

Tran, V. T., & Yang, B. S. (2009). Data-driven approach to machine condition prognosis using least square regression tree. *Journal of Mechanical Science and Technology*, vol. 23, pp. 1468-1475.

Yan, J., & Lee, J. (2007). A hybrid method for on-line performance assessment and life prediction in drilling operations. *IEEE International Conference on Automation and Logistics*, August 18-21, Jinan, Shandong , China.

Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, vol. 87(9), pp. 1423-1447.

Zio, E., & Maio, F. D. (2010). A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliability Engineering and System Safety*, vol. 95, pp. 49-57.

Dawn An received the B.S. degree and M.S. degree of mechanical engineering from Korea Aerospace University in 2008 and 2010, respectively. She is now a joint Ph.D. student at Korea Aerospace University and the University of Florida. Her current research is focused on the Bayesian inference, correlated parameter identification and the methodology for prognostics and health management and structural health monitoring.

Joo-Ho Choi received the B.S. degree of mechanical engineering from Hanyang University in 1981, the M.S. degree and Ph.D. degree of mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1983 and 1987, respectively. During the year 1988, he worked as a Postdoctoral Fellow at the University of Iowa. He joined the School of Aerospace and Mechanical Engineering at Korea Aerospace University, Korea, in 1997 and is now Professor. His current research is focused on the reliability analysis, design for life-time reliability, and prognostics and health management.

Nam Ho Kim received the B.S. degree of mechanical engineering from Seoul National University in 1989, the M.S. degree and Ph.D. degree of mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1991 and the University of Iowa in 1999, respectively. He worked as a Postdoctoral Associate at the University of Iowa from 1999 to 2001. He joined the Dept. of Mechanical & Aerospace Engineering at the University of Florida, in 2002 and is now Associate Professor. His current research is focused on design under uncertainty, design optimization of automotive NVH problem, shape DSA of transient dynamics (implicit/explicit) and structural health monitoring.

Dynamic Bayesian Networks for Prognosis

Gregory Bartram[1], Sankaran Mahadevan[2]

[1,2]*Vanderbilt University, Nashville, TN, 37212, USA*

gregory.w.bartam@vanderbilt.edu
sankaran.mahadevan@vanderbilt.edu

ABSTRACT

In this paper, a methodology for probabilistic prognosis of a system using a dynamic Bayesian network (DBN) is proposed. Dynamic Bayesian networks are suitable for probabilistic prognosis because of their ability to integrate information in a variety of formats from various sources and give a probabilistic representation of a system. Further, DBNs provide a platform naturally suited for seamless integration of diagnosis, uncertainty quantification, and prediction. In the proposed methodology, a DBN is used for online diagnosis via particle filtering, providing a current estimate of the joint distribution over the system variables. From this state estimate, future states of the system are predicted using the DBN and sequential Monte Carlo sampling. Prediction in this manner provides the necessary information to estimate the distribution of remaining use life (RUL). The DBN-based recursive prediction procedure may be used to estimate the system state between available measurements, when filtering is not possible. The prognosis procedure, which is system specific, is validated using a suite of offline hierarchical metrics. The prognosis methodology is demonstrated on a hydraulic actuator subject to a progressive seal wear that results in internal leakage between the chambers of the actuator.

1. INTRODUCTION

1.1. Background

The rise of complex and costly systems for use in extreme environments has resulted in new challenges in maintenance, planning, decision-making and monitoring for these systems. To reliably execute the missions they were designed for, these systems must be meticulously maintained. Traditional schedule-based maintenance results in unnecessary system downtime and the potential for serious problems to develop between routine maintenance. The alternative, condition-based maintenance (CBM) (Jardine, Lin, & Banjevic, 2006), monitors systems as they operate, alerting personnel when faults occur. Maintenance is performed on-demand, resulting in less downtime and lower costs. Further, online system measurements may occur on different time scales from one another or only be available in particular system configurations. This necessitates seamless integration of current state estimation and predictive techniques, which are part of a prognosis methodology.

Prognosis is the process of predicting the future state of a system coupled with information about the implications of that estimate of the system health state. The quantitative prognosis of a system is commonly expressed through the remaining useful life (RUL). RUL quantifies the amount of time until a system reaches some failure criterion, e.g. fault magnitude or performance metric crosses a threshold or system is no longer operable. Ideally, the uncertainty in RUL is quantified by estimating the distribution of RUL, resulting in a probabilistic prognosis. Importantly, probabilistic prognosis assesses the outlook for a specific instantiation of a system, or a particular unit under test (UUT). Measurement data updates the belief about the present state and RUL of the particular UUT. In this way, probabilistic prognosis differs from probabilistic reliability analysis, which aggregates data to obtain probabilistic reliability data for a population as opposed to an individual. Such population statistics may be suitable for tasks such as system design, but less helpful for operational and maintenance decisions that focus on individual units, as is the case in CBM.

A prognosis methodology should thus have several important characteristics. It should provide a distribution of RUL as opposed to a point estimate, thus accounting for the uncertainty coming from many sources (variability, information uncertainty, and model uncertainty). It should track the health of an individual unit. It should allow easy transition between situations when measurements are available and when they are unavailable. Finally, the methodology should survive rigorous validation.

Gregory Bartram et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Prognosis methodologies may be divided into statistical, data-based, model-based, and hybrid approaches (see e.g. (Jardine et al., 2006; Tran & Yang, 2009)). Statistical approaches include statistical process control (Goode, Moore, & Roylance, 2000), logistic regression (Yan, Koç, & Lee, 2004), survival models (Banjevic & Jardine, 2006; Vlok, Wnek, & Zygmunt, 2004), and stochastic process models (Lin & Makis, 2004; W. Wang, Scarf, & Smith, 2000; Wenbin Wang, 2002).

Data-based approaches consist of machine learning methods (support vector machines (Farrar & Worden, 2012), relevant vector machines (Tipping, 2001), neural networks (Dong & Yang, 2008; Farrar & Worden, 2012; Vachtsevanos & Wang, 2001; W. Q. Wang, Golnaraghi, & Ismail, 2004; Yam, Tse, Li, & Tu, 2001; Zhang & Ganesan, 1997)) and graphical models such as dynamic Bayesian networks (DBNs) hidden Markov models (HMMs) (Chinnam & Baruah, 2003; Kwan, Zhang, Xu, & Haynes, 2003). Liu et al. (2010) use adaptive recurrent neural networks for the estimation of battery RUL. Goebel et al. (2008) compare relevance vector machines (RVMs), Gaussian process regression (GPR) and neural network (NN) methods for prognosis. Gebraeel & Lawley (2008) use NNs for degradation modeling and test the methodology on ball bearings. Saha et al. (2009) compare relevance vector machines (RVMs, a Bayesian implementation of support vector machines) and particle filtering to estimate RUL distributions for batteries.

In model-based approaches, system models are used to estimate RUL or other relevant metrics. Such methods rely on accurate physics-based models for prediction. These include physical failure models (Kacprzynski, Sarlashkar, Roemer, Hess, & Hardman, 2004), filtering models (Orchard & Vachtsevanos, 2009, Lorton, Fouladirad, & Grall, 2013, B. Saha, Celaya, Wysocki, & Goebel, 2009, Khan, Udpa, & Udpa, 2011), and statistical models. Orchard and Vachtsevanos (Orchard & Vachtsevanos, 2009) use state estimation models combined with particle filtering for diagnosis and estimation of the RUL distribution of a planetary gear. Lorton et al. (Lorton et al., 2013) combine the differential equations of a system with system measurements via particle filtering for probabilistic model-based prognosis.

Hybrid methodologies combine multiple approaches, i.e., a combination of data-driven and model-based approaches. E.g. Kozlowski (2003) uses ARMA (autoregressive moving average) models (Box, Jenkins, & Reinsel, 2008), neural networks, and fuzzy logic for estimation of the state of health, state of charge, and state of life of batteries.

DBNs are probabilistic graphical models with diagnostic and prognostic capabilities. They have shown promise in several recent applications. Dong and Yang (2008) use DBNs combined with particle filtering to estimate the RUL distribution of drill bits in a vertical drilling machine. While very useful, particle filtering is not the only inference method available for prognosis. Jinlin and Zhengdao (2012) use DBNs modeling discrete variables and the Boyen-Koller algorithm for prognosis. Tobon-Mejia et al. (2012) use mixtures of Gaussian HMMs (a form of DBN) to estimate the RUL distributions for bearings. The junction tree algorithm is used for exact inference. The prognosis methodology is validated using the hierarchical metrics proposed by Saxena et al. (2010).

1.2. Motivation

While the preceding literature review represents a number of prognosis approaches, prognosis is still an emerging research area with room for much additional work. One promising approach that has received relatively little attention is based on DBNs. DBNs have many qualities that are attractive for prognosis.

1) The graphical representation of a problem provided by DBNs aids understanding of interactions in a system.

2) DBNs provide a probabilistic model of the system that accounts for uncertainty due to natural variability, measurement error, and modeling error.

3) DBNs can integrate many types of information that may be encountered during prognosis (including expert opinion, reliability data, mathematical models, operational data, and laboratory data) into a unified system model.

4) DBNs can update the distributions of all variables in the network when observations are obtained for any one or more variables. This allows the most recent system measurements to be accounted for in prognosis.

Additionally, many prognosis methodologies are application-specific. There is still a need for prognosis methodologies that can be applied to a wide range of problems.

1.3. Contributions

In this paper, a framework for probabilistic prognosis is proposed. The methodology advances the use of DBNs in prognosis by building upon previous work in system modeling under heterogeneous information (Bartram & Mahadevan, 2013). Further, the DBN-based methodology addresses the need for a general prognosis framework for developing validated prognosis methodologies for any system.

The DBN is constructed from prior information, including physics of failure models — a key function of a prognosis methodology. The DBN is a store of prior information, but also provides the means for integrating current measurements into a probabilistic estimate of the current state of a system. Particle-filter based inference is used for diagnosis, and forward sampling in the DBN is used for recursive prediction. The particle-based probabilistic state estimate of the system that results from particle filtering is ideal for diagnosis uncertainty quantification, and provides a seamless transition from diagnosis to future state prediction using sequential Monte Carlo sampling. The ability of the methodology to estimate RUL is validated using metrics from Saxena et al. (2010). In the second, online state estimation is desired, but measurements are not available or available periodically. The methodology is illustrated for a hydraulic actuator with a seal leak.

The remainder of this paper is organized as follows. Section 2 details the proposed prognosis methodology, including system modeling, diagnosis, prediction, and validation. In Section 3, the proposed methodology is demonstrated on a hydraulic actuator system with a progressive internal leak. Section 4 discusses conclusions and future work.

2. Proposed Prognosis Framework

The challenge of prognosis is to minimize the uncertainty in the estimated distribution of RUL given constraints on available information about the system, operating environment and loading conditions, computational resources, and time horizon. In this paper, a DBN-based prognosis framework is proposed. The prognosis framework first constructs a DBN-based system model using heterogeneous information sources. Expert opinion, reliability data, mathematical models, and operational and laboratory data are used in the construction of the DBN model. In particular, inclusion of physics of failure models is important in prognosis. The evolution of phenomena such as cracking, wear, and corrosion play a large role in determining the health of a system. The system model is used for diagnosis to obtain information about the current state of the system. A sequential Monte Carlo then predicts future system states and estimates the RUL distribution. Finally, the prognosis capability of the resulting system model, diagnostic, and predictive algorithms are validated using a four step hierarchical procedure. The prognosis procedure is shown in Fig. 1.

2.1. Dynamic Bayesian Networks

A dynamic Bayesian network is the temporal extension of a static BN. A static BN, also referred to as a belief network and directed acyclic graph (DAG), is a probabilistic graphical representation of a set of random variables and their conditional dependencies. Variables are represented by nodes (vertices) and conditional dependence is represented by directed edges. Unconnected nodes are conditionally independent of each other. The acyclic requirement means that no paths exist in the graph where, starting at node x_i, it is possible to return to node x_i.

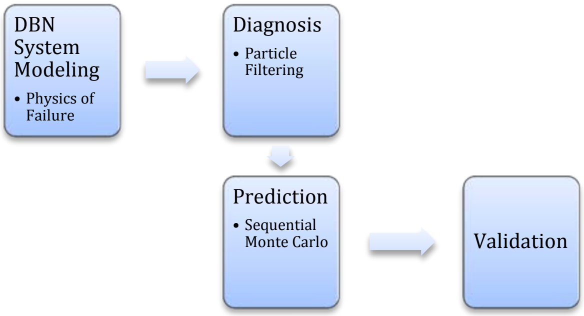


Figure 1. Proposed Prognosis Methodology

A DBN describes the joint distribution of a set of variables **x** on the interval [0, ∞). This is a complex distribution, but may be simplified by using the Markov assumption. The Markov assumption requires only the present state of the variables $\mathbf{x}^t$ to estimate $\mathbf{x}^{t+1}$, i.e. $p(\mathbf{x}^{t+1} \mid \mathbf{x}^0, \ldots, \mathbf{x}^t) = p(\mathbf{x}^{t+1} \mid \mathbf{x}^t)$ where p indicates a probability density function and bold letters indicate a vector quantity. Additionally, the process is assumed to be stationary, meaning that $p(\mathbf{x}^{t+1} \mid \mathbf{x}^t)$ is independent of t. This approach to modeling DBNs is developed by Friedman et al. (Friedman, Murphy, & Russell, 1998).

A DBN may be composed of all discrete variables, all continuous variables, or hybrid set of discrete and continuous variables. A conditional probability distribution (CPD) is chosen for each variable, e.g. Gaussian, tabular (multinomial), softmax, deterministic, logic, etc. See Koller and Friedman (2009) for a detailed explanation of CPDs. For modeling systems with faults, it is advantageous to consider a hybrid system, typically with the continuous variables being modeled as continuous and the faults being discrete. Theory for networks with Gaussian continuous variables is developed in Heckerman and Geiger (1995) and Lauritzen (1992).

DBNs provide a flexible modeling framework, allowing integration of expert opinion, reliability data, mathematical models (including system state space, surrogate, and physics of failure models), existing databases of operational and laboratory data, and online measurement information. Bartram and Mahadevan (2013) have proposed a methodology for integration of such heterogeneous information into DBN system models. In the next section, that discussion is extended to consider physics of failure models, which are of particular importance in prognosis.

2.2. Physics of Failure Models

A key distinction between a system model capable of diagnosis and one capable of prognosis is that a prognostic model can estimate the evolution of damage in the future while a diagnosis model only needs the ability to infer the current state of damage. Diagnostic procedures based on fault signatures or pattern recognition are examples of this. While they may be able to detect and isolate damage, quantification can be done using least-squares based estimation, they do not necessarily have any ability to model progressive damage mechanisms such as crack growth, wear, and corrosion. One of the challenges of prognosis is developing accurate and comprehensive physics of failure models. These damage mechanisms are complex, varying with system design and dynamics, and can interact in many ways.

For illustration, the example problem in this paper considers a dynamic seal in a hydraulic actuator. Seal failure is discussed in great detail in (Naval Surface Warfare Center, 2011). A dynamic seal prevents leakage when there is relative motion between two surfaces. The seal under consideration prevents leakage between the two chambers of the actuator. Modeling the failure of a seal can become complicated very quickly, as a number of factors influence seal failure, including, material characteristics, amount of seal compression, surface irregularities, seal size, fluid pressure, pressure pulses, temperature, fluid viscosity, fluid contamination, fluid/material compatibility, allowable leakage levels, and assembly and quality control procedures. The failure symptoms include excessive leakage and slow mechanical response. Many mechanisms and causes of these symptoms are described in (Naval Surface Warfare Center, 2011).

In this paper, the wear mechanism is considered for a seal in a hydraulic actuator. Generally, seal leakage is due to wear caused by friction between the seal and piston, which removes seal material and allows fluid to pass between the chambers of the actuator. There are multiple wear mechanisms including adhesive wear, abrasive wear, surface fatigue, fretting wear, and erosive wear (Jones, 1983). Lancaster (1969) explains many of the complexities of abrasive wear while Briscoe and Sinha (2002) and Briscoe (1981) review wear of polymers. Due to the complexity of the mechanisms of wear, wear is typically modeled through the use of an experimentally determined wear rate.

Nikas (2010) has written an extensive literature review on seal wear in actuators. The leakage area is the result of the removal of seal material — typically a polytetrafluoroethylene (PTFE) polymer — which is a function of load, distance traveled, material properties of the actuator and seal, geometry of the actuator, temperature, hydraulic fluid viscosity, and contaminants. Experimentally determined wear rates (mm^3/m/N) are available for PTFE composites used in hydraulic actuators e.g. Sawyer et al. (2003) and Khedkar et al. (2002).

The volume of material removed from the seal per cycle depends on the friction force and sliding distance per cycle and may be calculated by

$$V(t) = w_{seal}(t)F(t)d(t) \tag{1}$$

where w_{seal} is the wear rate of the seal in mm^3/N/m, F is the frictional force on the seal, and d is the total sliding distance, and *t* refers to the load cycle.

For the seal shown in Fig. 2, where L is the contact length of the seal and P is pressure, the leakage area (considered in Eqs. 16-29 as in (Thompson, Pruyn, & Shukla, 1999)) based on the volume of material removed is assumed to be $a_{leak} = V(t)/L$.

While wear is a continuous process, in this paper the occurrence of wear is modeled as a binary event, where modeling begins when the leakage area has reached a value that has detectable effects. The occurrence is modeled using an empirically derived seal failure rate, which modifies an experimentally determined base failure rate for the seal. Details of deriving the failure rate are available in (Naval Surface Warfare Center, 2011).

The wear rate itself varies with factors such as the age of the seal, temperature, contaminants in the fluid. The load experienced by the seal also varies as does the velocity of the actuator. As a result the volume of material removed and leakage area are nonlinear functions. However, for the sake

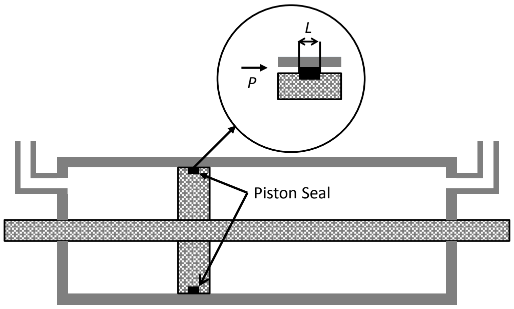


Figure 2. Hydraulic actuator diagram showing dynamic seals

of demonstration, it is assumed that the leakage area and volume of material removed vary linearly. This implies that the wear rate is steady, which is possible outside of the initial wear-in phase and under constant environmental conditions. Additionally, the load and velocity of the actuator are assumed to remain steady.

2.3. Diagnosis

Diagnosis is the process of detecting and isolating damage in a system and quantifying the magnitude of damage. When the probability of a fault occurring crosses the detection threshold, a fault isolation procedure finds fault set candidates to further analyze. To isolate candidate fault sets, statistical inference computes the probability of each fault set. The magnitude of the fault may then be estimated.

In the context of prognosis, diagnosis (or more generally, filtering) provides the initial conditions for prognosis of a UUT. The initial condition for prognosis has a large impact on the accuracy and precision of the RUL distribution. As such, it is important to understand and account for diagnosis uncertainty.

Uncertainty in diagnosis is due to natural variability, measurement error, model error, hypothesis testing error, error in inference, and any approximations in optimization or least squares procedures used for estimating fault magnitudes. Sankararaman and Mahadevan propose a methodology for quantifying the uncertainty in diagnosis. This is an integral part of the diagnosis procedure, and it is expanded in this paper to accommodate a particle filter (PF) based diagnosis procedure.

Detection and isolation can be performed using a DBN model of the system to estimate the state of the system as measurements, $\mathbf{z}^t$, become available. The simplest procedure is to "unroll" the two time slice network and compute the states of all the unobserved variables in the system, $\mathbf{x}^t$, including faults, using standard inference techniques such as the clique tree algorithm or Markov chain Monte Carlo (MCMC) (Koller & Friedman, 2009). However, exact inference is generally a computationally intractable problem (Boyen & Koller, 1998). As a result, approximate inference based on Bayesian recursive filtering is pursued.

2.3.1. Bayesian Recursive Filtering

The procedure for updating the belief about the system state as new information becomes available is called Bayesian recursive filtering. Bayes' theorem is the engine for performing the update. Diagnosis of a dynamic system may be achieved by maintaining the joint distribution over the system variables, parameters, and faults and as new noisy measurements become available via Bayesian recursive filtering. The joint distribution provides the best estimate of whether faults have occurred and what values system parameters and responses may have. This joint distribution is commonly called the belief state σ^t. $\sigma^t = p(\mathbf{x}^t|\mathbf{z}^{1:t})$, where $p(\mathbf{x}^t|\mathbf{z}^{1:t})$ is the distribution over the variables $\mathbf{x}^t$ estimate given all previous measurements $\mathbf{z}^{1:t}$. The belief state estimate includes estimates of the states of faults and system parameters, whose states are otherwise unknown. Equation (2), derived from Bayes' theorem (see Appendix 1), is the engine for belief state updating.

$$\sigma^{t+1}(\mathbf{x}^{t+1}) = \frac{p(\mathbf{z}^{t+1}|\mathbf{x}^{t+1})p(\mathbf{x}^{t+1}|\mathbf{z}^{1:t})}{p(\mathbf{z}^{t+1}|\mathbf{z}^{1:t})} \tag{2}$$

$p(\mathbf{z}^{t+1}|\mathbf{x}^{t+1})$ is the likelihood of the measurements, $p(\mathbf{x}^{t+1}|\mathbf{z}^{1:t})$ is the prior state estimate at time *t*, $p(\mathbf{z}^{t+1}|\mathbf{z}^{1:t})$ is a normalizing constant, and $\sigma^{t+1}(\mathbf{x}^{t+1})$ is the posterior state estimate at time *t*.

Complete tutorials on Bayesian recursive filtering are available in Koller and Friedman (2009) and Ristic and Arulampalam (2004).

2.3.2. Particle Filtering

Under certain assumptions, such as when the system is linear Gaussian, the belief state $\sigma^{t+1}(\mathbf{x}^{t+1})$ will be of a known parametric form and computationally efficient solutions to the filtering problem (e.g. Kalman filter, extended Kalman filter, unscented Kalman filter) are available. Outside such assumptions, a computationally feasible method for inference in the DBN is found in particle filtering, a form of sequential Monte Carlo based on Bayesian recursive filtering (see e.g. Chen (2003)).

Particle filtering is a method for approximating the distribution of the belief state with a set of samples and weights. Common particle filtering method are based on sequential importance sampling (SIS), which improves upon the basic sequential MC by weighting point masses (particles) according to their importance sampling density, thus focusing on the samples that affect the posterior belief state the most. A comprehensive tutorial on particle filters is given by Ristic et al. (2004) and in Koller and Friedman (2009).

A summary of the SIS algorithm for one time step is as follows. A previous (or initial if t = 1) set of N_s weights w_t^i and N_s corresponding particles $\mathbf{x}_t^i$ are given initially or known from the previous time step, where *i* denotes the *i*th particle. These particles represent an approximation of the belief state by

$$\begin{aligned}\sigma^{t+1}(\mathbf{x}^{t+1}) &= p(\mathbf{x}^{t+1}|\mathbf{z}^{1:t+1}) \\ &\approx \sum_{i=1}^{Ns} \delta(\mathbf{x}-\mathbf{x}_i^{t+1})w_i^{t+1}\end{aligned} \tag{3}$$

N_s samples are drawn from the importance distribution, $q(\mathbf{x}_i^{t+1}|\mathbf{x}_i^t, \mathbf{z}^{t+1})$, where $\mathbf{z}^{t+1}$ are the measurements at the *t+1*th time step. In a DBN, sampling is performed in the two-slice template after computing $\mathbf{x}^t$ and $\mathbf{z}^{t+1}$. The values for the remaining $\mathbf{x}^{t+1}$ are then sequentially sampled in topological order (parent then child).

A weight w_i^{t+1} is then computed for each particle $\mathbf{x}_i^{t+1}$ up to a normalizing constant based on the ratio of the belief state to the importance density,

$$w_i^{t+1} \propto \frac{\sigma(\mathbf{x}_i^{1:t+1}|\mathbf{z}^{1:t+1})}{q(\mathbf{x}_i^{1:t+1}|\mathbf{z}^{1:t+1})} \tag{4}$$

using the weight update equation in Eq. (5), which is derived from the ratio of the pdf of the belief state to the pdf of the importance sampling density.

$$w_i^{t+1} \propto \frac{p(\mathbf{z}^{t+1}|\mathbf{x}_i^{t+1})p(\mathbf{x}_i^{t+1}|\mathbf{x}_i^t)}{q(\mathbf{x}_i^{t+1}|\mathbf{x}_i^t, \mathbf{z}^{t+1})} \tag{5}$$

The weights $\mathbf{w}^{t+1}$ are normalized so their sum is equal to 1. The normalized weights and points $\{\mathbf{x}_i^{t+1}\}_{i=1}^{N_2}$ form an approximation to the belief state estimate in Eq. (3).

The basic SIS algorithm suffers from the degeneracy phenomenon, wherein all but a few of the particles have negligible weight after only a few updates. This tends to waste computational effort on particles with practically zero probability. Two techniques to reduce this phenomenon are choosing an optimal importance density $q(\mathbf{x}_i^{t+1}|\mathbf{x}_i^t, \mathbf{z}^{t+1})$ and resampling. The optimal importance density may only be determined analytically when the system variables are discrete with a finite number of possible values or when the system variables are Gaussian. In other cases, suboptimal approximations based on local linearization (Doucet, Godsill, & Andrieu, 2000) or Gaussian approximations using the unscented transform (West & Harrison, 1997) may be used. Often, for convenience the importance density $q(\mathbf{x}^{t+1}|\mathbf{x}_i^t, \mathbf{z}^{t+1})$ is taken as the prior $p(\mathbf{x}_i^{t+1}|\mathbf{x}_i^t)$ or the likelihood $p(\mathbf{z}^{t+1}|\mathbf{x}_i^{t+1})$. If the prior is used, $q(\mathbf{x}^{t+1}|\mathbf{x}_i^t, \mathbf{z}^{t+1}) = p(\mathbf{x}^{t+1}|\mathbf{x}_i^t)$, and the weight update in Eq. (5) simplifies to

$$w_i^{t+1} \propto w_i^t p(\mathbf{z}^{t+1}|\mathbf{x}_i^{t+1}) \tag{6}$$

Resampling focuses the particle filter on the particles with the largest weights. An empirical CDF is constructed based on the weights $\mathbf{w}_{t.}$ Particles are sampled (with replacement), replicating the particles with the largest weights. The result is N_s particles all with weight $1/N_s$. Resampling may be performed after every update or when a measure of degeneracy, the effective sample size, N_{eff}, falls below a threshold. N_{eff} may be taken as (Ristic & Arulampalam, 2004)

$$N_{eff} = \frac{N_s}{\sum_{i=1}^{N_s}(w_i^{t+1})^2} \tag{7}$$

Many variations of the SIS particle filter exist with different importance densities $q(\mathbf{x}^{t+1}|\mathbf{x}_i^t, \mathbf{z}^{t+1})$ and resampling procedures (Ristic & Arulampalam, 2004). In this paper, an algorithm for systems with multiple operating modes (Andrieu, Davy, & Doucet, 2003) that extends the auxiliary particle filter (Pitt & Shephard, 1999) is used.

2.4. Fault Diagnosis and Diagnosis Uncertainty Quantification

When using a particle filter, the belief state itself provides the information necessary for fault detection, isolation, and damage quantification. The marginal distribution over combinations of the discrete fault indicator variables is a multinomial distribution, whose parameters are easily calculated from the particles representing the current belief state. Given m fault indicator variables that can take on values of true or false, there are $n = 2^m$ combinations of faults, including the healthy condition.The i^{th} combination at the t^{th} cycle has an expected probability $p_i^t = \sum N_i^t w_i^t / N_s$, where N_s is the number of samples used in particle filtering, N_i^t is the number of occurrences of the i^{th} fault combination, and w_t^i are the normalized weights for those particles.

The probability of *any* fault (detection probability) is then $p_F^t = 1 - p_0^t$, where p_0^t is the probability of the fault combination where no faults occur. When p_0^t is greater than some threshold, an alert may be issued to a decision maker and a prognosis procedure may be triggered. The remaining p_i^t $(i \neq 0)$ are the isolation probabilities of each fault combination. From the belief state, $\sigma^{t+1}(\mathbf{x}^{t+1})$, the marginal distributions over damage parameters may be constructed from the particles and their weights.

The probabilities $\mathbf{p}_t$ that parameterize a multinomial distribution are themselves uncertain and follow a Dirichlet distribution. Based on the Dirichlet distribution, the variance of p_k^i is

$$Var[p_t^i] = \frac{N_k^i(N_s - N_k^i)}{{N_s}^2(N_s + 1)} \tag{8}$$

The uncertainty in $\mathbf{p}_t$ is directly dependent on the number of samples, N_s. With the detection and isolation probabilities and their corresponding uncertainties as well as estimates of the distributions of damage parameters known, a decision maker is better able to access the criticality of damage and the appropriate actions to make to balance safety and cost concerns.

2.4.1. Prediction

In probabilistic prognosis, possible future states of the system are generated and the remaining useful life (RUL) distribution, $r(t)$, of the particular unit under test (UUT) is estimated. RUL is the amount of time a UUT is usable until corrective action is required and may be measured in hours, minutes, cycles, etc. Measurements are unavailable and the system model is assumed to be valid under future operating conditions. Prediction can be initiated at any time in the life of a system based on the last available state estimate. However, in this paper, the time of prognosis, t_P, the first time point for which a prognosis estimate is obtained, is after the time of fault detection, t_D. Figure 3 illustrates these important prognosis time indices.

One approach to prediction when performing particle filtering on a DBN is a basic sequential Monte Carlo. Starting with the last belief state estimate (with measurements available), particles are recursively sampled through the two time slice DBN until some termination criteria is met, such as $Pr(r(t) = 0)$ is above some target threshold. Thus, there are N_s trajectories of the variables of interest beginning at time t, $\{\mathbf{\Phi}(t)\}_{i=1}^{N_s}$. Each trajectory consists of a series of predictions for the variables of interest, $\mathbf{\Phi}(t) = \{\boldsymbol{\varphi}(t|t), \boldsymbol{\varphi}(t+1|t), \dots \boldsymbol{\varphi}(EoP|t)\}$, where the end of prediction (EoP) is the time index of the last prediction before the end of life (EoL) is reached. Particle weights are fixed from the last available measurement, as there is no basis for updating the weights (Eq. 5). This results in a particle-based approximation of RUL (similar to the belief state approximation), using the last available set of weights. When a new measurement is obtained, a new RUL distribution is estimated.

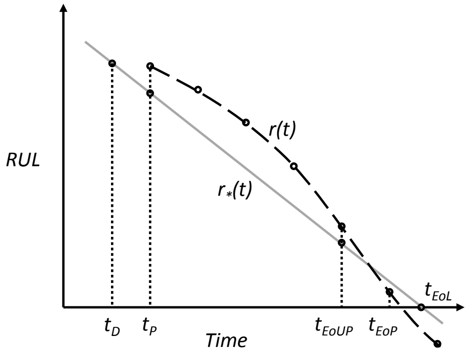


Figure 3. Prognosis time indices: $r_*(t)$ is the ground truth RUL, t_{EoUP} is the end of useful prognosis, dashed line depicts mean $r(t)$.

To further tailor the prognosis to a particular UUT, the conditional probability distributions in the DBN may be updated as measurements become available. This may be performed via Bayesian updating of the distributions. If a conjugate prior is available, the update can be performed analytically. Otherwise, techniques such as Markov chain Monte Carlo (MCMC) may be required.

The RUL distribution is sensitive to many aspects of the problem. The initial state estimate provided by the diagnosis algorithm must be accurate. As such, the filtering algorithm and number of particles are important algorithmic design decisions. These decisions also involve a tradeoff between accuracy and computational effort, which must be considered. Optimal sensor placement and improved sensor reliability also impact the accuracy of the diagnosis.

The accuracy of predictive models, including those for inputs (loads) and physics of failure models, is another large source of uncertainty in the RUL estimate. Because the prediction is recursive with no measurements available to correct the prediction, errors in prediction compound quickly and must be minimized.

2.4.2. Measurement Gaps

Systems may experience periods of times where measurements are unavailable. This may be a result of the system configuration, availability of measurements, failure of sensing systems, or the desire to have system state estimates at a higher frequency than the available measurements. For example, offline inspection data may be available for an aircraft on the ground, while onboard sensing provides a steady stream of information about temperature, altitude, windspeed, pressure, etc. These onboard measurements may only be available for portions of a flight (perhaps during cruising but not takeoff or landing).

Using the same recursive sampling used for RUL estimate, predictions may be produced and used to fill in the information gaps. When a measurement becomes available, the particle filtering algorithm is used to update the last predicted system state. The particle filter update may be performed as long as at least one measurement is available. The process is shown in Fig. 4.

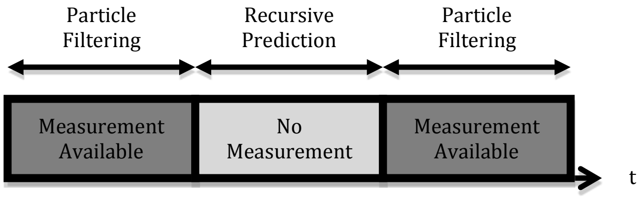


Figure 4. Handling measurement gaps

2.5. Prognosis Validation

Prognosis validation is essential to establish confidence in the RUL estimate. Many sources of uncertainty, including modeling errors, sensor faults, data noise, and unpredictable loading conditions and operating environments, strongly affect prognosis. Therefore, validation of a prognosis procedure must be done using strong performance metrics. These metrics must be carefully chosen, as many issues arise when evaluating prognosis algorithms, such as time scales or the ability to improve accuracy as more measurements are obtained (Saxena et al., 2010). Saxena et al. (2010) propose a standard offline four metric hierarchical test to evaluate a prognosis algorithm. This hierarchical test assumes that prognosis improves as more measurements become available. Combined, these four metrics provide a means for testing and comparing prognostic algorithms.

The first two metrics examine the accuracy of the RUL estimates by determining the probability p that the RUL estimate is between $\pm\alpha$ of the ground truth RUL. This probability p is compared to a threshold value, β. It is desirable for p to be greater than β. The primary difference between the first two metrics is in how α is determined, which results in a stricter test for the second metric than the first.

In the first metric, prognostic horizon (PH) is considered. Prognostic horizon indicates the time at which RUL estimates using a particular prognostic algorithm for a particular system are within acceptable limits. The upper and lower limits are the ground truth RUL plus or minus a constant α, which is some percentage of the EoL value. PH is the difference between the true EoL time and the time when the prognostic algorithm attains this acceptable level of accuracy ($p > \beta$). As this is a validation metric, the true EoL is known. A longer PH is indicative of a better prognostic algorithm. Figure 5a provides a visual representation of prognostic horizon.

Prognostic horizon maintains upper and lower bounds that are always the same distance from the true RUL. The second validation metric, $\alpha - \lambda$ accuracy, utilizes a stricter criterion that gradually tightens the limits about the RUL estimate (Fig. 5b). Additionally, the accuracy of the RUL is considered at time t_λ, where $0 \leq \lambda \leq 1$, $t_\lambda = t_P + \lambda(t_{EoL} - t_P)$, and t_P is the time at which a prognosis estimate is first obtained. This metric reflects the idea that, as more information is collected about the system, the RUL estimate is expected to improve, and thus the accuracy requirement for the RUL estimate should become more stringent. The $\alpha - \lambda$ accuracy is equal to 1 when the increasingly stringent accuracy requirements are met, and zero otherwise.

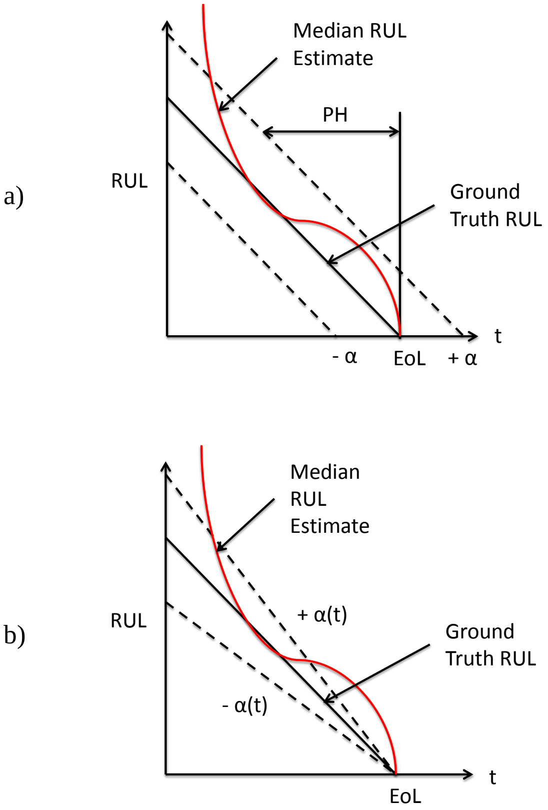


Figure 5. a) Prognostic horizon with +/- α bounds about the ground truth RUL
b) +/- α bounds for evaluating α-λ accuracy

In step three, the relative accuracy (RA) of the prognostic algorithm is calculated. Instead of merely indicating that accuracy requirements have been met, the accuracy of the RUL estimates are quantified. At t_λ

$$RA_\lambda = 1 - \frac{|r_*(t_\lambda) - r(t_\lambda)|}{r_*(t_\lambda)} \qquad (9)$$

where $r(t_\lambda)$ is a central tendency point such as the mean or median of the RUL estimate at t_λ and $r_*(t_\lambda)$ is the ground truth RUL. The RA is computed separately for each time step at which RUL is estimated. RA is a value between 0 and 1, and values closer to 1 indicate better accuracy.

Finally, if the prognostic algorithm satisfies all the previous metrics, a final metric to compute is convergence. Convergence is a measure of how quickly a metric, such as RA, improves with time. A high rate of convergence is desirable and leads to a larger PH. To estimate convergence of a prognosis algorithm based on some metric M,

$$C_M = [(x_c - t_P)^2 + {y_c}^2]^{1/2} \tag{10}$$

where

$$x_c = \frac{\frac{1}{2}\sum_{i=P}^{EoUP}({t_{i+1}}^2 - {t_i}^2)M(t_i)}{\sum_{i=P}^{EoUP}(t_{i+1} - t_i)M(t_i)} \tag{11}$$

and

$$y_c = \frac{\frac{1}{2}\sum_{i=P}^{EoUP}({t_{i+1}}^2 - {t_i}^2)M(t_i)^2}{\sum_{i=P}^{EoUP}(t_{i+1} - t_i)M(t_i)} \tag{12}$$

$M(\mathrm{t_i})$ is the non-negative prediction accuracy, EoUP is the end of useful prediction, and P is the time at which the prognosis algorithm makes its first prediction. End of useful prediction is the time after which corrective action cannot be performed before EoL. A high rate of convergence is better and leads to a larger PH.

2.6. Summary of Prognosis Framework

This section presented a framework for probabilistic prognosis. DBNs are used as a system modeling paradigm due to their ability to handle uncertainty and to integrate many types of information, both in the offline model construction phase and the online belief state updating phase. For prognosis, it is of particular importance to model complex physics of failure phenomena and integrate such models into the DBN. After the DBN model is established, the model is used for tracking the state of a particular UUT. Particle filtering is used to update the belief state as new measurements are obtained. Uncertainty in the state estimate (diagnosis) is quantified, and when a fault is detected, estimation of RUL via recursive prediction begins. The result is an estimate of the distribution of RUL. Section 2.4 considers the situation when there are gaps in the availability of measurements.

When a prognosis procedure (DBN model of system combined with available measurements and filtering algorithm), is designed for a particular system, it is then validated using the 4 step hierarchical procedure outlined in Section 2.5.

3. Illustrative Example

A hydraulic actuator system was considered to demonstrate the proposed methodology. Such a system is often used to manipulate the control surfaces of aircraft. The system consists primarily of three subsystems: a hydraulic actuator, critical center spool valve, and an axial piston pump (Fig. 6). The pump moves hydraulic fluid through the servovalve and into the actuator. The servovalve controls the flow of hydraulic fluid into the actuator, thus modulating the position of the actuator. Expert opinion, reliability data, mathematical models, operational data, and laboratory data were used to construct a DBN model of the spool valve and hydraulic subsystems.

First, expert opinion is invoked to determine the scope of the problem, variables and faults to model, and establish the DBN structure. Next, reliability data is drawn upon to determine the conditional probabilities for the faults. The mathematical model of the system is used to generate predictions of the system variables. The predictions are treated similar to operational and laboratory data and used to train a regression model for estimating the reduction in seal orifice area, which is equivalent to the seal leakage area. Considering the actuator cross section in Fig. 7, the surface area of the seal is $(r_2^2 - r_1^2)$, where r_1 and r_2 are the inner and outer radii of the seal, respectively.

3.1. DBN Model Construction

3.1.1. Expert Opinion

Expert opinion was considered first to define the basic parameters of the problem. A DBN representation of the system was chosen because heterogeneous information

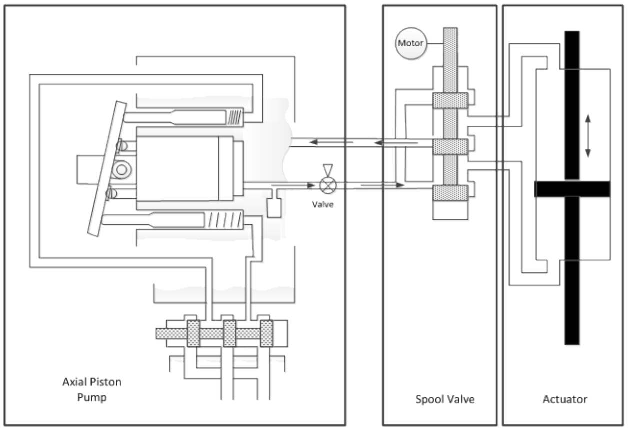


Figure 6. Hydraulic actuator system

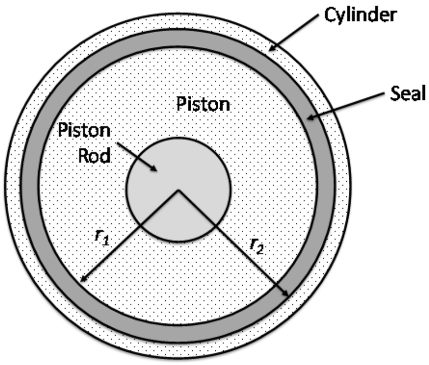


Figure 7. Actuator cross section

sources were available, the intended use of the model is diagnosis and prognosis, and the system is dynamic. Seven state variables and six discrete faults were selected to model the behavior of the system.

A generic initial structure for the DBN is first selected (Fig. 8) based on expert opinion. This generic two time slice structure consists of the set of faults, ***F***, model parameters, ***θ***, system state, ***y***, and measurements, ***z***. In this structure, faults cause changes in system parameters, which then cause changes in system responses, which are observed. ***F*** contains the persistent variables in the DBN – their future values depend upon their present values. The observations, ***z*,** while not connected across time slices, are nonetheless *not* independent across time slices, but correlated via ***θ***.

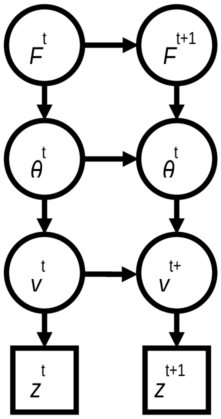


Figure 8. Generic DBN structure.

Table 1. List of faults and affected parameters.

Fault	Parameter Affected
Seal Leak	Leakage Area
Water Leak into System	Hydraulic Fluid Bulk Modulus
Air Leak Into System	Hydraulic Fluid Bulk Modulus
Pressure Valve Malfunction	Supply Pressure
Pump Pressure Sensor Fault	Supply Pressure
Electrical Fault	Control Signal

Table 1 lists the faults considered in the actuator system and the parameter affected by that fault (the faults are described further in Section 3.1). For each fault, a binary variable is added to the network at time *t* and *t* + 1. Similarly, a Gaussian variable is added at time *t* and *t* + 1 for each affected parameter. Links are drawn pointing from faults to affected parameters. The parameters are assumed to have Gaussian distributions, whose mean and variance depend on the health state of the system. The leakage area parameter is a special case, as it is zero when no leakage exists. Upon instantiation of a leak, its value is assumed to follow a Gaussian distribution. Thereafter, the leak is assumed to grow according to a polynomial regression model (Section 3.1, Mathematical Models), which is constructed using laboratory data.

Parameters from the current time step and initial conditions from the previous time step are input into a physics-based model of the actuator, which estimates the system responses, assumed to be Gaussian variables. Measurements are then connected to the corresponding system response. Links are also drawn between like faults at time *t* and *t* + 1 and like parameters at time *t* and *t* + 1. Finally, a Gaussian variable is added at time *t* and *t* + 1 for each measurement available. The resultant DBN is shown in Fig. 9 with parameters described in Table 2.

3.1.2. Published Reliability Data

The DBN model of the system should be able to simulate multiple faults and extrapolate system behavior multiple steps into the future for the model to be a useful diagnosis and prognosis tool. The overall failure rate for an actuator may be determined by estimating the base failure rate and making empirical corrections for temperature and fluid contamination (Naval Surface Warfare Center, 2011). The

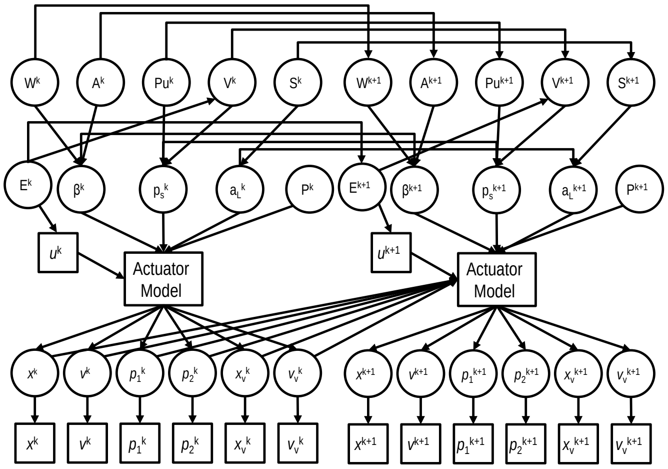


Figure 9. DBN structure as a result of expert opinion

Table 2. DBN variables.

State Variable	Symbol	Unit	Type	Note
Actuator position	x_{act}	m	continuous	measured
Actuator velocity	v_{act}	m/s	continuous	measured
Pressure in chamber 1	P_1	Pa	continuous	measured
Pressure in chamber 2	P_2	Pa	continuous	measured
Valve position	x_{valve}	m	continuous	measured
Valve velocity	v_{valve}	m/s	continuous	measured
Control signal	u	V	continuous	input
Water leak	W	-	binary	inferred
Air leak	A	-	binary	inferred
Pump sensor fault	P	-	binary	inferred
Valve fault	V	-	binary	inferred
Seal leak	S	-	binary	inferred
Fluid bulk modulus	β	MPa	continuous	inferred
Supply pressure	p_s	MPa	continuous	inferred
Leakage area	a_{leak}	mm^2	continuous	inferred

RIAC Databook (2006) and the Handbook of Reliability Prediction Procedures for Mechanical Equipment (Naval Surface Warfare Center, 2011) give failure rates for many mechanical systems. For illustration of the methodology, a handful of the possible faults for the actuator system are considered in this paper. Table 3 lists the faults that have been considered, the subsystem where they are located, and the information source for that fault.

The failure rates were then used to calculate the probability of each fault occurring. These probabilities correspond to parameters of the discrete fault indicator variables in the DBN. See Bartram and Mahadevan (Bartram & Mahadevan, 2013) for details.

3.1.3. Mathematical Behavior Models

Several mathematical models are used in this example. A physics-based model of a hydraulic actuator, described by Kulakowski et al. (2007) and Thompson et al. (1999) (see Appendix), is integrated into the DBN as a deterministic conditional probability distribution (Koller & Friedman, 2009). This model has been implemented in the Matlab Simulink environment.

Table 3. Faults Considered

Fault	Subsystem	Information Source
Seal Leak	Actuator	RIAC Databook (*RIAC Automated Databook*, 2006), Literature (Sepeheri, Karpenko, An, & Karam, 2005)
Water Leak into System	Piping/Fittings	RIAC Databook (*RIAC Automated Databook*, 2006), Literature (Sepeheri et al., 2005)
Air Leak Into System	Piping/Fittings	RIAC Databook (*RIAC Automated Databook*, 2006), Literature (Sepeheri et al., 2005)
Pressure Valve Malfunction	Pressure Valve	RIAC Databook (*RIAC Automated Databook*, 2006), Literature (Sepeheri et al., 2005)
Pressure Sensor	Piston Pump	Mathematical Model (Zeiger & Akers, 1986), Literature (Zeliang, 2005)
Electrical Fault	Electrical	RIAC Databook (*RIAC Automated Databook*, 2006)

For demonstration of the prognosis methodology, the load is synthesized using an ARIMA (autoregressive integrated moving average) model, which is treated as a deterministic conditional probability distribution in the DBN. In reality the load on a flight control actuator is depends on many variables related to the dynamics of the aircraft and the desired flight path (for e.g. see Mahulkar et al. (2010), Karpenko and Sepeheri (2003), and McCormick (1995)).

Finally, the physics of failure model for the seal leak is considered. The seal leakage area is modeled as in Section 2.2. The leakage area is modeled from laboratory data using a polynomial regression of the form $a_{leak}^{t} = c_1 + c_2(a_{leak}^{t+1})^2$.

3.1.4. Operational and Laboratory Data

Operational and laboratory data appear as historical databases and online measurement data. Laboratory data are used to train the polynomial regression model to estimate wear rate. Online measurement data (of the load *P*) are used to estimate the parameters of the ARIMA model used in load estimation.

3.2. Diagnosis

The actuator was operated for 20 seconds with a leak occurring after 6 seconds. At this point, the system has already reached the steady state. Measurements were obtained and updating performed at 0.5 second intervals. The system responses and load were assumed to be measurable while the system parameters including wear rate and leakage area were assumed to be unobservable. Inference via particle filtering (N_s = 250) was performed on the DBN to obtain filtered estimates of the system state.

After obtaining the state estimate at cycle *t*, the probability of detection was calculated as in Section 2.2. If the probability of detection exceeded 95%, an alarm was triggered. The fault was then isolated and quantified. Figure (10) shows maximum a posteriori (MAP) estimates of the system responses against their measured values. It is seen that the MAP system responses track the measured values closely. Figure 11 shows the MAP estimates of the system parameters, including the leakage area, and load against the ground truth values. This figure shows how the leakage area changes with time and how well the filtering procedure can infer the value of the leakage area. The system responses in Figure 11 are sensitive to changes in the supply pressure and leakage area, but insensitive to changes in the fluid bulk modulus. Changes in bulk modulus, however, may result in effects such as changes in wear rate that have not been included in the ground truth model. In both Fig. 10 and Fig. 11, the good estimates may be attributed to the use of an accurate physics-based model, but also to the use of synthetic measurement data, which may favorably bias the performance of filtering.

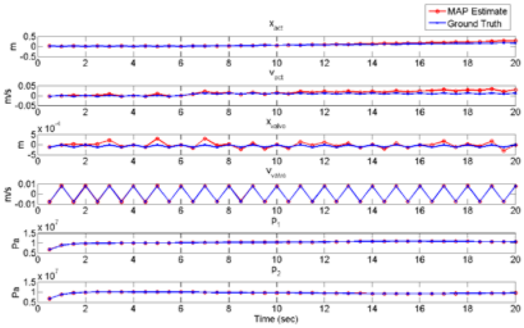


Figure 10. MAP estimates and measured values of actuator position and velocity, servovalve position and velocity, and pressure in each actuator chamber.

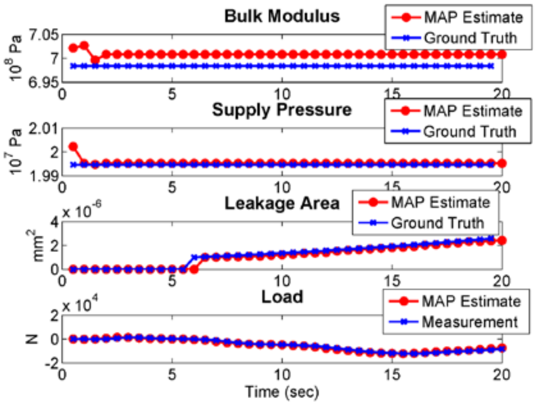


Figure 11. MAP estimate system parameters and load with ground truth and measured values.

3.3. Diagnosis Uncertainty

Diagnosis uncertainty was quantified after performing the diagnostic tasks of detection, isolation, and quantification. Figure 12 shows a kernel density estimate of the seal leak area from the particles at *t* = 6.5 seconds. Figure 13 shows the detection probability as it evolves with time. The detection probability passes the detection threshold soon after the fault occurs.

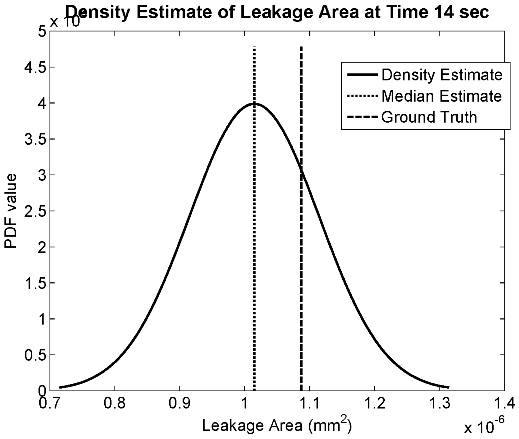


Figure 12. Kernel density estimate of leakage area estimate from particle filtering

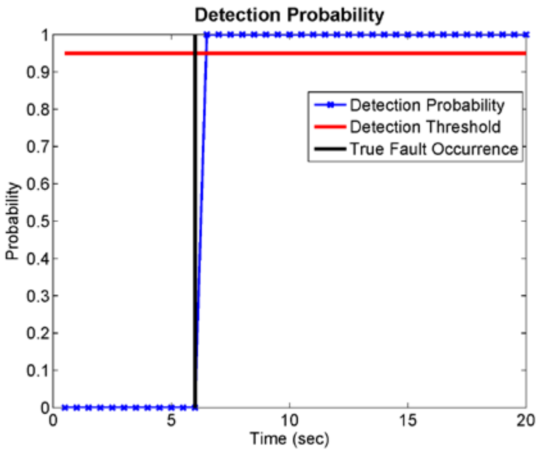


Figure 13. Fault detection probability with actual fault time

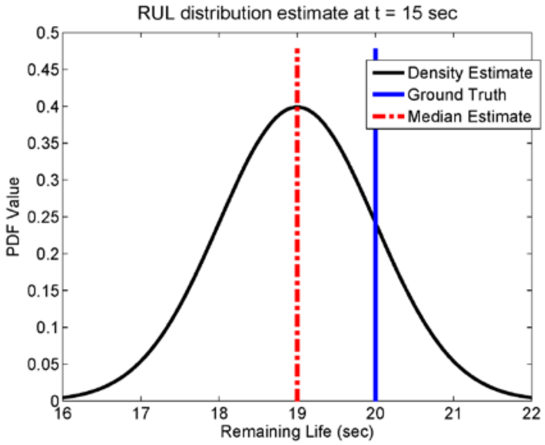


Figure 14. RUL density estimate at t = 15 sec

3.4. Prediction

After diagnosing the leak, estimation of the RUL distribution was performed at the time of prognosis, t_P, as per Section 2.3. The RUL distribution assumes a failure threshold for leakage area of 2E-6 m^2. The resulting RUL distribution is shown in Fig. 14.

3.5. Prognosis Validation

By continuing to estimate the new RUL distribution as new measurements become available, the performance of the prognostic algorithm may be evaluated. In Fig. 15, median RUL estimates are plotted against the ground truth RUL with +/- α bounds. The +/- α bounds are selected to be +/- 10% of the ground truth EoL about the current ground truth RUL. Figure 16 indicates whether the probability of the RUL estimate being between the +/- α bounds at a particular time is greater than a threshold value, taken as 0.8. From this information, it is also determined that the prognostic horizon is 10 seconds (or 20 time steps with a sampling frequency of 2 samples/sec) because the first time that $0.8 \le \pi\left[r\left(t_j\right)\right]_{-\alpha}^{+\alpha}$ is at t = 7, the EoL is t = 17. This is 10 seconds before the EoL.

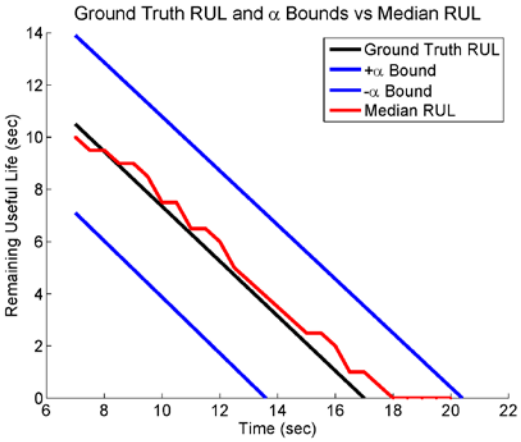


Figure 15. Ground truth RUL, median RUL, and α bounds with α = 0.10

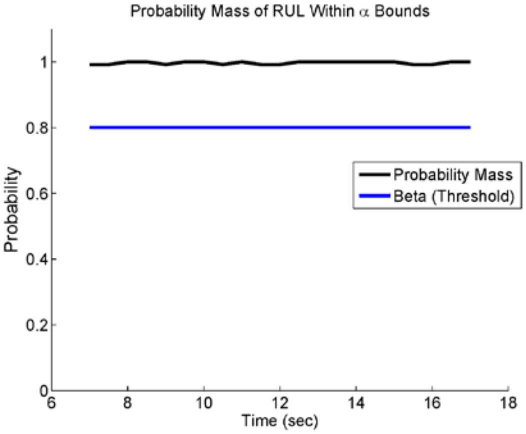


Figure 16. Probability that RUL is within α bounds with α = 0.10

+/- α bounds that narrow as the EoL approaches are considered in Fig. 17 for λ = 0.5 and α = 0.20. λ = 0.5 considers the accuracy of the RUL estimate halfway between the time of prognosis and end of life, termed t_λ. Figure 18 shows the λ-α accuracy, which is a binary value that indicates whether the probability of the RUL estimate being between the +/- α bounds at a particular time is greater than a threshold value, taken as 0.8 here. Although the median RUL estimate appears close to the ground truth

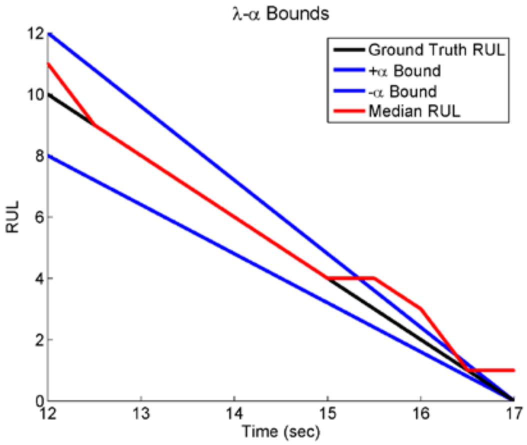


Figure 17. Bounds used for calculating λ-α accuracy with α = 0.20

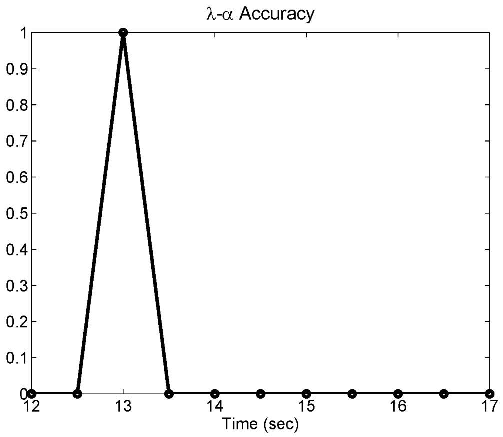


Figure 18. Probability RUL is within α bounds with α = 0.10

RUL, the λ-α accuracy is generally zero, indicating that the RUL estimate is too diffuse to pass this test. This indicates that model error in the physics of failure model is the dominant source of error as opposed to other errors that may decrease as the EoL is approached, such as errors in load estimation.

Based on the relative accuracy, the convergence is estimated to be 8.40. When comparing prognostic algorithms, larger convergence values are desirable.

Figure 19 shows the relative accuracy of the RUL density estimate based on the median RUL value, and shows that the median values are accurate.

3.6. Discussion

The DBN-based methodology successfully integrates heterogeneous sources of information to diagnose the system and estimate RUL. Particle-filter based inference

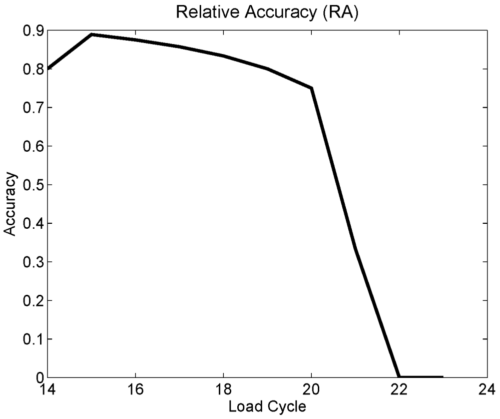


Figure 19. Relative accuracy based on median RUL estimate

provides a seamless method for switching between probabilistic diagnosis and prediction while facilitating uncertainty quantification.

The prognosis validation results indicate that the methodology provides reasonable median estimates of RUL, even as the RUL density estimates are diffuse. Additional measurements primarily affect the median RUL estimate, not the variance of RUL, primarily due to the simplifying assumptions that remove feedback from the actuator dynamic model into the leakage area model. Inclusion of inspection data may reduce the uncertainty in the leak area estimate and thus the RUL estimate. The accuracy of prognosis, of course, will vary depending on the system, available information, loading conditions, and environmental conditions.

Computational effort is a persistent issue in particle-based methodologies, affected by the complexity of the system, models involved, simplifying assumptions, filtering algorithms, etc. The prognosis methodology described in this paper is flexible with respect to these decisions, so computational effort will vary.

Thus far, the methodology has only been demonstrated using synthetic data, and needs to be tested further using real-world data. Further, more complex physics of failure models should be considered.

4. Conclusion

A methodology for DBN-based probabilistic prognosis is presented in this paper, considering heterogeneous information sources and diagnosis uncertainty. First, expert opinion is used to establish the system definition and basic assumptions. Reliability data is used to calculate conditional probabilities for fault indicator variables for damage at the support and a crack. Operational and laboratory data are organized in a database and used for estimating a polynomial regression model. This system model is used in online diagnosis via particle filter-based inference. The particles resulting from filtering integrate seamlessly into a sequential Monte Carlo predictive procedure, used for estimating RUL distribution. The prognosis results are validated using a four step hierarchical procedure. In the future, the methodology needs to be extended to systems of larger dimension, thus requiring feature selection, dimensional reduction, and more efficient inference.

ACKNOWLEDGEMENT

This work was partly supported by funds from the U. S. Air Force Research Laboratory, WPAFB through subcontract to General Dynamics Information Technology (Contract No. USAF-0060-43-0001, Project Monitor: Mark Derriso). Funding has also been provided by the NASA ARMD/AvSP IVHM project under NRA Award No. NNX09AY54A (Project Manager: Dr. K. Goebel, NASA Ames Research Center; subcontract to Clarkson University, Principal Investigator: Dr. Y. Liu). The support is gratefully acknowledged.

APPENDIX 1: BAYESIAN RECURSIVE FILTERING

Given the belief state $\sigma^t(\mathbf{x}^t)$, before obtaining $\mathbf{z}^{t+1}$, the belief state at *t* + 1 is

$$\sigma^{t+1}(\mathbf{x}^{t+1}) = p(\mathbf{x}^{t+1}|\mathbf{z}^{1:t}) \tag{13}$$

which may be expanded by summing over the states of $\mathbf{x}^t$ as

$$\sigma^{t+1}(\mathbf{x}^{t+1}) = \sum_{\mathbf{x}^t} p(\mathbf{x}^{t+1}|\mathbf{x}^t, \mathbf{z}^{1:t}) p(\mathbf{x}^t|\mathbf{z}^{1:t}) \tag{14}$$

Using the Markov assumption, which says that the future $\mathbf{x}^{t+1}$ is independent of all else given the previous state $\mathbf{x}^t$, the term $\mathbf{z}^{1:t}$ may be eliminated from $p(\mathbf{x}^{t+1}|\mathbf{x}^t, \mathbf{z}^{1:t})$, resulting in

$$\sigma^{t+1}(\mathbf{x}^{t+1}) = \sum_{\mathbf{x}^t} p(\mathbf{x}^{t+1}|\mathbf{x}^t) p(\mathbf{x}^t|\mathbf{z}^{1:t}) \tag{15}$$

where $p(\mathbf{x}^t|\mathbf{z}^{1:t}) = \sigma^t(\mathbf{x}^t)$. Upon receiving the measurement at time *t*+1, Bayes' rule may be used to update the belief state, and Eq. 3 becomes

$$\sigma^{t+1}(\mathbf{x}^{t+1}) = p(\mathbf{x}^{t+1}|\mathbf{z}^{1:t}, \mathbf{z}^{t+1}) \tag{16}$$

By Bayes' rule expansion of the right hand side of Eq. 4,

$$\sigma^{t+1}(\mathbf{x}^{t+1}) = \frac{p(\mathbf{z}^{t+1}|\mathbf{x}^{t+1}, \mathbf{z}^{1:t}) p(\mathbf{x}^{t+1}|\mathbf{z}^{1:t})}{p(\mathbf{z}^{t+1}|\mathbf{z}^{1:t})} \tag{17}$$

Because the measurements $\mathbf{z}^{t+1}$ and $\mathbf{z}^{1:t}$ are conditionally independent given $\mathbf{x}^{t+1}$ (Section 2.2 Fig. 2a), $p(\mathbf{z}^{t+1}|\mathbf{x}^{t+1}, \mathbf{z}^{1:t}) = p(\mathbf{z}^{t+1}|\mathbf{x}^{t+1})$, resulting in

$$\sigma^{t+1}(\mathbf{x}^{t+1}) = \frac{p(\mathbf{z}^{t+1}|\mathbf{x}^{t+1}) p(\mathbf{x}^{t+1}|\mathbf{z}^{1:t})}{p(\mathbf{z}^{t+1}|\mathbf{z}^{1:t})} \tag{18}$$

where $p(\mathbf{x}^{t+1}|\mathbf{z}^{1:t})$ is equivalent to Eq. 15.

APPENDIX 2: HYDRAULIC ACTUATOR MODEL

Parameters and variables for the system are given in Table 1A.

$$\dot{x}_{act} = v_{act} \tag{19}$$

$$\dot{v}_{act} = \frac{1}{m}\left[(P_1 - P_2)A_{pist} - b_{act}v_{act} - k_{act}x_{act} - F_{ext}\right] \tag{20}$$

$$\dot{P}_1 = \frac{1}{C_{f1}}\left(Q_1 - A_{pist}Q_2 + Q_{leak}\right) \tag{21}$$

$$\dot{P}_2 = \frac{1}{C_{f2}}\left(V_2 x_{act} - Q_2 - Q_{leak}\right) \tag{22}$$

$$\dot{x}_{valve} = v_{valve} \tag{23}$$

$$\dot{v}_{valve} = a_1 v_{valve} + a_0 x_{valve} + b_0 e_{command} \tag{24}$$

$$C_{f1} = \frac{V_1(x_{act})}{\beta} \tag{25}$$

$$C_{f2} = \frac{V_2(x_{act})}{\beta} \tag{26}$$

$$\text{If } x_{valve} > 0, \begin{cases} Q_1 = C_d w_{valve} x_{valve} sign(P_s - P_1)\sqrt{\frac{2}{\rho}|P_s - P_1|} & (27) \\ Q_2 = C_d w_{valve} x_{valve}\sqrt{\frac{2}{\rho}(P_2)} & (28) \end{cases}$$

$$\text{If } x_{valve} < 0, \begin{cases} Q_1 = C_d w_{valve} x_{valve}\sqrt{\frac{2}{\rho}(P_1)} & (29) \\ Q_2 = C_d w_{valve} x_{valve} sign(P_s - P_2)\sqrt{\frac{2}{\rho}|P_s - P_2|} & (30) \end{cases}$$

$$Q_{leak} = C_d a_{leak}\sqrt{\frac{2}{\rho}|P_2 - P_1|} sign(P_2 - P_1) \tag{31}$$

Table 1A. Model parameters and variables for a spool valve and a hydraulic actuator.

Parameter/variable	**Symbol**	**Nominal Value/ Unit**
Actuator position	x_{act}	m
Actuator velocity	v_{act}	m/s
Servovalve position	x_{valve}	m
Servovalve velocity	v_{valve}	m/s
Pressure in chamber 1	P_1	Pa
Pressure in chamber 2	P_2	Pa
Combined mass of actuator and load	m_{act}	40 kg
Combined damping of actuator and load	b_{act}	800 Ns/m
Combined stiffness of actuator and load	k_{act}	10^6 N/m
Piston annulus area	A_{pist}	0.0075 m^2
Valve port width	w_{valve}	0.0025 m
Spool valve model coefficients	b_0	90 m/Vs^2
	a_0	360,000 $1/s^2$
	a_1	1/s
Hydraulic fluid bulk modulus	β	1000 MPa
Hydraulic fluid density	ρ	847 kg/m^3
Discharge coefficient	C_d	0.7
Supply pressure	P_{supply}	20 MPa
Chamber 1 volume	V_1	m^3
Chamber 2 volume	V_2	m^3
Chamber 1 fluid capacitance	C_{f1}	$m^3/(kg/s)$
Chamber 2 fluid capacitance	C_{f2}	$m^3s/(kg/s)$
Volumetric flow rate into chamber 1	Q_1	m^3/s
Volumetric flow rate out of chamber 2	Q_2	m^3/s
Externally applied force	F_{ext}	0 N
Input voltage	$e_{command}$	Sin(2*pi*t) V
Leakage volumetric flow rate	Q_{leak}	0 m^3/s
Leakage area	a_{leak}	0 m^2

REFERENCES

Andrieu, C., Davy, M., & Doucet, A. (2003). Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions. *IEEE Transactions on Signal Processing*, *51*(7), 1762–1770. doi:10.1109/TSP.2003.810284

Banjevic, D., & Jardine, A. K. S. (2006). Calculation of reliability function and remaining useful life for a Markov failure time process. *IMA Journal of Management Mathematics*, *17*(2), 115–130. doi:10.1093/imaman/dpi029

Bartram, G., & Mahadevan, S. (2013). Integration of Heterogeneous Information in SHM Models. *Structural Control and Health Monitoring*, Accepted.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2008). *Time series analysis: forecasting and control*. Hoboken, N.J.: John Wiley.

Boyen, X., & Koller, D. (1998). Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence* (pp. 33–42). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=2074094.2074099

Briscoe, B. (1981). Wear of polymers: an essay on fundamental aspects. *Tribology International*, *14*(4), 231–243. doi:10.1016/0301-679X(81)90050-5

Briscoe, B. J., & Sinha, S. K. (2002). Wear of polymers. *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, *216*(6), 401–413. doi:10.1243/135065002762355325

Chen, Z. (2003). Bayesian filtering: From Kalman filters to particle filters, and beyond. *Statistics*, 1–69.

Chinnam, R. B., & Baruah, P. (2003). Autonomous diagnostics and prognostics through competitive learning driven HMM-based clustering. In *Proceedings of the International Joint Conference on Neural Networks, 2003* (Vol. 4, pp. 2466–2471 vol.4). Presented at the Proceedings of the International Joint Conference on Neural Networks, 2003. doi:10.1109/IJCNN.2003.1223951

Dong, M., & Yang, Z. (2008). Dynamic Bayesian network based prognosis in machining processes. *Journal of Shanghai Jiaotong University (Science)*, *13*(3), 318–322. doi:10.1007/s12204-008-0318-y

Doucet, A., Godsill, S., & Andrieu, C. (2000). On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. *STATISTICS AND COMPUTING*, *10*(3), 197–208.

Farrar, C. R., & Worden, K. (2012). *Structural Health Monitoring: A Machine Learning Perspective*. John Wiley & Sons.

Friedman, N., Murphy, K., & Russell, S. (1998). Learning the Structure of Dynamic Probabilistic Networks. In *UAI'98 Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (pp. 139–147). Morgan Kaufmann Publishers Inc. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.6615

Gebraeel, N. Z., & Lawley, M. A. (2008). A neural network degradation model for computing and updating residual life distributions. *Automation Science and Engineering, IEEE Transactions on*, *5*(1), 154–163.

Goebel, K., Saha, B., Saxena, A., Mct, N., & Riacs, N. (2008). A comparison of three data-driven techniques for prognostics. In *62nd Meeting of the Society For*

Machinery Failure Prevention Technology (MFPT) (pp. 119–131).

Goode, K. B., Moore, J., & Roylance, B. J. (2000). Plant machinery working life prediction method utilizing reliability and condition-monitoring data. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, *214*(2), 109–122. doi:10.1243/0954408001530146

Heckerman, D., & Geiger, D. (1995). Learning Bayesian Networks: A unification for discrete and Gaussian domains. *PROCEEDINGS OF ELEVENTH CONFERENCE ON UNCERTAINTY INARTI CIAL INTELLIGENCE*. Retrieved from http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.156.7976

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510. doi:10.1016/j.ymssp.2005.09.012

Jinlin, Z., & Zhengdao, Z. (2012). Fault prognosis for data incomplete systems: A dynamic Bayesian network approach. In *Control and Decision Conference (CCDC), 2012 24th Chinese* (pp. 2244–2249). Presented at the Control and Decision Conference (CCDC), 2012 24th Chinese. doi:10.1109/CCDC.2012.6244360

Jones, M. H. (Ed.). (1983). *Industrial Tribology: The Practical Aspects of Friction, Lubrication and Wear*. North Holland.

Kacprzynski, G. J., Sarlashkar, A., Roemer, M. J., Hess, A., & Hardman, B. (2004). Predicting remaining life by fusing the physics of failure modeling with diagnostics. *JOM*, *56*(3), 29–35. doi:10.1007/s11837-004-0029-2

Karpenko, M., & Sepehri, N. (2003). Robust Position Control of an Electrohydraulic Actuator With a Faulty Actuator Piston Seal. *Journal of Dynamic Systems, Measurement, and Control*, *125*(3), 413–423. doi:10.1115/1.1592194

Khan, T., Udpa, L., & Udpa, S. (2011). Particle filter based prognosis study for predicting remaining useful life of steam generator tubing. In *Prognostics and Health Management (PHM), 2011 IEEE Conference on* (pp. 1 –6). doi:10.1109/ICPHM.2011.6024323

Khedkar, J., Negulescu, I., & Meletis, E. I. (2002). Sliding wear behavior of PTFE composites. *Wear*, *252*(5–6), 361–369. doi:10.1016/S0043-1648(01)00859-6

Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Kozlowski, J. D. (2003). Electrochemical cell prognostics using online impedance measurements and model-based data fusion techniques. In *2003 IEEE Aerospace Conference, 2003. Proceedings* (Vol. 7, pp. 3257–3270). Presented at the 2003 IEEE Aerospace Conference, 2003. Proceedings. doi:10.1109/AERO.2003.1234169

Kulakowski, B. T., Gardner, J. F., & Shearer, J. L. (2007). *Dynamic modeling and control of engineering systems*. Cambridge University Press.

Kwan, C., Zhang, X., Xu, R., & Haynes, L. (2003). A novel approach to fault diagnostics and prognostics. In *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA ’03* (Vol. 1, pp. 604–609 vol.1). Presented at the IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA ’03. doi:10.1109/ROBOT.2003.1241660

Lancaster, J. K. (1969). Abrasive wear of polymers. *Wear*, *14*(4), 223–239. doi:10.1016/0043-1648(69)90047-7

Lauritzen, S. L. (1992). Propagation of Probabilities, Means, and Variances in Mixed Graphical Association Models. *Journal of the American Statistical Association*, *87*(420), 1098–1108. doi:10.2307/2290647

Lin, D., & Makis, V. (2004). Filters and parameter estimation for a partially observable system subject to random failure with continuous-range observations. *Advances in Applied Probability*, *36*(4), 1212–1230. doi:10.1239/aap/1103662964

Liu, J., Saxena, A., Goebel, K., Saha, B., & Wang, W. (2010). *An Adaptive Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-ion Batteries*. DTIC Document.

Lorton, A., Fouladirad, M., & Grall, A. (2013). A methodology for probabilistic model-based prognosis. *European Journal of Operational Research*, *225*(3), 443–454. doi:10.1016/j.ejor.2012.10.025

MacCormick, B. W. (1995). *Aerodynamics, Aeronautics, and Flight Mechanics*. John Wiley & Sons, Incorporated.

Mahulkar, V., McGinnis, H., Derriso, M., & Adams, D. E. (2010). *Fault Identification in an Electro-Hydraulic Actuator and Experimental Validation of Prognosis Based Life Extending Control*. DTIC Document.

Naval Surface Warfare Center. (2011). *Handbook of Reliability Prediction Procedures for Mechanical Equipment*. West Bethesda, Maryland 20817-5700. Retrieved from http://www.navsea.navy.mil/nswc/carderock/pub/mechrel/products/handbook.aspx

Nikas, G. K. (2010). Eighty years of research on hydraulic reciprocating seals: Review of tribological studies and related topics since the 1930s. *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, *224*(1), 1–23. doi:10.1243/13506501JET607

Orchard, M. E., & Vachtsevanos, G. J. (2009). A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*. doi:10.1177/0142331208092026

Pitt, M. K., & Shephard, N. (1999). Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, *94*(446), 590–599. doi:10.1080/01621459.1999.10474153

RIAC Automated Databook. (2006). Reliability Information Analysis Center. Retrieved from http://www.theriac.org

Ristic, B., & Arulampalam, S. (2004). *Beyond the Kalman filter : particle filters for tracking applications*. Boston, MA: Artech House.

Saha, B., Celaya, J. R., Wysocki, P. F., & Goebel, K. F. (2009). Towards prognostics for electronics components. In *2009 IEEE Aerospace conference* (pp. 1–7). Presented at the 2009 IEEE Aerospace conference. doi:10.1109/AERO.2009.4839676

Saha, Bhaskar, Goebel, K., & Christophersen, J. (2009). Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement and Control*, *31*(3-4), 293–308.

Sankararaman, S., & Mahadevan, S. (n.d.). Uncertainty quantification in structural damage diagnosis. *Structural Control and Health Monitoring*. doi:10.1002/stc.400

Sawyer, W. G., Freudenberg, K. D., Bhimaraj, P., & Schadler, L. S. (2003). A study on the friction and wear behavior of PTFE filled with alumina nanoparticles. *Wear*, *254*(5–6), 573–580. doi:10.1016/S0043-1648(03)00252-7

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for Offline Evaluation of Prognostic Performance. *International Journal of Prognositcs and Health Management*, (1). Retrieved from http://www.phmsociety.org/sites/phmsociety.org/files/phm_submission/2010/ijPHM_10_001.pdf

Sepeheri, N., Karpenko, M., An, L., & Karam, S. (2005). A test rig for experimentation on fault tolerant control and condition monitoring algorithms in fluid power systems: from designing through implementation. *Transactions of the Canadian Society for Mechanical Engineering*, *29*(3), 441–458.

Thompson, D. F., Pruyn, J. S., & Shukla, A. (1999). Feedback design for robust tracking and robust stiffness in flight control actuators using a modified QFT technique. *International Journal of Control*, *72*(16), 1480 – 1497.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, *1*, 211–244.

Tobon-Mejia, D. A., Medjaher, K., Zerhouni, N., & Tripot, G. (2012). A Data-Driven Failure Prognostics Method Based on Mixture of Gaussians Hidden Markov Models. *Reliability, IEEE Transactions on*, *61*(2), 491–503.

Tran, V. T., & Yang, B.-S. (2009). Machine Fault Diagnosis and Prognosis: The State of The Art. *The International Journal of Fluid Machinery and Systems (IJFMS)*, *2*(1), 61–71.

Vachtsevanos, G., & Wang, P. (2001). Fault prognosis using dynamic wavelet neural networks. In *AUTOTESTCON Proceedings, 2001. IEEE Systems Readiness Technology Conference* (pp. 857–870). Presented at the AUTOTESTCON Proceedings, 2001. IEEE Systems Readiness Technology Conference. doi:10.1109/AUTEST.2001.949467

Vlok, P.-J., Wnek, M., & Zygmunt, M. (2004). Utilising statistical residual life estimates of bearings to quantify the influence of preventive maintenance actions. *Mechanical Systems and Signal Processing*, *18*(4), 833–847. doi:10.1016/j.ymssp.2003.09.003

Wang, W. Q., Golnaraghi, M. F., & Ismail, F. (2004). Prognosis of machine health condition using neuro-fuzzy systems. *Mechanical Systems and Signal Processing*, *18*(4), 813–831. doi:10.1016/S0888-3270(03)00079-7

Wang, W., Scarf, P. A., & Smith, M. a. J. (2000). On the application of a model of condition-based maintenance. *Journal of the Operational Research Society*, *51*(11), 1218–1227. doi:10.1057/palgrave.jors.2601042

Wang, Wenbin. (2002). A model to predict the residual life of rolling element bearings given monitored condition information to date. *IMA Journal of Management Mathematics*, *13*(1), 3–16. doi:10.1093/imaman/13.1.3

West, M., & Harrison, J. (1997). Bayesian Forecasting and Dynamic Models (Springer Series in Statistics).

Yam, R. C. M., Tse, P. W., Li, L., & Tu, P. (2001). Intelligent Predictive Decision Support System for Condition-Based Maintenance. *The International Journal of Advanced Manufacturing Technology*, *17*(5), 383–391. doi:10.1007/s001700170173

Yan, J., Koç, M., & Lee, J. (2004). A prognostic algorithm for machine performance assessment and its application. *Production Planning & Control*, *15*(8), 796–801. doi:10.1080/09537280412331309208

Zeiger, G., & Akers, A. (1986). Dynamic analysis of an axial piston pump swashplate control. *ARCHIVE: Proceedings of the Institution of Mechanical Engineers, Part C: Mechanical Engineering Science 1983-1988 (vols 197-202)*, *200*(13), 49–58. doi:10.1243/PIME_PROC_1986_200_093_02

Zeliang, L. (2005, November). *Condition Monitoring of Axial Piston Pump* (Master's Thesis). University of Saskatchewan, Saskatoon, Saskatchewan, Canada. Retrieved from http://library2.usask.ca/theses/available/etd-11252005-202705/unrestricted/EricLithesis2005NovA.pdf

Zhang, S., & Ganesan, R. (1997). Multivariable Trend Analysis Using Neural Networks for Intelligent Diagnostics of Rotating Machinery. *Journal of Engineering for Gas Turbines and Power*, *119*(2), 378–384. doi:10.1115/1.2815585

Fault Detection and Isolation for Autonomous Parafoils

Matthew R. Stoeckle[1], Amer Fejzic[2], Louis S. Breger[3], and Jonathan P. How[4]

[1,4] *Massachusetts Institute of Technology, Cambridge, MA, 02139, USA*
stoke927@mit.edu
jhow@mit.edu

[2,3] *Draper Laboratory, Cambridge, MA, 02459, USA*
afejzic@draper.com
lbreger@draper.com

Abstract

Autonomous precision airdrop systems are widely used to deliver supplies to remote locations. Payloads that are delivered far from their intended targets or with high impact velocity may be rendered unusable. Faults occurring during flight can severely degrade vehicle performance, effectively nullifying the value of the guided system, or worse. Quickly detecting and identifying faults enables the choice of an appropriate recovery strategy, potentially mitigating the consequences of an out-of-control vehicle and recovering performance. This paper presents a multi-observer, multi-residual fault detection and isolation (FDI) method for an autonomous parafoil system. The detection and isolation processes use residual signals generated from observers and other system models. Statistical methods are applied to evaluate these residuals and determine whether a fault has occurred, given a priori knowledge of system uncertainty characteristics. Several examples are used to illustrate the detection and isolation algorithm online using available navigation and telemetry outputs. Tests of this FDI method on a large number of high-fidelity simulations indicate that it is possible to detect and isolate some common faults with a high percentage of success and a very small chance of raising a false alarm.

1. Introduction

Autonomous precision airdrop is used to deliver payloads to areas that would be dangerous or difficult to reach through more conventional means. Missions for guided parafoils include military resupply of troops and humanitarian efforts (Hattis & Tavan, 2007). As described in Hattis, Campbell, Carter, McConley, and Tavan (2006), the goal of the system is to land the payload as close as possible to the target while minimizing ground speed at impact. Flight testing has shown that a variety of faults can occur (Tavan, 2006). These faults increase target miss distances and landing speeds, potentially rendering payloads unusable. In addition, the possibility of an in-flight fault and resulting behavior could preclude delivering supplies to more densely populated areas where an out-of-control vehicle could pose a danger to persons or property. Detecting, isolating, and responding to faults can improve performance and expand the space of missions available for guided parafoils. This work designs and implements a Fault Detection and Isolation (FDI) strategy that is effective in the unique conditions under which the parafoil operates.

Online systems for FDI fall into two categories: those that exploit hardware redundancy and those that rely on analytical redundancy (Hwang, Kim, Kim, & Seah, 2010). Systems with a large number of sensors, actuators, and measurements employ hardware redundancy for FDI or system health management (Figueroa et al., 2009) (Figueroa, Schmalzel, Morris, Turowski, & Franzl, 2010). The parafoil has a minimal number of sensors, and so analytical redundancy methods are used.

Isermann and Ballé (1997) define FDI terminology. A fault is defined as an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition. Fault detection is the determination of the faults present in a system and the time of detection. Fault isolation is the determination of the kind, location, and time of detection of a fault. The process of isolation follows that of detection.

For FDI to be effective, 1) the effects of faults must be distinguishable from the effects of unknown inputs including modeling errors, disturbances, and measurement uncertainty, and 2) faults must be distinguishable from each other (Frank, 1994). This is typically accomplished by considering a resid-

Matthew Stoeckle et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ual signal (Hwang et al., 2010). The residual signal chosen has approximately zero mean when no fault is present and nonzero mean when a fault has occurred. In this context, a residual signal is the difference between a measurable system output and the corresponding expected output. After the residual has been generated, it is evaluated. The goal of the evaluation process is to determine whether a fault alarm should be raised based on the properties of the residual signal.

A large group of FDI methods are classified as observer-based. These methods use an observer of the nominal system to generate the expected system output. This output is used along with measurements from the actual system to generate the residual signal. Though a simulation of the system with no feedback can also be used to generate the residual signal, an observer is chosen to make the residual generation process robust to differences in initial conditions.

A common method of residual generation that uses observers is called the fault detection filter (FDF). This method generates a residual signal that is projected onto subspaces associated with various faults, so that detection and isolation are both possible (Beard, 1971) (Jones, 1973). See Douglas and Speyer (1995) for a robust implementation of the FDF. For isolation, the FDF requires that each fault under consideration acts on the system in a known, unique way. This is not the case for the parafoil system; many faults act on the control lines and are not distinguishable from each other.

The eigenstructure assignment approach is used to de-couple effects of disturbances from those of faults by nulling the transfer function from the disturbances to the residual signal (Patton & Chen, 2000). A weighting matrix that is used to assign eigenvectors to the closed-loop observer of the system accomplishes this task. In order to construct this weighting matrix, however, there must be more independent outputs of the system than independent disturbances (Patton & Chen, 2000). The parafoil system is a single-output system, so eigenstructure assignment is not possible.

The FDI method presented in this paper is observer-based, but takes a different approach than the FDF. Many existing observer-based methods incorporate isolation into the detection process by exploiting the system property that each fault under consideration is distinguishable from all other faults (Frank, 1994). However, this is not the case for many faults that occur on the parafoil system. As a result, the detection and isolation processes are separate for this work.

For detection, a residual signal is generated using observer-based methods. This residual is evaluated using hypothesis testing. If the magnitude of the residual signal crosses above a predetermined detection threshold, a fault is declared. Sargent et al. (2011) use hypothesis testing with thresholds for FDI on the Orbital Cygnus vehicle. Rossi (2012) uses hypothesis testing for health management of spacecraft.

If a fault is declared, isolation is performed. In this paper, isolation is broken into two phases. The first phase uses a residual signal from actuator data. If evaluation of this signal indicates that an actuator fault has occurred, isolation is complete. However, if the first phase of isolation does not declare an actuator fault, phase two begins.

Phase two of isolation uses a bank of fault-specific observers to differentiate between non-actuator faults. The purpose of these observers is to determine when the system exhibits characteristics of a particular fault (Willsky, 1976). Evaluating residual signals from these observers indicates if a specific non-actuator fault is present. Successful isolation will result in the declaration of a fault on one of the actuators or the declaration of a particular non-actuator fault.

Section 2 gives an overview of how the parafoil and payload system operates. Section 3 describes common faults that have been observed in flight tests. Sections 4 and 5 describe the detection and isolation methods, respectively. Section 6 combines detection and isolation into the full FDI algorithm. Section 7 concludes the paper.

2. Parafoil System Overview

A typical system consists of a canopy, airborne guidance unit (AGU), and payload (Figure 1). An example of a parafoil system, as well as some performance characteristics, is given in Bergeron, Fejzic, and Tavan (2011).

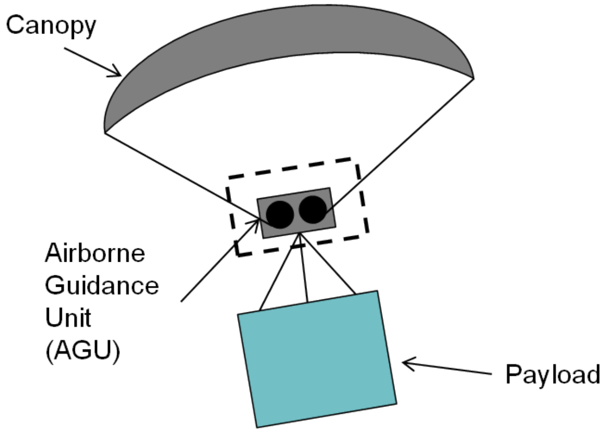


Figure 1. Parafoil canopy, AGU, and payload.

Lateral control of the parafoil is accomplished using two control lines that attach to the left and right trailing edges of the canopy. These lines are wound around two motors on the AGU (Figure 2). Details of guidance, navigation, and control (GN&C) implementation on the parafoil system are described by Carter, George, Hattis, Singh, and Tavan (2005).

The motors retract and extend the control lines, deflecting the trailing edges of the canopy and inducing a nonzero turn rate. The motors on the AGU are equipped with encoders that mea-

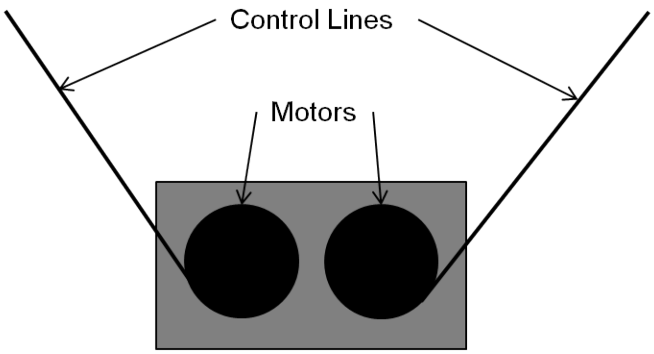


Figure 2. Parafoil AGU and control lines.

sure the deflections of the control lines. The resulting performance of the parafoil is estimated using this information. A payload is attached underneath the AGU by several support lines.

Different dynamical models of varying complexity have been developed for the parafoil and payload system (Barrows, 2009) (Ward, Montalvo, & Costello, 2010). The observer that will be used for FDI is based on a linearized model of the lateral dynamics, derived from nonlinear dynamics in Crimi (1990). Lateral dynamics were chosen because they accurately reflect the heading rate response of the system while allowing for a straightforward linear model. A similar linearized model of lateral parafoil dynamics is used in Slegers and Costello (2004) for model predictive control, and is applicable to observer-based FDI as well. The model used in this paper is described by the linear, time-invariant (LTI) system,

$$\dot{x}(t) = (A + \Delta A)x(t) + (B + \Delta B)u(t) + Wd(t) \quad (1)$$

$$y(t) = Cx(t) + \eta(t) \quad (2)$$

where $A, \Delta A \in \mathbb{R}^{4x4}$, $B, \Delta B \in \mathbb{R}^{4x2}$, $C \in \mathbb{R}^{1x4}$, and $W \in \mathbb{R}^4$. A, B, and C are known dynamics, control, and output matrices, respectively. The matrix B can be written as $\begin{bmatrix} b_1 & -b_1 \end{bmatrix}$, where $b_1 \in \mathbb{R}^4$ (i.e., both motors affect the system equally, but in opposite directions). The matrix W determines how the process noise acts on the system states. The matrices ΔA and ΔB represent unknown modeling errors. The matrices A and B are determined from known system parameters. The example parafoil used for simulation in this work has the following parameters: canopy weight 70 lbs, canopy area 900 ft^2, canopy span 50 ft, canopy chord length 16 ft, and nominal payload weight 1800 lbs. These parameters are among many used to determine the linearized lateral dynamics.

The states of the LTI system in Eqs. (1) and (2) are $x(t) = \begin{bmatrix} \beta(t) & \dot{\phi}(t) & \phi(t) & \dot{\psi}(t) \end{bmatrix}^T$, where $\beta(t)$ is the sideslip angle, $\dot{\phi}(t)$ is the roll rate, $\phi(t)$ is the roll angle, and $\dot{\psi}(t)$ is the yaw rate. The control input is $u(t) = \begin{bmatrix} \delta_R(t) & \delta_L(t) \end{bmatrix}^T$, where $\delta_R(t)$ and $\delta_L(t)$ are the right and left motor deflections, or motor toggles, respectively. Under healthy conditions, the deflection at each motor will match the corresponding deflection of the control line. This will not be the case when some faults occur. The effects of these faults on the system dynamics are discussed in Section 3. The disturbance term $d(t) \in \mathbb{R}$ is the process noise of the system. The chosen output $y(t) \in \mathbb{R}$ is the heading rate of the system, which is subject to uncertainty that is captured in the navigation error term $\eta(t) \in \mathbb{R}$. Heading rate in the context of the parafoil is defined as the rate at which the airspeed velocity vector of the parafoil rotates with respect to the inertial North axis (see Figure 3).

The effects of the uncertainty and noise terms, ΔA, ΔB, $d(t)$, and $\eta(t)$, on the FDI process cause the residual signals to be nonzero even when no fault has occurred. However, the size of the residual during a healthy flight is small compared to the size of the signal when a fault has occurred. In other words, faults are still observable even if the noise and uncertainty terms are neglected. Therefore, neglecting these four terms and formulating the FDI problem using an observer-based approach as opposed to a Kalman filter-based approach was chosen to minimize the computational complexity.

After ignoring these terms, the lateral system dynamics reduce to,

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3)$$

$$y(t) = Cx(t) \quad (4)$$

This simpler linearized lateral dynamics model will be used for residual generation.

The parafoil system has no sensors for measuring heading rate directly; instead, this quantity is estimated using an Extended Kalman Filter (EKF). The only state information that the parafoil software has access to is position and translational velocity data from the onboard GPS. The GPS measures the ground speed as well as the sink rate of the parafoil. An EKF is used to estimate the wind velocity, and from this information the airspeed velocity and the heading rate are estimated, similarly to work done in Ward et al. (2010). The system states used in Eqs. (1-4) are not available from the EKF and are unknown.

Figure 3 shows some parafoil states and provides insight into the estimation of heading rate. The GPS measures the ground velocity V_g. The EKF estimates the wind velocity V_w. The airspeed V_a is estimated using vector addition, and from that an estimate of the heading angle χ is obtained. The EKF uses this information to generate an estimate of the heading rate $\dot{\chi}$.

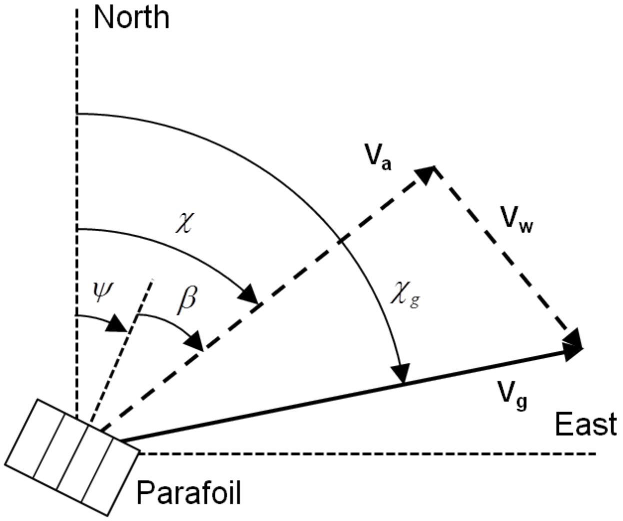


Figure 3. Parafoil states and reference frame.

3. Common Faults

Flight testing of the parafoil system has shown that some faults occur frequently (Tavan, 2006). A hierarchy of common faults is shown in Figure 4. Four of the faults shown in the hierarchy are considered for FDI. These faults are chosen because they have effects that are both well-defined and well-understood. The faults are: stuck motor, severe saturation, broken control line, and criss-crossed control lines.

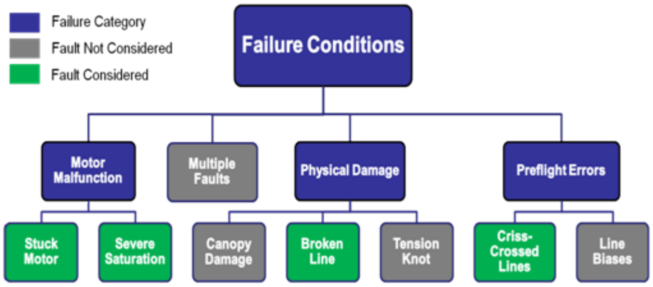


Figure 4. Fault hierarchy.

A stuck motor fault occurs when one of the motors on the AGU fails to respond to commands. When this fault occurs, the line attached to the faulty motor is stuck at an unknown position while the other control line is free to move as commanded. If detection fails, motor commands will be given that the motor will not be able to respond to.

Under nominal conditions, a known motor saturation limit exists. This limit is the maximum line deflection possible for each motor. A severe saturation fault occurs when the maximum possible line deflection is significantly less than expected; a saturation limit of 0.65 meters is considered significant for this work. If this fault is not detected, it is possible that a large command will be given but the corresponding deflection will not occur. At that point, the system will likely continue to command a large deflection to achieve a desired heading rate that it can never attain.

A broken line fault occurs when one of the control lines that are attached to the motors on the AGU breaks. In this case, the motor is still free to turn, but there is no corresponding response in line deflection. This prevents the parafoil from turning in the direction of the side on which the line is broken. This fault often occurs upon canopy deployment.

It is possible, while rigging the lines to the AGU, that the control line attached to the left trailing edge of the parafoil is spooled around the right motor, and vice versa. In this case, a command to the right motor will yield a deflection in the left control line, and a command to the left motor will yield a deflection in the right control line. This fault is called criss-crossed lines. This is an example of a fault that has a straightforward recovery strategy. No change to the existing guidance strategy is necessary; the controller need only reverse the commands given to each motor to achieve the desired performance. However, this recovery approach cannot be implemented unless FDI successfully detects and isolates the fault.

The non-actuator faults are added to the linearized lateral model of the parafoil dynamics in Eq. (3) as follows,

$$\dot{x}(t) = Ax(t) + (B + \Delta B_f)u(t) \tag{5}$$

where $\Delta B_f \in \mathbb{R}^{4x2}$ are changes to the dynamics that occur when either a broken control line or criss-crossed lines fault is present. When a broken left line occurs, $\Delta B_f = \begin{bmatrix} 0 & b_1 \end{bmatrix}$. When a broken right line occurs, $\Delta B_f = \begin{bmatrix} -b_1 & 0 \end{bmatrix}$. When a criss-crossed lines fault occurs, $\Delta B_f = \begin{bmatrix} -2b_1 & 2b_1 \end{bmatrix}$. When any of these non-actuator faults occur, the deflections of the left and right control lines will not match the motor toggles in $u(t)$.

Actuator faults enter the system in a different way. The motor toggles $u(t)$ result from passing commanded toggles $u_{cmd}(t)$ into the motors on the AGU. Therefore, actuator faults (i.e. stuck motor and severe saturation) are not modeled in Eq. (5) but instead manifest themselves in a value of $u(t)$ that is different from what is expected. Actuator faults are identified by comparing $u(t)$ to $u_{nom}(t)$, the nominal, or expected, motor toggle. This term is introduced in Section 4.

4. Detection

Fault detection is the process of determining the faults present in a system and the time at which those faults occurred (Isermann & Ballé, 1997). A detection alarm is raised for any fault; it is not necessary during detection to know which particular fault has occurred. Detection is accomplished by comparing a known system output with an expected system output. The difference between these two quantities is a residual signal. This signal should be chosen such that it is large when a fault is present and small otherwise (Frank, 1994). If the residual signal is large, a fault is declared. This paper uses an observer-based fault detection method, where the expected system output is generated using an observer. This observer is designed to model the parafoil and payload system when no faults are present.

Detection is broken into two phases: residual generation and residual evaluation (Hwang et al., 2010). Residual generation is the process of constructing the residual signal. Residual evaluation is the process of taking this signal and using it to either validate or reject a null hypothesis. The null hypothesis is that the system is healthy; a rejection of this hypothesis indicates a fault. Residual evaluation is performed using a threshold, which is designed so that if the residual signal rises above this threshold there is a reasonable probability that a fault is present (Frank, 1994),

$$\text{If } r(t) \leq \lambda_{th}, \text{ null hypothesis confirmed} \tag{6}$$

$$\text{If } r(t) > \lambda_{th}, \text{ null hypothesis rejected; fault} \tag{7}$$

where $r(t) \in \mathbb{R}$ is a time-varying residual signal, and $\lambda_{th} \in \mathbb{R}$ is a mission-specific constant threshold value. The FDI method in this work uses the parafoil heading rate output for residual generation (see Section 2). The parafoil guidance system commands the parafoil by specifying a desired heading rate. If the parafoil is not tracking the heading rate as expected, the system is likely in a faulty condition.

4.1. Residual Generation

To generate the heading rate residual signal, an observer is constructed based on the linearized lateral dynamics described in Eqs. (3) and (4). Figure 5 shows that a motor toggle command u_{cmd} is passed through both the AGU motor, which is subject to actuator faults, and a model of a healthy motor.

The output of the AGU motor is the actual motor toggle u; the output of the motor model is the nominal motor toggle u_{nom}. The difference between these two signals is small when no actuator faults are present. The nominal input u_{nom} is used as the input to the observer, which is constructed as follows,

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu_{nom}(t) + L(y(t) - \hat{y}(t)) \tag{8}$$

$$\hat{y}(t) = C\hat{x}(t) \tag{9}$$

where $\hat{x}(t) \in \mathbb{R}^4$ is an estimate of the system states $x(t)$, $\hat{y}(t) \in \mathbb{R}$ is the observer estimate of the heading rate, and A, B, and C are the matrices from the parafoil dynamics in Eqs. (3) and (4). The feedback gain $L \in \mathbb{R}^4$ is designed to make $A - LC$ stable. Error terms are defined as,

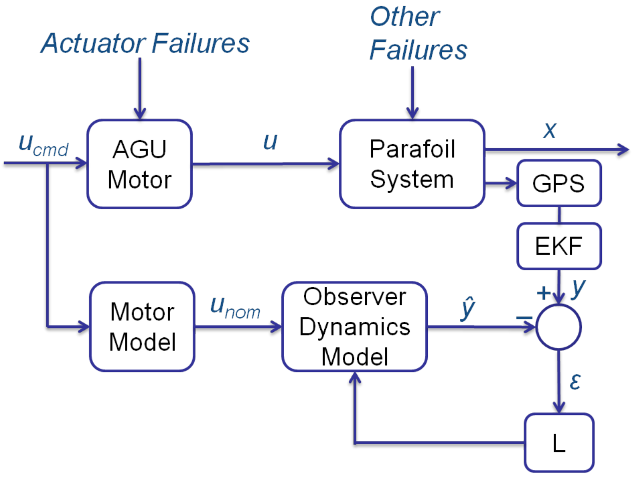


Figure 5. Heading rate observer block diagram.

$$e(t) = \hat{x}(t) - x(t) \tag{10}$$

$$\dot{e}(t) = \dot{\hat{x}}(t) - \dot{x}(t) \tag{11}$$

$$\varepsilon(t) = \hat{y}(t) - y(t) = Ce(t) \tag{12}$$

and the residual signal used for fault detection is chosen as,

$$r(t) = \varepsilon^2(t) \tag{13}$$

Squaring $\varepsilon(t)$ ensures that the residual signal is non-negative. This aids the residual evaluation process.

Error dynamics of the observer/plant system can yield insight into the behavior of the residual signal. Though the error $e(t)$ is not measurable, Eq. (12) shows that $\varepsilon(t)$ is a function of $e(t)$. By substituting Eqs. (5) and (8) into Eq. (11), the error dynamics are shown as,

$$\dot{e}(t) = (A - LC)e(t) + B(u_{nom}(t) - u(t)) - \Delta B_f u(t) \tag{14}$$

Equation (14) shows that there are several instances where $\dot{e}(t)$ can become nonzero. The first term, $(A - LC)e(t)$, will decay to zero exponentially for a stabilizing L. The gain L can always be chosen to stabilize $A - LC$ if (A, C) is observable (Van de Vegte, 1994). The second term in Eq. (14) will be nonzero when $u_{nom}(t)$ is not equal to $u(t)$. There are two expected sources of error between $u_{nom}(t)$ and $u(t)$. The first is modeling errors between the motor model and the actual motor. The second is an actuator fault, where $u(t)$ is not behaving as expected. If the motor model accurately models behavior of the actual motor, only an actuator fault will cause a noticeable increase in the magnitude of that term, and thus a significant increase in $\dot{e}(t)$. The third term will be nonzero when a broken line or criss-crossed lines fault has occurred.

4.2. Residual Evaluation

A threshold for detection is the main tool used in this work for residual evaluation. This threshold is chosen such that there is a high probability of a fault when the residual is above the threshold and a low probability of a fault when the residual signal is below the threshold. Statistical methods are used for threshold determination. To improve detection statistics, the residual signal at each time step is smoothed over the previous 25 seconds of flight using a moving average. The time period over which smoothing occurs can be varied according to design needs. A longer period better emphasizes the trend of the signal while filtering out noise, but will cause a lag between the occurrence of a fault and the response of the signal. This parameter was tuned numerically to achieve desired detection characteristics.

When designing a threshold, the goal is to minimize two quantities: probability of missed detection $P(MD)$ and probability of false alarm $P(FA)$ (Rossi, Breger, Benson, Sargent, & Fesq, 2012) (Sturza, 1988). The probability of missed detection is the probability that a fault has occurred and no fault alarm is raised; the probability of false alarm is the probability that an alarm is raised when no fault has occurred. These quantities are predicted by collecting data from simulations of healthy flights and flights in which a fault has occurred. Cumulative density functions (CDFs) of data from simulated healthy flights and flights in which a fault occurs are useful in visualizing how a chosen threshold affects $P(FA)$ and $P(MD)$.

In order to collect the data used in the CDFs, a high-fidelity, nonlinear, 6 degree-of-freedom (DOF) simulation was used, similar to the model described in Ward et al. (2010). Each simulation is generated with different initial conditions. These conditions include not only three-dimensional position, orientation, and velocity, but environmental conditions such as wind profile and system irregularities. System irregularities are variations from parafoil-to-parafoil, turn bias and lift-to-drag ratio for example, that change the flight characteristics. The linearized lateral dynamics used for the observer initialize the state vector to zero, and do not account for wind or differences between each individual parafoil system.

To generate CDFs that will accurately show $P(FA)$ and $P(MD)$ for various thresholds, large data sets were collected that reflect the range of conditions a parafoil system experiences during a healthy flight as well as flights in which faults of varying type and severity have occurred. For the healthy data set, 1000 Monte Carlo simulations were performed that varied the following characteristics: initial position, initial altitude, initial velocity, initial attitude, payload weight, wind conditions, and turn bias (i.e., nonzero heading rate in the presence of zero control). No fault occurred during any of these simulations. The data of interest from each Monte Carlo run is the maximum value that the smoothed residual signal reaches during the simulation. This maximum bounds the smoothed residual signal expected from healthy flights.

Data collection from the fault cases was treated differently than data collection from healthy flight simulations. Again, 1000 Monte Carlo simulations were performed, but for these simulations random faults were inserted. The nature of each fault was chosen with uniform probability from the four fault cases discussed in Section 3. The time of the fault was chosen with uniform probability to be an integer value between 0 and 500 seconds. This choice ensured sufficient time for detection, because most simulated flights last longer than 600 seconds. The severity of the fault, if applicable, was randomized. For example, the broken line fault severity need only be randomized as a left or right line break, but a stuck motor fault occurs on either the left or right motor and has a particular value (e.g., 0.5 meters) at which the motor is stuck. Multiple faults were not considered.

The relevant data collected from these flights is the maximum value reached by the smoothed residual signal during the first 60 seconds after the fault occurs. In flight, there is a window of time after a fault occurs at which point recovery from the fault is either impossible or impractical. Thus, the detection method is only given a predetermined amount of time to raise an alarm. This time period is a design parameter that should be set based on mission requirements and recovery techniques in use. Using the collected data, CDFs were generated (Figure 6) and the performance of various thresholds were analyzed. The green circles shown on Figure 6 mark $P(FA)$ and $P(MD)$ on the CDF. $P(MD)$ is the intersection of the threshold line with the fault data curve. The intersection of the threshold line with the healthy curve is $1 - P(FA)$.

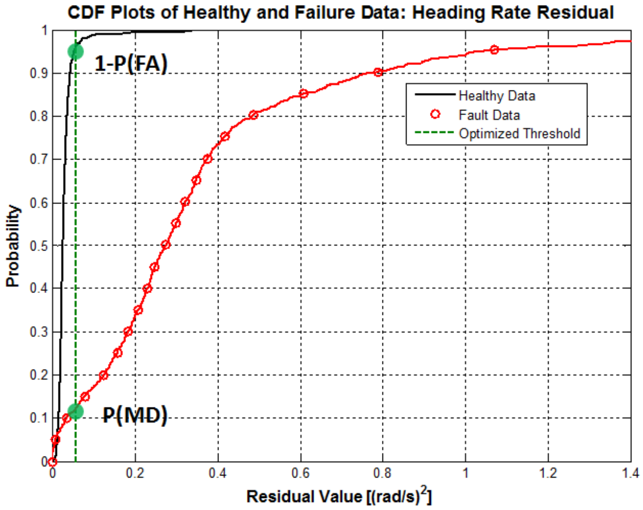


Figure 6. CDFs for heading rate residual.

The selection of an appropriate threshold given the data in Figure 6 depends on the emphasis placed on minimizing $P(FA)$ versus $P(MD)$. The following figure of merit (FOM) is used to penalize $P(FA)$ and $P(MD)$ as desired (Rossi, Benson, Sargent, & Breger, 2012):

$$FOM = 1 - \frac{c_1 P(FA) + c_2 P(MD)}{c_1 + c_2} \tag{15}$$

The weightings c_1 and c_2 can be varied according to design needs, where a higher weighting on either $P(FA)$ or $P(MD)$ indicates that it is more important to minimize that particular probability. Then, a threshold is determined that maximizes the chosen FOM. Table 1 shows thresholds, $P(FA)$, and $P(MD)$ for different weightings. The threshold shown on Figure 6, 0.0574 (rad/s)^2, as a vertical dotted line was chosen by placing an equal weighting on $P(FA)$ and $P(MD)$.

Table 1. Optimized thresholds for various FOMs.

c1	c2	Threshold $(\frac{\text{rad}}{\text{sec}})^2$	P(FA)	P(MD)	FOM
1.0	0.0	0.3995	0.000	0.733	1.000
0.8	0.2	0.0995	0.012	0.172	0.956
0.6	0.4	0.0607	0.032	0.127	0.927
0.5	0.5	0.0574	0.038	0.119	0.921
0.4	0.6	0.0574	0.038	0.119	0.913
0.2	0.8	0.0573	0.038	0.119	0.897
0.0	1.0	0.0017	0.999	0.017	0.983

The results presented are from simulations and not flight data. However, the same procedure can be applied to actual systems. The detection threshold can be adjusted as needed to achieve the desired performance given anticipated increases in process noise and navigation error.

4.3. Detection Results

Performance evaluation of the detection method consists of comparing predicted $P(MD)$ and $P(FA)$ (Table 1) with the corresponding probabilities resulting from the implementation of the detection method in simulation. The fault detection method presented in this paper was tested on 1000 Monte Carlo simulations of flights with randomized conditions. Each flight was chosen with equal probability to have no fault, a stuck motor fault, severe saturation, a broken line, or criss-crossed lines. The severity of each fault was randomized where applicable. Table 2 shows results from these simulations as well as the predicted values of $P(FA)$ and $P(MD)$ from Table 1.

Figure 7 plots $P(MD)$ versus $P(FA)$ for the data summarized in Table 2. The closer the data are to the origin, the better the performance (Rossi, Benson, et al., 2012). This plot also indicates an important point about fault detection: there is always a tradeoff between false alarm and missed detection (Rossi, 2012). These quantities are minimized according to design criteria by maximizing the figure of merit in Eq. (15).

Table 2. Comparison of $P(MD)$ and $P(FA)$ between predicted and simulated results for detection.

Threshold $(\frac{\text{rad}}{\text{sec}})^2$	P(FA) Predicted	P(FA) Sim	P(MD) Predicted	P(MD) Sim
0.3995	0.000	0.000	0.733	0.594
0.0995	0.012	0.005	0.172	0.201
0.0607	0.032	0.051	0.127	0.154
0.0574	0.038	0.058	0.119	0.149
0.0573	0.038	0.058	0.119	0.149
0.0017	0.999	0.796	0.017	0.064

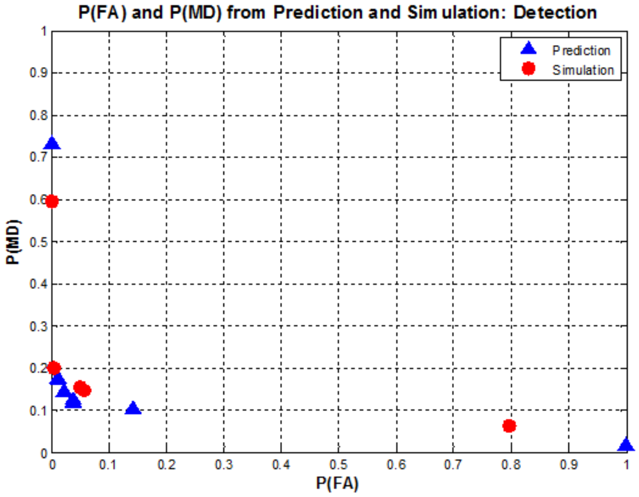


Figure 7. Comparison of predicted and simulated false alarm and missed detection rates.

4.4. Detection Example

Figure 8 shows an example of the detection process on a simulated flight. The fault, a stuck right motor at 0.863 meters, occurs 126 seconds into the flight. The top subplot of Figure 8 indicates nominal tracking of motor commands up until the time of the fault. Once this occurs, there is a large difference between the command and the response in the right motor. This difference is reflected in the smoothed residual signal, which is shown in the bottom subplot. The threshold chosen for detection during this flight is 0.0574 (rad/s)^2, which is the optimized value when equal weighting is placed on $P(FA)$ and $P(MD)$ (Table 1). The smoothed residual is below the threshold, but not zero, before the fault. However, once the fault occurs, the difference between $u(t)$ and

$u_{nom}(t)$ is large enough to cause the smoothed residual signal to cross the chosen threshold. When the residual crosses the threshold at 134 seconds a detection alarm is raised. After the alarm is raised, the isolation procedure begins.

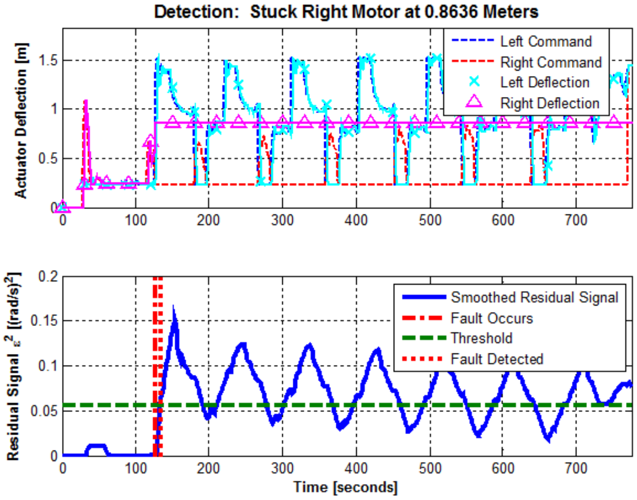


Figure 8. Detection example for stuck right motor fault.

5. Isolation

Once the detection algorithm has determined that a fault has occurred, the isolation process attempts to determine which particular fault is present. The data from the AGU motors can be used to determine if the fault is actuator-related. The first phase in the isolation process considers a motor residual signal, similarly to how the heading rate residual was evaluated during detection in Section 4. Each signal, one for each motor, should be small when the motor is behaving well and large when an actuator fault has occurred. These signals are evaluated using hypothesis testing. Appropriate thresholds must be selected such that $P(FA)$ and $P(MD)$ are minimized.

If the results of the first phase indicate that the fault is not actuator-related, phase two of isolation uses a bank of fault-specific observers (Willsky, 1976) to attempt to declare that a particular non-actuator fault has occurred. Some faults, such as a stuck motor and severe saturation, are difficult to model a priori because each of these faults is parameter-dependent (e.g. a stuck left motor at 0.5 meters). Other faults, such as criss-crossed lines and broken line, can be modeled in a straightforward manner as the effects of the faults are well-known. When a residual signal generated from a fault-specific observer is small, it is likely that the system has experienced the fault associated with that particular observer. Successful isolation will result in the declaration of a left motor fault, right motor fault, broken left line, broken right line, or a criss-crossed lines fault.

5.1. Motor Residual

Each motor on the AGU is equipped with an encoder that measures the corresponding control line deflection. This measurement is used as the performance metric for the motors on the AGU. Outputs from a nominal motor model are required to construct the motor residual signals used for isolation. The motor model used in the heading rate observer (Figure 5) is the nominal motor. The actual motor will output the motor deflection as measured by the encoders, and the nominal motor will output an expected motor deflection. The absolute value of the difference between these two signals is the motor residual signal,

$$r_{m,R}(t) = |\varepsilon_{m,R}(t)| = |\delta_{R,nom}(t) - \delta_R(t)| \qquad (16)$$

$$r_{m,L}(t) = |\varepsilon_{m,L}(t)| = |\delta_{L,nom}(t) - \delta_L(t)| \qquad (17)$$

where $r_{m,R}(t), r_{m,L}(t) \in \mathbb{R}$ are the right and left motor residual signals, respectively. The residual generation process is shown in Figure 9.

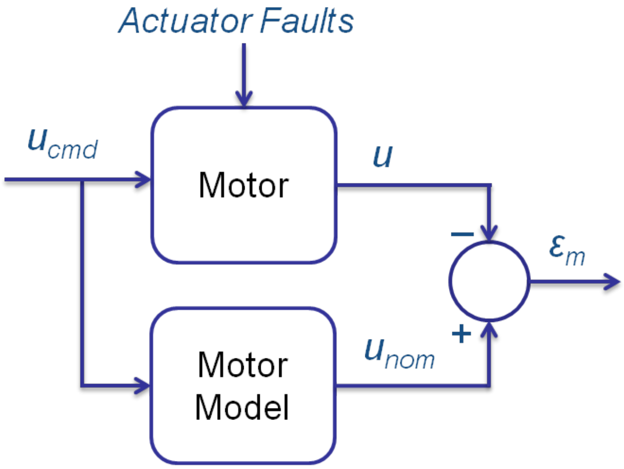


Figure 9. Motor residual block diagram.

Each motor residual is smoothed to emphasize the trend of the signal, similarly to the way the heading rate residual was smoothed in Section 4. Instead of the 25 second smoothing period used for the heading rate residual, each motor residual is smoothed over the previous 10 seconds at each time step. The behavior of the motor residuals is not as erratic as that of the heading rate residual and a longer smoothing period is not necessary. Once each motor residual has been generated and smoothed, it is evaluated by choosing a threshold. If a smoothed motor residual signal crosses above the chosen threshold within 5 seconds after detection occurs, an actuator fault in that particular motor is declared. This 5 second window is sufficient to detect most motor faults, as shown in the results in Section 6.1. This short amount of time also helps to minimize total isolation time if the fault is not actuator-related and fault-specific observers must be used.

Figure 10 shows CDF plots of healthy data and data from flights in which an actuator fault has occurred. The healthy data represents the maximum value of the smoothed motor residual observed during the entire flight. All of this data is for a left motor fault. Results for right motor faults are almost identical and are not presented in this work. The fault data, which randomizes the type, time, and severity of each actuator fault, represents the maximum value of the smoothed motor residual observed during the first 5 seconds after the fault is declared.

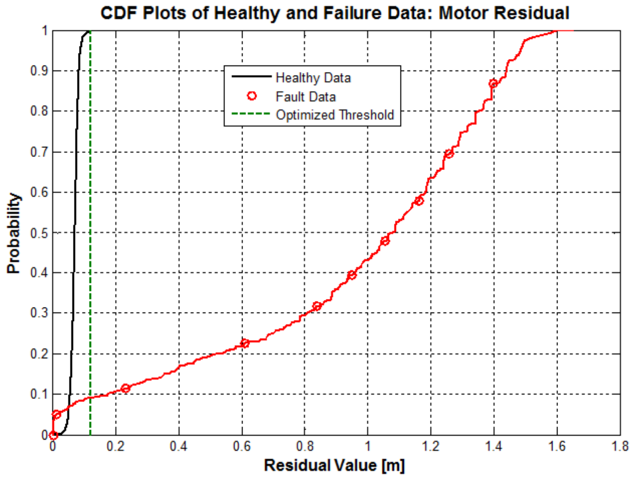


Figure 10. CDFs for motor residual.

As with the heading rate residual in Section 4, weightings on the FOM in Eq. (15) must be chosen for $P(FA)$ and $P(MD)$ to determine an optimized threshold. For equal weightings on each probability, the optimized threshold is 0.1237 meters. This threshold is shown as a vertical dotted line in Figure 10. Table 3 shows optimized thresholds for several different FOMs.

Table 3. Optimized thresholds for various FOMs (motor residual).

c1	c2	Threshold (m)	P(FA)	P(MD)	FOM
1.0	0.0	0.1363	0.000	0.094	1.000
0.8	0.2	0.1363	0.000	0.094	0.981
0.6	0.4	0.1238	0.000	0.093	0.962
0.5	0.5	0.1237	0.001	0.093	0.953
0.4	0.6	0.1198	0.002	0.092	0.944
0.2	0.8	0.1002	0.015	0.086	0.928
0.0	1.0	0.0074	0.999	0.040	0.960

5.2. Fault-Specific Observers

If the evaluation of the motor residual signals from Section 5.1 indicates nominal performance, phase two of the isolation procedure begins, which uses fault-specific observers. As with the heading rate residual, the residuals for fault-specific observers are constructed by differencing the EKF-estimated heading rate from the actual system and the expected heading rate output from the system observer. However, the expected heading rate signal comes from an observer that uses the dynamics of a system with a specific fault implemented. A block diagram of the residual generation process is shown in Figure 11. As with the heading rate and motor residuals, each fault-specific observer residual is smoothed. The residuals are smoothed at each time step over the previous 5 seconds of flight. This is to ensure that the residuals respond quickly to faults in order to minimize total isolation time. As with the heading rate residual, this smoothing parameter was tuned to achieve desired results from the fault-specific observers.

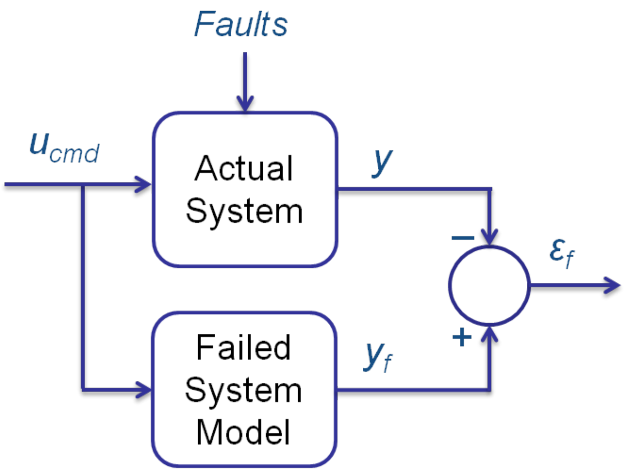


Figure 11. Fault-specific observer block diagram.

Fault-specific observers are constructed for the broken left line, broken right line, and criss-crossed lines faults. First, consider the broken left line fault. A simple modification to the nominal observer in Eq. (8) is required to construct this observer. The broken left line fault-specific observer is,

$$\dot{\hat{x}}_{bl}(t) = A\hat{x}_{bl}(t) + b_1\delta_R(t) + L(y(t) - \hat{y}_{bl}(t)) \qquad (18)$$

$$\hat{y}_{bl}(t) = C\hat{x}_{bl}(t) \qquad (19)$$

The observer for a broken right line is similar,

$$\dot{\hat{x}}_{br}(t) = A\hat{x}_{br}(t) - b_1\delta_L(t) + L(y(t) - \hat{y}_{br}(t)) \qquad (20)$$

$$\hat{y}_{br}(t) = C\hat{x}_{br}(t) \tag{21}$$

where $\hat{x}_{bl}(t), \hat{x}_{br}(t) \in \mathbb{R}^4$ are state estimates for the broken left and right line observers, respectively, and $\hat{y}_{bl}(t), \hat{y}_{br}(t) \in \mathbb{R}$ are the outputs from each observer. The only modification needed to construct these observers is to remove either δ_L or δ_R from u_{nom} in Eq. (8), depending upon which line has broken. The corresponding residual signals for these fault-specific observers are,

$$r_{bl}(t) = (\hat{y}_{bl}(t) - y(t))^2 \tag{22}$$

$$r_{br}(t) = (\hat{y}_{br}(t) - y(t))^2 \tag{23}$$

where $r_{bl}(t), r_{br}(t) \in \mathbb{R}$ and $y(t)$ is the EKF-estimated heading rate for the system.

The observer for the criss-crossed lines fault is,

$$\dot{\hat{x}}_{cc}(t) = A\hat{x}_{cc}(t) - Bu_{nom}(t) + L(y(t) - \hat{y}_{cc}(t)) \tag{24}$$

$$\hat{y}_{cc}(t) = C\hat{x}_{cc}(t) \tag{25}$$

where $\hat{x}_{cc}(t) \in \mathbb{R}^4$ is the state estimate for the criss-crossed line observer, and $\hat{y}_{cc}(t) \in \mathbb{R}$ is the output from the observer. The only modification to Eq. (8) needed to create the observer is to reverse the effect of the input. The residual signal for the criss-crossed lines observer is

$$r_{cc}(t) = (\hat{y}_{cc}(t) - y(t))^2 \tag{26}$$

where $r_{cc}(t) \in \mathbb{R}$.

These residuals represent how well each observer models the current condition of the system, and should be small only when the fault that is modeled in the observer is present in the actual system. Thresholds need to be designed such that when the smoothed residual is above the threshold, it is likely that the corresponding fault is not present. These thresholds are used in the second isolation phase.

The second phase of isolation begins by assuming that all three faults considered in this phase (i.e., broken left line, broken right line, criss-crossed lines) are present in the system. At each time step, the smoothed residual signals associated with these three faults are evaluated. If a certain signal has crossed above its corresponding threshold, the fault associated with the signal is eliminated from consideration. Once two of the three signals have crossed their corresponding thresholds, the fault associated with the residual that remained below its threshold is declared and isolation is successful. The FDI method is given 90 seconds after the fault occurs to both detect and isolate the fault. As described in Section 4.2, this time limit is imposed to force a successful isolation to occur in a reasonable amount of time such that there is sufficient time remaining in the flight for recovery. This parameter can be changed according to mission requirements.

If, at the end of the 90 second isolation period, one or fewer smoothed fault-specific observer residuals have crossed their corresponding thresholds, a final check is performed to attempt to isolate the correct fault. The relative size of each smoothed residual signal that has not already crossed its threshold is compared and the fault associated with the smallest signal is declared. The relative size R of each smoothed residual signal is given by

$$R = \max(r_s(t))/T \tag{27}$$

where T is the chosen threshold for the fault-specific observer, and the maximum of the corresponding smoothed residual signal $r_s(t) \in \mathbb{R}$ is computed over the 90 second isolation period.

There are several instances where isolation can fail. The first case occurs when all three of the smoothed fault-specific observer residual signals cross above their corresponding thresholds. This results in inconclusive isolation and no further action is taken. The other category of failure during isolation is called false isolation. This occurs when a fault is declared that is different from the actual fault that has occurred. This type of failure can occur during the motor observer phase if a motor fault is incorrectly declared, or during the fault-specific observer phase if the incorrect fault is declared. Several probabilities are used to assess the effectiveness of the isolation method. The probability that the correct fault is isolated given successful detection is $P(ISO)$. The probabilities that, once detection occurs, the isolation phase is inconclusive or declares an incorrect fault are given by $P(INC)$ and $P(FI)$, respectively.

5.2.1. Broken Line

The results for both the broken left line and broken right line faults are almost identical, so only the results from simulations of a broken left line are presented in this paper. In order to determine an appropriate threshold for the smoothed broken left line residual signal, the behavior of the signal was analyzed under three conditions: a broken left line is present, a broken right line is present, and a criss-crossed lines fault is present. CDFs showing the maximum value of the smoothed residual observed during the first 90 seconds after the occurrence of a fault for 1000 Monte Carlo simulations of all three fault cases are shown in Figure 12. A threshold for the broken line fault must be chosen so that the smoothed residual signal remains under the threshold for a large percentage of flights in which a broken left line fault occurs and crosses above the threshold for a large percentage of flights in which a broken right line or criss-crossed lines fault occurs. The thresholds for the broken line and criss-crossed lines faults were chosen to provide a high rate of successful isolation while minimizing the time at which isolation completes. The threshold chosen for the broken line fault is 0.07 (rad/s)^2 and is shown as a vertical dotted line on Figure 12.

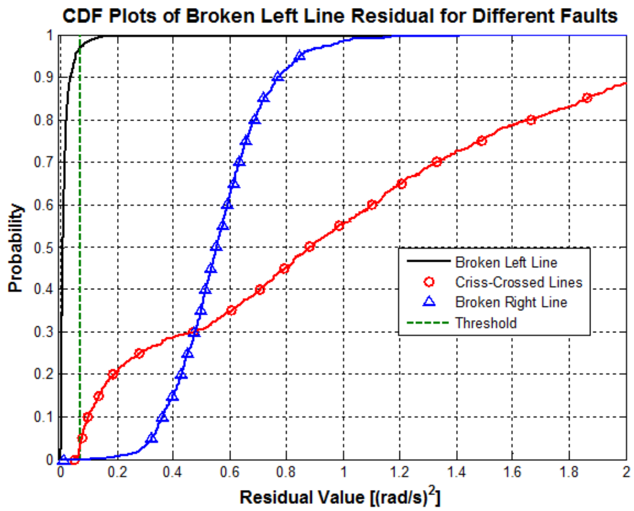


Figure 12. CDFs for broken left line residual.

The data from the CDFs in Figure 12 indicate that, given the chosen threshold, the probability that a broken left line is present but the smoothed residual signal associated with the broken left line still crosses the threshold is 2.9%. The probability that a broken right line is present but the smoothed broken left line residual remains under the threshold for the 90 second isolation period is 0.1%. The probability that a criss-crossed lines fault is present but the smoothed broken left line residual remains under the threshold for the duration of the isolation period is 2.9%. These probabilities can be similarly obtained for the broken right line case.

5.2.2. Criss-Crossed Lines

Figure 13 shows CDFs of data collected from the smoothed criss-crossed lines residual signal in the presence of a criss-crossed lines fault as well as broken left and right line faults. The CDFs show the maximum value reached during the first 90 seconds after the fault occurs for 1000 Monte Carlo simulations of the three fault cases. The threshold chosen for the smoothed criss-crossed lines residual signal is 0.13 (rad/s)^2 and is shown as a vertical dotted line on Figure 13.

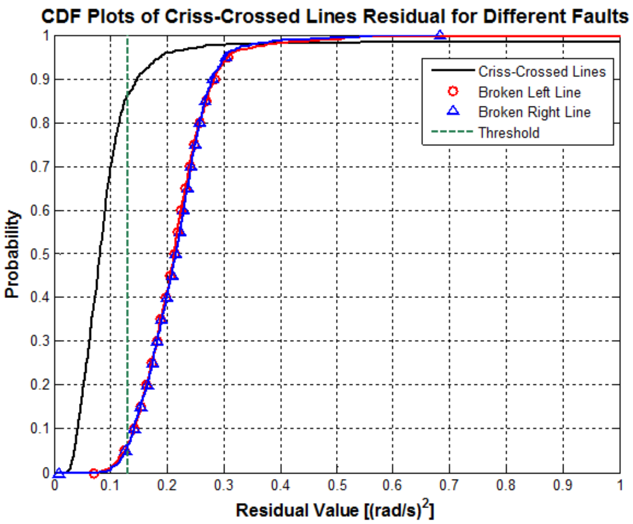


Figure 13. CDFs for criss-crossed lines residual.

The data from the CDFs in Figure 13 indicate that, given the chosen threshold, the probability that a criss-crossed lines fault is present but the residual signal still crosses over the threshold is 13.8%. The probability that a broken right or left line is present but the residual remains under the threshold for the isolation period is 5.9%. Isolation results for both the broken line and criss-crossed lines cases are presented in Section 6.1.

6. Full FDI Implementation and Results

Sections 4 and 5 outline methods for both detection and isolation of faults for the parafoil and payload system. Figure 14 shows an overview of the integrated process.

The first step is fault detection. The heading rate signal is monitored throughout the entire flight. If, at any point, the smoothed residual signal rises above a predetermined threshold, a fault is declared. Once the alarm is raised, the algorithm progresses to the isolation method.

Isolation begins by evaluating the motor residual signal at the time of fault detection. If the smoothed residual from either the left or right motor is above the predetermined threshold, a fault in the corresponding motor is declared. With the declaration of an actuator fault, the FDI process ends.

Alternately, if the motor residual does not cross the threshold within 5 seconds after detection, the isolation algorithm progresses to the evaluation of residuals from a bank of fault-specific observers. If the smoothed residual signal associated with one of these observers is small, the algorithm declares that the fault corresponding to that particular observer has occurred. If none of the fault-specific observer residuals indicate that the system is exhibiting the characteristics of any known fault, then FDI has failed. In this case, it is likely that a non-actuator fault that does not have an observer associated with it has occurred. If more than one fault-specific observer models the actual system well, FDI is unsuccessful. Multiple faults occurring during the same flight are not considered in this work.

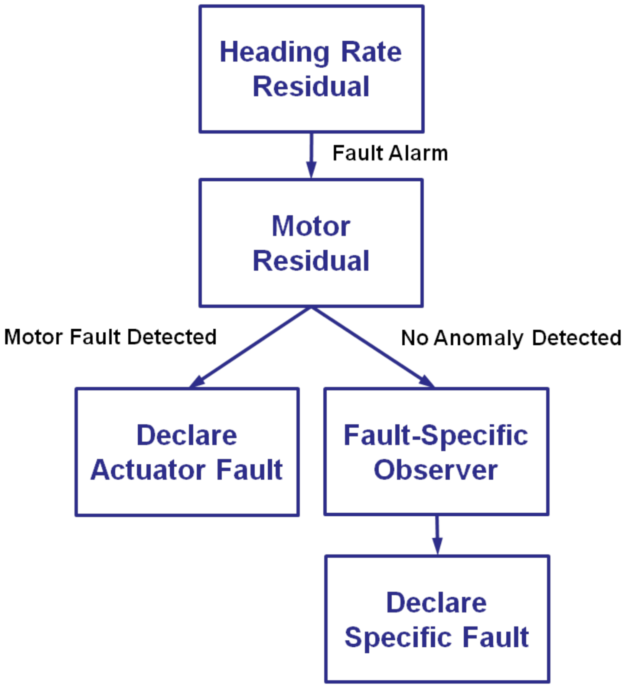


Figure 14. FDI Procedure.

6.1. Results

This section presents results of the full FDI implementation for three different fault categories: motor fault (i.e. stuck motor and unexpected saturation), broken line, and criss-crossed lines. This paper contains no comparison with parafoil FDIR work in the literature, as no such work was identified at the time of publication. Using the detection threshold for an equal weighting on $P(FA)$ and $P(MD)$ from Table 1, the motor residual threshold for an equal weighting on $P(FA)$ and $P(MD)$ from Table 3, the broken line fault-specific observer threshold chosen in Section 5.2.1, and the criss-crossed lines fault-specific observer threshold chosen in Section 5.2.2, the performance of the full FDI method was tested on 1000 Monte Carlo simulations in which a fault in one of the three categories being considered occurred at a random integer time during the first 500 seconds of flight. The performance of the FDI method is evaluated in terms of the probability of detecting the fault, $P(DET)$, as well as $P(ISO)$, $P(INC)$, and $P(FI)$. Table 4 shows FDI results for all three fault categories. The results compiled for motor faults and the broken line fault are compiled for the left motor and control line, respectively. Results from a right motor fault or a broken right line are almost identical to those obtained from the left side and are not presented in this work.

Table 4 indicates that a large portion of missed detections occur when attempting to detect motor faults. This is due in part to severe saturation. It is possible that, during the 60 second detection period, no motor command is given that is greater than the severe saturation limit. In this case, the parafoil behaves as if no fault has occurred. Cases like this, and other scenarios in which the motor command and motor response are very similar, generally result in unsuccessful detection.

Table 4. FDI results for three fault categories.

Fault	P(DET)	P(ISO)	P(INC)	P(FI)
Left Motor Fault	0.732	0.969	0.003	0.029
Broken Left Line	0.989	0.990	0.004	0.006
Criss-Crossed Lines	0.997	0.953	0.019	0.028

Despite the detection issues for the motor fault, the rate of isolation given a successful fault detection is high. The broken line and criss-crossed line faults have high isolation rates as well, validating the use of both the motor observer and fault-specific observers.

6.2. Examples

The first example of the full FDI implementation is for a stuck left motor at 1.295 meters. Figure 15 shows the heading rate residual and motor residual for a stuck left motor fault.

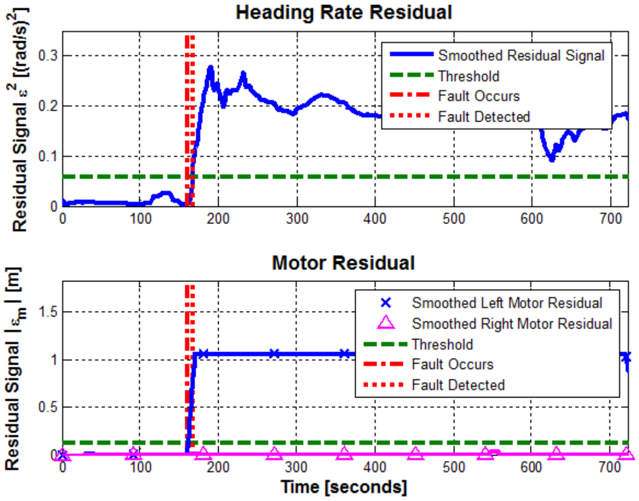


Figure 15. Stuck left motor fault: Heading rate and motor residuals.

The top subplot in Figure 15 shows the heading rate residual for the stuck motor case. A fault alarm is raised at 167 seconds, 7 seconds after the fault occurs. After this alarm is raised, the motor residual, shown in the bottom subplot, is evaluated to determine if the fault is actuator-related. The plot indicates that, at the time of detection, the left motor is showing off-nominal behavior while the right motor is behaving well. A left motor fault is declared and FDI is complete.

The second example demonstrates successful FDI for a broken left line fault. The heading rate and motor residuals are shown in Figure 16, and the fault-specific observer residuals are shown in Figure 17 and Figure 18. The fault occurs at 193 seconds and a detection alarm is raised 15 seconds later. The motor residual is examined next. This residual indicates that both motors are behaving nominally, meaning that the fault is not related to the actuators. The next step is to consider the bank of fault-specific observers. Successful isolation occurs at 213 seconds, when both the broken right line and criss-crossed line residuals have crossed their corresponding thresholds. The broken left line residual remains under its threshold for the entire flight. The algorithm reports a broken left line 20 seconds after the fault occurs, successfully completing the FDI procedure.

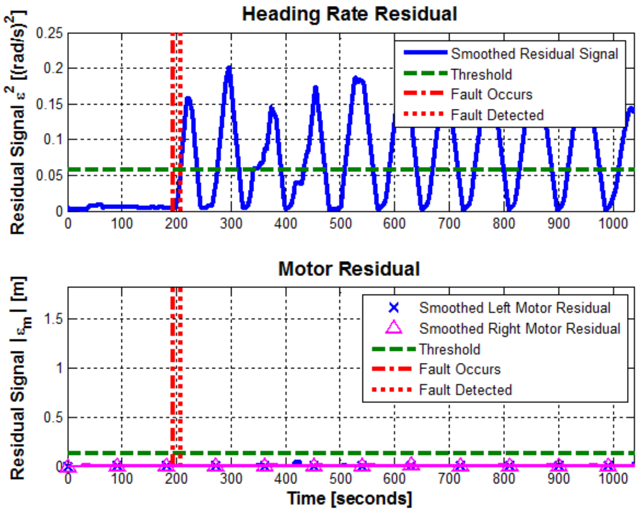


Figure 16. Broken left line fault: Heading rate and motor residuals.

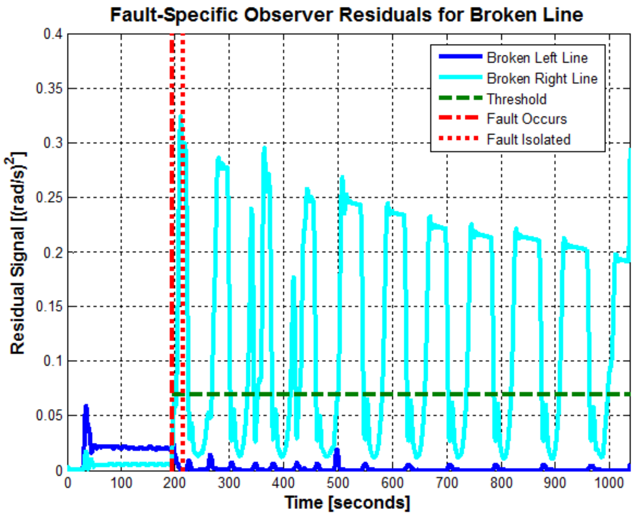


Figure 17. Broken left line fault: Broken line fault-specific observer residuals.

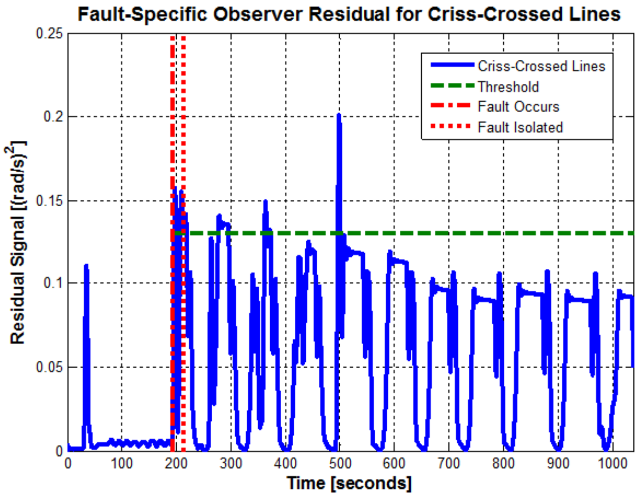


Figure 18. Broken left line fault: Criss-crossed line fault-specific observer residual.

7. Conclusion

This paper presented a fault detection and isolation method for an autonomous parafoil system. The detection method evaluates a residual signal generated from navigated heading rate and an observer based on a nominal system model. The isolation method uses separate residual signals generated from motor telemetry to determine whether a given fault is actuator-related. Other faults are isolated using fault-specific observers. We combined these methods and evaluated the approach against four common faults using high-fidelity Monte Carlo simulations. The results of these simulations showed that choosing an appropriate detection threshold allows for a high rate of detection with minimal false alarms. Choosing a threshold of 0.0995 (rad/s)^2 yielded an 82.8% success rate for detecting faults with a 0.5% rate of false alarms. For a broken line fault, successful detection occurs 98.9% of the time. Given successful detection, successful isolation of a broken line fault occurs at a rate of 99.0%. This work introduced a method for parafoil FDI that can detect and isolate common faults in an effective, timely, and predictable manner.

References

Barrows, T. (2009). Multibody parafoil model. In *20th AIAA aerodynamic decelerator systems technology conference and seminar.* Seattle, WA. doi: 10.2514/6.2009-2945

Beard, R. V. (1971). *Failure accommodation in linear systems through self-reorganization.* Doctoral dissertation, Massachusetts Institute of Technology, Cambridge, MA.

Bergeron, K., Fejzic, A., & Tavan, S. (2011). Accuglide 100: Precision airdrop guidance and control via glide slope control. In *21th AIAA aerodynamic decelerator systems technology conference and seminar.* Dublin, Ireland.

Carter, D., George, S., Hattis, P. D., Singh, L., & Tavan, S. (2005). Autonomous guidance, navigation, and control of large parafoils. In *18th AIAA aerodynamic decelerator systems technology conference and seminar.* Munich, Germany. doi: 10.2514/6.2005-1643

Crimi, P. (1990). Lateral stability of gliding parachutes. *Journal of Guidance, Control, and Dynamics*, *13*(6), 1060-1063. doi: 10.2514/3.20579

Douglas, R. K., & Speyer, J. L. (1995). Robust fault detection filter design. In *Proceedings of the american control conference* (Vol. 1, p. 91-96). Seattle, WA. doi: 10.1109/ACC.1995.529214

Figueroa, F., Schmalzel, J., Morris, J., Turowski, M., & Franzl, R. (2010). Integrated system health management: Pilot operational implementation in a rocket engine test stand. In *AIAA Infotech@Aerospace.* Atlanta, GA. doi: 10.2514/6.2010-3454

Figueroa, F., Schmalzel, J., Walker, M., Venkatesh, M., Kapadia, R., Morris, J., … Smith, H. (2009). Integrated system health management: Foundational concepts, approach, and implementation. In *AIAA Infotech@Aerospace conference.* Seattle, WA. doi: 10.2514/6.2009-1915

Frank, P. M. (1994). Enhancement of robustness in observer-based fault detection. *International Journal of Control*, *59*(4), 955-981. doi: 10.1080/00207179408923112

Hattis, P., Campbell, D. P., Carter, D. W., McConley, M., & Tavan, S. (2006). Providing means for precision airdrop delivery from high altitude. In *AIAA guidance, navigation, and control conference and exhibit.* Keystone, CO.

Hattis, P., & Tavan, S. (2007). Precision airdrop. *Aerospace America*, *April*, 38-42.

Hwang, I., Kim, S., Kim, Y., & Seah, C. E. (2010). A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, *18*(3), 636-653. doi: 10.1109/TCST.2009.2026285

Isermann, R., & Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, *5*(5), 709-719. doi: 10.1016/S0967-0661(97)00053-1

Jones, H. L. (1973). *Failure detection in linear systems*. Doctoral dissertation, Massachusetts Institute of Technology, Cambridge, MA.

Patton, R. J., & Chen, J. (2000). On eigenstructure assignment for robust fault diagnosis. *International Journal of Robust and Nonlinear Control*, *10*(14), 1193-1208. doi: 10.1002/1099-1239(20001215)10:14<1193::AID-RNC523>3.0.CO;2-R

Rossi, C. (2012). *Vehicle health monitoring using stochastic constraint suspension*. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

Rossi, C., Benson, D., Sargent, R., & Breger, L. S. (2012). Model-based design for vehicle health monitoring. In *Infotech@Aerospace.* Garden Grove, CA. doi: 10.2514/6.2012-2577

Rossi, C., Breger, L., Benson, D., Sargent, R., & Fesq, L. (2012). Vehicle health monitoring using stochastic constraint suspension. In *AIAA guidance, navigation, and control conference.* Minneapolis, MN.

Sargent, R., Mitchell, I., Breger, L., Benson, D., Bessette, C., Zanetti, R., & Groszkiewicz, J. E. (2011). A fault management strategy for autonomous rendezvous and capture with the ISS. In *Infotech@Aerospace.* St. Louis, MO. doi: 10.2514/6.2011-1497

Slegers, N., & Costello, M. (2004). Model predictive control of a parafoil and payload system. In *AIAA atmosphere flight mechanics conference and exhibit.* Providence, RI.

Sturza, M. A. (1988). Navigation system integrity monitoring using redundant measurements. *Journal of the Institute of Navigation*, *35*(4), 69-87.

Tavan, S. (2006). Status and context of high altitude precision airdrop delivery systems. In *AIAA guidance, navigation, and control conference and exhibit.* Keystone, CO.

Van de Vegte, J. (1994). *Feedback control systems*. Prentice Hall.

Ward, M., Montalvo, C., & Costello, M. (2010). Performance characteristics of an autonomous airdrop system in realistic wind environments. In *AIAA atmosphere flight mechanics conference.* Toronto, ON, Canada.

Willsky, A. S. (1976). A survey of design methods for failure detection in dynamic systems. *Automatica*, *12*, 601-611.

Ensemble classifiers for drift detection and monitoring in dynamical environments

Imen Khamassi[1], Moamar Sayed-Mouchaweh[2], Moez Hammami[1] and Khaled Ghédira[1]

[1]*University of Tunisia, Search Laboratory SOIE, Higher Institute of Management of Tunis*
imenkhamassi@yahoo.fr
moezhammami@gmail.com
khaled.ghedira@isg.rnu.tn

[2]*Univ Lille Nord de France, F-59000 Lille, France, Mines-Douai, IA, F-59500 Douai, France*
moamar.sayed-mouchaweh@mines-douai.fr

ABSTRACT

Detecting and monitoring changes during the learning process are important areas of research in many industrial applications. The challenging issue is how to diagnose and analyze these changes so that the accuracy of the learning model can be preserved. Recently, ensemble classifiers have achieved good results when dealing with concept drifts. This paper presents two ensembles learning algorithms BagEDIST and BoostEDIST, which respectively combine the Online Bagging and the Online Boosting with the drift detection method EDIST. EDIST is a new drift detection method which monitors the distance between two consecutive errors of classification. The idea behind this combination is to develop an ensemble learning algorithm which explicitly handles concept drifts by providing useful descriptions about location, speed and severity of drifts. Moreover, this paper presents a new drift diversity measure in order to study the diversity of base classifiers and see how they cope with concept drifts. From various experiments, this new measure has provided a clearer vision about the ensemble's behavior when dealing with concept drifts .

1. INTRODUCTION

Recently, research in machine learning has shown its usefulness for automatic monitoring and diagnostics in industrial applications, especially when data is continuously generated and it is unpractical to store them all. Another issue is caused by the high speed of arrival of these data which require real time treatments and high computational efforts. Hence, the learning model must be able to classify this huge amount of data when environments are non-stationary; and the main challenge occurs when the underlying distribution that generates the data streams changes over time; which is known as "concept drift". Formally, the term "concept" refers to the distribution of the joint probability $P(X, w)$ in a certain point of time, where X represents the input attributes and w represents the class labels. A *concept drift* is a real or virtual change in this distribution. As stated by Tsymbal (2004), the *real concept drift* affects the posterior probability $P(w = w_i/X)$ which means that the target concept of the same values of attributes changes; we note $w_i \in W_c$ the set of c different class labels and $1 \leq i \leq c$.The *virtual concept drift* affects the class-conditional probability $\mathrm{P}(X/w = w_i)$ which means that the distribution of the underlying data within the same class changes. It is worth underlining that a drift can also affects the prior probability $P(w = w_i)$ of a particular class; this is known by "concept evolution". This form of drift can be due to *merging concepts* which refer to the emergence of novel classes, or to *crossing concepts* which refer to the fusion of existing classes (Masud et al., 2011).

Three steps are required to handle a concept drift:

- Monitoring step: When data are considered as non-stationary, methods with triggered mechanisms, namely informed methods (Ikonomovska et al., 2009), are used in order to provide descriptions about location, width and severity of the change. These methods can monitor the performance indicators of a learner, the estimators of data distributions or the learner's structure and parameters.

- Updating step: During this stage, the updating strategies differ according to the methods used to handle concept drifts. The blind methods implicitly adapt the learner to the current concept at regular intervals without any drift detection (Kolter and Maloof, 2007). Whereas, the informed methods can either relearn the model from scratch,

Khamassi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

or update it using recent data when a change is detected. So the question is: how much data to remember or to forget? And what is the optimal size of the data-window in order to contain the most recent and significant data? In many studies (Sobhani and Beigy, 2011), the window size was a priori fixed. However, if the size is too small, we may not have enough data to train the leaner which risks being over fitted; and if the size is too large, the learner may keep outdated concepts which risks to reduce its accuracy. The fixed size window can work well if the width and severity of the change are known before or if we have rigorous instructions provided by an expert, but this is rarely the case. Recent studies have opted to windows with dynamic size which is reduced whenever a drift is detected and enlarged otherwise (Gama et al., 2004; Baena-García et al., 2006). In other studies, Lazarescu et al. (2004) have used multiple windows with different sizes in order to progressively adapt and predict the change.

- Diagnostic step: This step aims at interpreting the detected changes in concepts or in the learner parameters and structure.This interpretation can then be used as a short-term prognosis about the future tendency of the current system situation. Notice that it is important to differentiate between noise in data and real changes (Sayed-Mouchaweh, 2010; Lughofer and Angelov, 2011). Ideally, a trade-off between robustness against noise and flexibility in tracking concepts drifts must be reached. But unfortunately these two requirements seem to be contradictory.

In the light of these challenges, we present two ensembles learning algorithms BagEDIST and BoostEDIST, which respectively combine the Online Bagging and the Online Boosting with the drift detection method EDIST. EDIST is a new drift detection method which monitors the distance (the number of instances) between two consecutive errors of classification, and tracks concept drifts through two adaptive data-windows $\mathrm{W_G}$ and $\mathrm{W_0}$. EDIST makes use of a statistical hypothesis test in order to compare $\mathrm{W_G}$ and $\mathrm{W_0}$ error distance distributions and checks whether the averages differ by more than the adjusted threshold ε .

The idea behind BagEDIST and BoostEDIST is to develop an ensemble learning algorithm which explicitly handles concept drifts by providing useful descriptions about location, speed and severity of drifts.

Moreover, this paper presents a new Drift Diversity Measure (D) in order to study the diversity of base classifiers and see how they cope with concept drifts. From various experiments, this new measure has provided a clearer vision about the ensemble's behavior when dealing with concept drifts.

The rest of the paper is organized as follows. In Section 2, we briefly discuss related work. In Section 3, we describe the drift detection method EDIST. In Section 4, BagEDIST and BoostEDIST are presented. In Section 5, the Drift Diversity Measure is decribed. In Sections 6 and 7, we explain the experimental setup, the results and the analysis of the different improvement stages of our method. Finally, conclusions and some ideas for future work are discussed in Section 8.

2. Discussion and related work

In this section, the blind and the informed methods are discussed. The blind methods handle concept drifts implicitly, while the informed methods detect drifts by monitoring some performance indicators of the model (See Figure 1).

2.1. Informed Methods

These methods are useful when we expect to provide descriptions about the occurrence, the severity and the width of the encountered drifts. They proceed by monitoring and diagnosing these changes so that the accuracy of the learner can be conserved whatever the nature of the encountered drifts. These methods can make use of single classifier based on instance selection or batch collection; or an ensemble of classifiers with change detector.

2.1.1. Single classifier with performance monitoring

This kind of methods monitors the indicators of performance of the model such as accuracy, precision and recall (Klinkenberg, 2001). These indicators are monitored constantly and compared to a confidence level or an adjusted threshold. Two well-known performance-based methods can be cited: Drift Detection Method (DDM) proposed by Gama et al. (2004); and Early Drift Detection Method (EDDM) proposed by Baena-García et al. (2006).

a. DDM: Drift Detection Method

The Drift Detection Method (DDM) monitors the number of errors produced by the learner and considers that the error rate follows the binomial distribution. In a sample of n instances, the distribution gives the probability of misclassification p_i with standard deviation $s_i = \sqrt{p_i(1-p_i)/i}$ for each instance i of the sampled sequence. According to probability approximately correct leaning model (PAC) used by Mitchell (1997), if the distribution of instances is stationary, the error rate decreases as the number of instances increases, thus a significant increase in the error rate during the training implies a change in the distribution. DDM stores p_{min} and s_{min} which correspond respectively to the minimum probability and the minimum standard deviation, and then it defines two levels as follows:

- The Warning level when $p_i + s_i \geq p_{min}$ $+2 \,.\, s_{min}$; beyond this level, instances are stored for a possible change of distribution.
- The Drift level when $p_i + s_i \geq p_{min}$ $+3.\, s_{min}$; beyond this level, the drift is confirmed and the learner is reset using instances stored since the Warning level. Note that p_{min} and s_{min} are reset too.

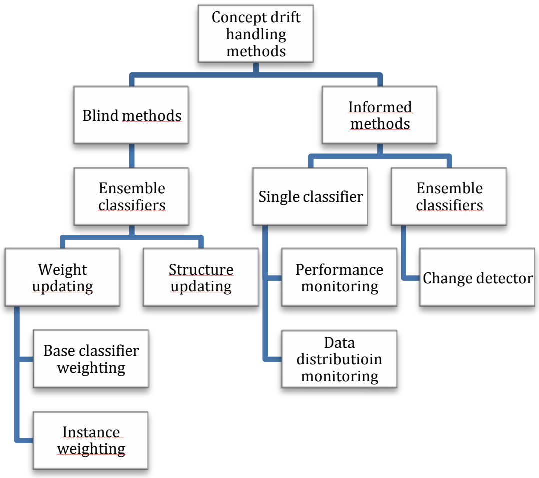


Figure 1. General scheme of classification methods handling concept drift

DDM has good ability to detect abrupt and global drifts which affect the whole dataset. However, it presents low adaptability to gradual and local drifts which slowly affect some parts of the dataset.

b. EDDM: Early Drift Detection Method

The idea behind the Early Drift Detection Method (EDDM) is to consider the distance between two consecutive errors of classification. Notice that the error distance is represented by the numbers of instances between two consecutive classification errors. This approach assumes that if the distribution of the instances is stationary, the learning model will improve its prediction and the error distance will increase as the number of instances increases. Thus, a significant decrease in the error distance implies a drift. EDDM calculates the average distance between two errors p'_i and its standard deviation s'_i for each instance i of the sampled sequence and compare them to p'_{max} and s'_{max} reached before, and then defines two levels as follows:

- The Warning level when $(p'_i + 2.s'_i)/(p'_{max} + 2.s'_{max}) < \alpha$; beyond this level, instances are stored for a possible change of distribution.
- The Drift level when $(p'_i + 2.s'_i)/(p'_{max} + 2.s'_{max}) < \beta$; beyond this level, the drift is confirmed and the learner is reset using instances stored since the Warning level. Note that p'_{max} and s'_{max} are reset too. α and β are respectively set to 0.95 and 0.9 after some experiments. The method considers the thresholds for detecting concept drifts when a minimum of 30 errors occur; then estimates the current error distance distribution and compares it with future ones.

EDDM is more suited to detect gradual and local drifts than DDM, but it presents sensitivity to the values of α and β in the sense that bigger values are suited to detect gradual drifts whereas smaller values are more suited to detect abrupt drifts. Hence, a trade-off between these values and different types of drifts is required to reach good results.

2.1.2. Single classifier with data distribution monitoring

This kind of methods detects changes by measuring differences between consecutive batches of data. Sobhani and Beigy (2011) present a method which process chunk by chunk and measures differences between two consecutive batches of data. The idea is to find nearest neighbor in previous batch of data for each instance in current batch, then compare their corresponding class labels. The authors use the *heom distance* to quantify the similarity between data batches and the drift alarm is launched when there is a significant increase of the degree of drift (DoF) value (for more details please refer to Sobhani and Beigy (2011)).

2.1.3. Ensemble classifiers with change detector

The ensemble classifiers with change detector explicitly handle concept drifts by providing useful description about the change. This new kind of technique is becoming an interesting area of research, because:

- it combines the flexibility of the ensemble classifier to cope with different types of drifts and,
- it can provide useful descriptions about location, width and severity of the drift.

ADWIN bagging method (Bifet et al. 2009) combines the online version of bagging for data streams developed by Oza and Russell (2001) with ADWIN which represents a change detector (Bifet et al. 2007). The idea behind ADWIN is to track the average of a stream of bits or real-valued numbers and keep a dynamic sliding window of recently seen instances, with the property that the window has the maximal length statistically consistent with the hypothesis “there has been no change in the average value inside the window".
These approached have achieved good accuracy when dealing with different kind of drifts, but they present two main issues; the first is how to maintain the diversity of the classifiers overall the learning process? and the second is how can we improve the run time and the memory consumption when resources are scarce?

2.2. Blind methods

The ensemble classifiers approaches are considered as blind methods when they are implicitly adapted to changes without any drift detection tool (Minku et al. 2010).

2.2.1. Structure-updating ensemble classifiers

In the dynamic ensemble classifiers the structure of the ensemble evolves to deal with the concept drifts. A possible strategy is “replace the loser”: the individual classifiers are re-evaluated and the worst one is replaced by a new one trained on recent data (Kuncheva, 2004).

The Dynamic online Ensemble Learning Algorithm (DELA) is characterized by dynamic and continuous structural update of the ensemble classifiers as soon as the global accuracy decreases. DELA makes use of distinct and incremental base learners which handle drift detection by their nature, and then updates the structural in two ways:

- The addition of classifiers is made when (i) the ensemble fails to predict the correct label or when (ii) it makes as many as errors as half of a window of length Q.
- The removal of classifiers is made when (i) the ensemble fails to predict the correct label or when (ii) the base learner makes an error on each step over the last window of length Q.

2.2.2. Weight-updating ensemble classifiers

In the weight-updating ensemble classifiers the structure of the ensemble is fixed, whereas the weights of the base classifiers or of the instances may evolve.

a. Weight-updating base classifiers

This kind of approach makes use of ensemble classifiers where each component classifier is evaluated and receives a weight reflecting its performance on the most recent batch of data.

In the Accuracy Weighted Ensemble (AWE), Kolter and Maloof (2007) propose to train a new offline learning classifier on each incoming data batch, then use that batch to evaluate all the existing classifiers in the ensemble. To evaluate each classifier, they propose to derive weights by estimating the error rate on the most recent data batch. For the first k data chunks, AWE takes a set of all available classifiers, but it selects only the k best components to form an ensemble. The predictions of components are aggregated by a weighted voting rule. However, the main problem of AWE is the tuning of the batch size of the most recent data used to evaluate all the existing classifiers.

The Accuracy Updated Ensemble (AUE) is another ensemble approach which proceed by updating each classifier features as new instances are available. Brzezinski and Stefanowski (2011) use online learning classifiers updated with recent instances, then adjust their weights. During the weighting process AUE preserves only the k best classifiers according to a simple weighting function. When no concept drift occurs, each base classifiers can be trained on more instances as if it was built on bigger batch of data; and should update its features to become more accurate. Good results have been achieved; however AUE has to ensure additional diversity of the ensemble components.

b. Instance-updating ensemble classifiers

This kind of approach makes use of ensemble classifiers where the weights of the instances are modified in order to preserve diversity when dealing with concept drift.

The Leveraging bagging approach (Bifet et al. 2010a) is an ensemble approach that makes use of a modified bagging approach by adding more randomization on the weights of the instances of the input stream in order to improve the accuracy of the ensemble classifier.

3. EDIST: ERROR DISTANCE BASED APPROACH FOR DRIFT DETECTION AND MONITORING

In this section, a new drift detection method EDIST is described. Then its performance is evaluated using datasets presenting concepts drifts of several width, severity and time change.

We consider the online learning framework where the instances arrive one at the time and we assume that the learning model is able to make a prediction as an instance is available. Once the prediction is made, the system can learn from instances and incorporate them to the learning model. Each instance is in the form of pairs $(\vec{x}_i, w_i)$ where $\vec{x}_i$ is the vector values of different attributes and w_i is the class label. The model prediction w_i' is correct when $w_i' = w_i$, false otherwise.

The idea behind EDIST draws its inspiration from the popular drift detection method EDDM which studies the distance between two consecutive errors of classification. Note that the distance is represented by the numbers of instances between two consecutive errors of classification. In EDIST, we track the concept drift through two data-windows. The first represents the global sliding window W_G which is adaptively adjusted by containing the recent read instances. The second W_0 represents the batch of current collected instances. Note that W_0 is constructed from a fixed number of consecutive errors of classification, thus it could contain a variable amount of instances at each step. In EDIST, we want to estimate the error distance distribution of W_G and W_0 and make a comparison between their error distance averages in order to check a difference.

In EDIST, we employ the same hypothesis used in EDDM and which assumes that if the distribution of the instances is stationary, the learning model will improve its prediction and the error distance will increase as the number of instances increases. Thus, a significant decrease in the error distance implies a change in the instances distribution and suggests that the learning model is not appropriate. Unlike EDDM which compares the current average of error distance and its standard deviation with the maximum average and standard deviation stored from previous instances, EDIST makes use of a statistical hypothesis test in order to compare W_G and W_0 error distance distributions and check whether the averages differ by more than the threshold ε. The novelty of our method is that there is no a priori definition of the threshold ε, in the sense that it does not require any a priori adjusting related to the expected width or severity of the change. ε is adaptively adjusting itself according to the statistical test used.

3.1. Statistical hypothesis test

Let $d_{1,} d_{2,} d_{3,} \dots d_{t,}$ ... be the sequence of the error distance values where each value d_t is available only at time t and independently generated from an error distance distribution D_t.

Let X_G and X_0 be the random variables of the two error distance distribution D_G and D_0 of respectively W_G and W_0, we assume that $X_G \sim N\left(\mu_G, \frac{\sigma_G}{\sqrt{N}}\right)$ *and* $X_0 \sim N\left(\mu_0, \frac{\sigma_0}{\sqrt{n}}\right)$ are normally distributed where N and n are the numbers of errors of classification respectively occurred in W_G and W_0. We pose $\mu_d = \mu_G - \mu_0$ and define:

- The null hypothesis (*H_0*): $\mu_d = 0$ which states that there is no change between the two distributions averages.
- The alternative hypothesis (H_1): $\mu_d > 0$ which states that there is a decrease in D_0's average, hence we detect a change.

3.2. Region of acceptance

We suppose that *H_0* is true, thus the random variable $X_d \sim N(\mu_d, \sigma_d)$ is normally distributed with $\mu_d = 0$ and $\sigma_d = \sqrt{\frac{\sigma_G}{\sqrt{N}} + \frac{\sigma_0}{\sqrt{n}}}$; and let α=0.05 be the test's probability of incorrectly rejecting *H_0*.

We want to calculate ε such that:

$$P(X_d \le \mu_d + \varepsilon) = 0.95 \qquad (1)$$

Let $T = \frac{X_d - \mu_d}{\sigma_d}$ be a random variable following the standard normal distribution $N(0, 1)$

$$P\left(T \le \frac{\varepsilon}{\sigma_d}\right) = 0.95 \qquad (2)$$

so, the cumulative distribution function of the normal distribution is defined as follows:

$$\varphi\left(\frac{\varepsilon}{\sigma_d}\right) = 0.95 \qquad (3)$$

and thus, according to the table of standard normal distribution, we can write:

$$\frac{\varepsilon}{\sigma_d} = t_{1-\alpha} \qquad (4)$$

Finally,

$$\varepsilon = t_{1-\alpha} * \sigma_d \qquad (5)$$

with $\sigma_d = \sqrt{\frac{\sigma_G}{\sqrt{N}} + \frac{\sigma_0}{\sqrt{n}}}$ and $t_{1-\alpha} = t_{0.95} = 1.65$

3.3. Decision rule

If $\mu_d \le \varepsilon$ then we accept the null hypothesis H_0 with a risk of 5% to be wrong, else we accept H_1.

As in DDM and EDDM, our method defines three thresholds (see Fig.2):

- The In-Control level: $\mu_d \le \varepsilon$ beyond this level we affirm that there is no change between the two distributions, so we enlarge W_G by adding W_0 's instances, and then we reset it in order to collect new ones.

- The Warning level: $\mu_d > \varepsilon + r * \sigma_d$; beyond this level, the instances are stored for an expected change. However, if the similarity between the two distributions increases, i.e. the drift is not confirmed after the Warning level, we consider that there is a false alarm and we remove the instances stored during this stage.

- The Drift level: $\mu_d > \varepsilon + s * \sigma_d$; beyond this level, the drift is confirmed and W_G is reset by only containing the instances stored since the Warning level, then the learning model is reconstructed from the new W_G.

Note that r and s are integer values which represent the amounts of change for defining respectively the Warning and Drift levels; where $s > r$.

In static context, when we consider that $r = 2$, it means that μ_d has a variance smaller than $2 * \sigma_d$ with 95% of confidence; and when we consider that $s = 3$, it means that μ_d have a variance smaller than $3 * \sigma_d$ with 99.7% of confidence. In practice, we have varied r and s values from 0 to 3 in order to study the relationship between those values and different types of drifts in non-stationary context.

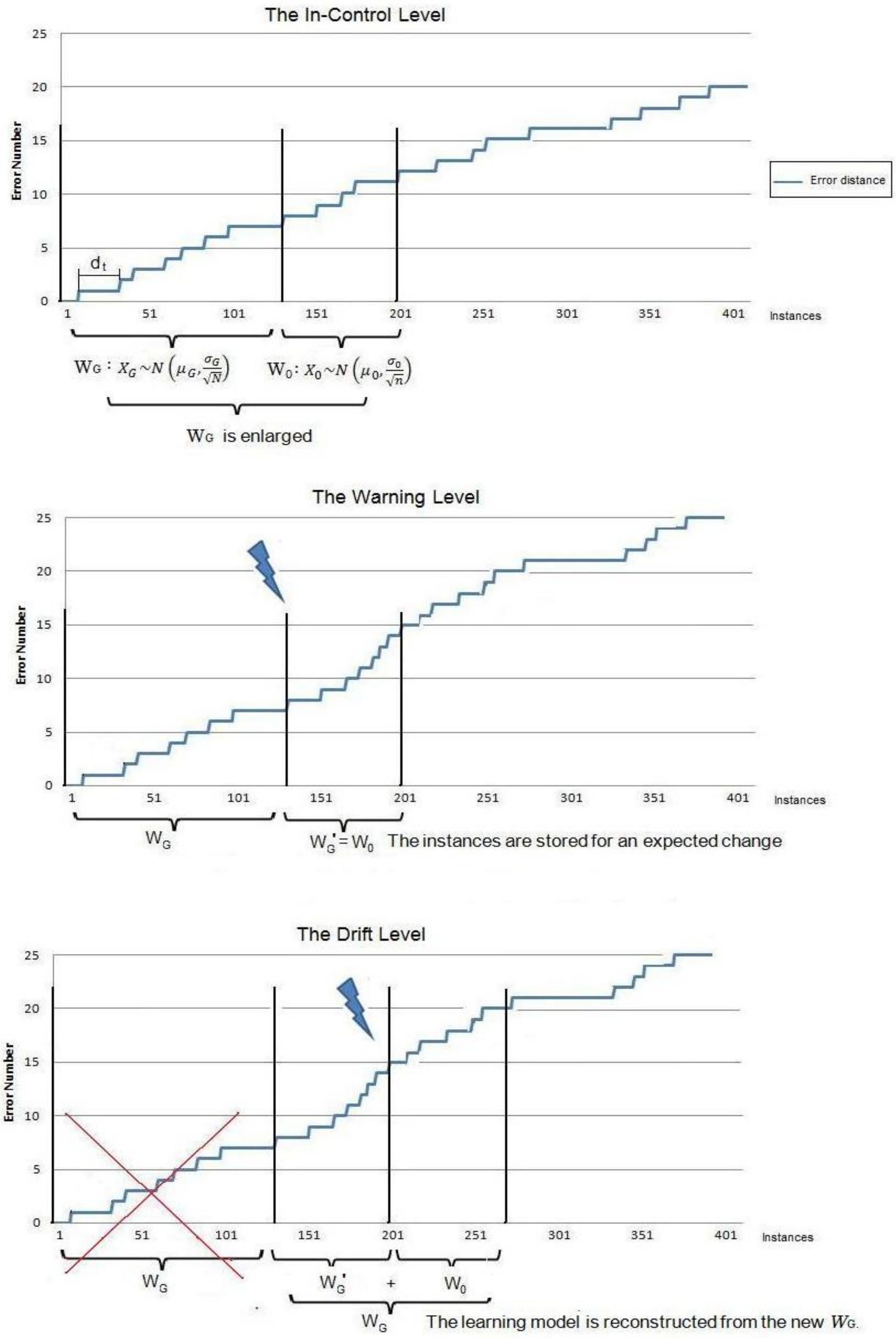


Figure 2. The three levels for drift detection and monitoring in EDIST

4. BAGEDIST AND BOOSTEDIST

In this section we describe the two ensembles learning algorithms BagEDIST and BoostEDIST, which respectively combine the Online Bagging and the Online Boosting with the drift detection method EDIST.

4.1. Online Bagging

The Off-line Bagging (Oza and Russell, 2001) builds a set of M base classifiers, training each classifier with a bootstrap sample of size N created by drawing random samples with replacement from the original training set. Each base classifier's training set contains each of the original training example k times where k is drawn from a $\text{Poisson}(1)$ distribution for large values of N.
In Online Bagging (Oza and Russell, 2001), each example is given a weight k drawn from a $\text{Poisson}(1)$ distribution, in order to be represented k times for each base classifier h_m. The ensemble prediction is done by unweighted majority vote (see Algorithm 1).

Algorithm 1: Online Bagging Algorithm

```
For all training examples $(x, y)$ do
    For each base model $h_m$ $(m \in \{1, \dots, M\})$ do
        1. Set $k \leftarrow Poisson(1)$
        2. Do $k$ times
            $updateModel(h_m, x)$
    end For
    $Prediction_{ensemble}(x) = argmax_{y \in Y} (\sum_{m=1}^{M} [y = Prediction_m(x)])$
end For
```

4.2. BagEDIST

BagEDIST is an ensemble learning algorithm which combines the Online Bagging algorithm with the drift detection method EDIST. The idea behind this combination is to develop an ensemble learning algorithm which explicitly handles concept drifts by providing useful descriptions about location, speed and severity of the changes. Unlike Online Bagging which assures the diversity by only weighting the training examples, BagEDIST proceeds by weighting the training examples and the base classifiers in the same time. As presented in algorithm 2, the weighting process of each classifier depends on its prediction performance and its ability to detect drifts. The accuracy acc_m of the classifier h_m represents the correct prediction rate, and sumDrift_m is the number of drifts detected by h_m. Notice that every classifier is equipped by EDIST as drift detection mechanism and when a drift is confirmed the classifier is relearned from the examples stored during the warning level of EDIST (more details are presented in section 2). The weight w_m is used to promote the classifier which presents good accuracy and high number of detected drifts. The parameter α is set to 2 to give more importance to classifier that achieved the best accuracy. The final ensemble prediction is done by weighted majority vote (see Algorithm 2).

Algorithm 2: BagEDIST Algorithm

```
For all training examples $(x, y)$ do
    For each base model $h_m$ $(m \in \{1, \dots, M\})$ do
        1. Set $k \leftarrow Poisson(1)$
        2. Do $k$ times
            $updateModel(h_m, x)$
        3. If $correctPrediction(h_m, x)$ then
                $acc_m \leftarrow computeAccuracy(h_m)$
        4. If $driftDetection(h_m, EDIST)$ then
                $sumDrift_m \leftarrow sumDrift_m + 1$
                reset $h_m$ using examples stored since the
                $Warning\ level$ defined by $EDIST$
           end If
        5. $w_m \leftarrow computeWeightBagEDIST(h_m, sumDrift_m, acc_m)$
    end For
    $Prediction_{ensemble}(x) = argmax_{y \in Y} (\sum_{m=1}^{M} w_m[y = Prediction_m(x)])$
end For
```

```
$computeWeightBagEDIST(h_m, sumDrift_m, acc_m)$
    If $(sumDrift_m = 0)$ then
        $w_m \leftarrow acc_m$
    else If $(sumDrift_m = 1)$ then
        If $accuracyMax(acc_m)$ then
            $w_m \leftarrow 1 + \alpha * acc_m$
        else
            $w_m \leftarrow 1 + acc_m$
        end If
    else
        If $accuracyMax(acc_m)$ then
            $w_m \leftarrow log(sumDrift_m) + \alpha * acc_m$
        else
            $w_m \leftarrow log(sumDrift_m) + acc_m$
        end If
    end If
Return $(w_m)$
```

4.3. Online Boosting

The Off-line Boosting (Oza and Russell, 2001) builds a set of M classifiers sequentially $\{\text{h}_1, \dots, \text{h}_\text{M}\}$ such that the training examples misclassified by classifier $\text{h}_{\text{m}-1}$ are given more weight for the next classifier h_m. Hence, the idea behind Boosting algorithms is to combine multiple base classifiers to obtain a small generalization error.
In Online Boosting (Oza and Russell, 2001) each example is given a weight k drawn from a $\text{Poisson}(\lambda_\text{d})$ distribution, where λ_d is increased when the training example is misclassified by the previous classifier; and decreased otherwise. $\lambda_\text{m}^\text{sc}$ and $\lambda_\text{m}^\text{sw}$ are the sum of λ_d values scaled by the half of the total weight N for respectively corrected and uncorrected examples. At final stage, each classifier h_m is weighted according to its $\lambda_\text{m}^\text{sc}$ and $\lambda_\text{m}^\text{sw}$ parameters then the final classification is done by weighted majority vote (see Algorithm 3).

Algorithm 3: Online Boosting Algorithm

```
For all training examples $(x, y)$ do
    $\lambda_d \leftarrow 1$
    For each base model $h_m$ $(m \in \{1, \dots, M\})$ do
        1. Set $k \leftarrow Poisson(\lambda_d)$
        2. Do $k$ times
            $updateModel(h_m, x)$
        3. If $correctPrediction(h_m, x)$ then
                $\lambda_m^{sc} \leftarrow \lambda_m^{sc} + \lambda_d$
                $\lambda_d \leftarrow \lambda_d * \frac{N}{2*\lambda_m^{sc}}$
            else
                $\lambda_m^{sw} \leftarrow \lambda_m^{sw} + \lambda_d$
                $\lambda_d \leftarrow \lambda_d * \frac{N}{2*\lambda_m^{sw}}$
            end If
        4.
    $w_m \leftarrow computeWeightBoosting(\lambda_m^{sc}, \lambda_m^{sw})$
    end For
    $Prediction_{ensemble}(x) = argmax_{y \in Y} (\sum_{m=1}^{M} w_m [y = Prediction_m(x)])$
end For
```

```
$computeWeightBoosting(\lambda_m^{sc}, \lambda_m^{sw})$
Calculate
    $\epsilon_m \leftarrow \frac{\lambda_m^{sw}}{\lambda_m^{sw} + \lambda_m^{sc}}$
    $\beta_m \leftarrow \frac{\epsilon_m}{1 - \epsilon_m}$
    $w_m \leftarrow log\left(\frac{1}{\beta_m}\right)$
Return $(w_m)$
```

4.4. BoostEDIST

BoostEDIST is an ensemble learning algorithm which combines the Online Boosting algorithm with the drift detection method EDIST. The first originality of BoostEDIST is that the training examples misclassified by classifier $\mathrm{h}_{\mathrm{m-1}}$and which triggered a drift are given more weight for the next classifier h_{m}. Hence, a new parameter $\lambda_{\mathrm{m}}^{\mathrm{drift}}$ is used to compute the sum of λ_{d} values when a drift is detected, and it is scaled by the total number of drifts (TotalDrifts) detected by the ensemble. Like in BagEDIST, every classifier is equipped by EDIST as drift detection mechanism and when a drift is confirmed the classifier is relearned from the examples stored during the warning level of EDIST. The second originality of BoostEDIST is that the weighting process of base classifier depends on its prediction performance and its ability to detect drifts. The weight w_{m} is computed such that the classifier which presents high $\lambda_{\mathrm{m}}^{\mathrm{sc}}$ and $\lambda_{\mathrm{m}}^{\mathrm{drift}}$ values is promoted. The final ensemble prediction is done by weighted majority vote (see Algorithm 4).

Algorithm 4: BoostEDIST Algorithm

```
For all training examples $(x, y)$ do
    $\lambda_d \leftarrow 1$
    For each base model $h_m$ $(m \in \{1, \dots, M\})$ do
        1. Set $k \leftarrow Poisson(\lambda_d)$
        2. Do $k$ times
            $updateModel(h_m, x)$
        3. If $correctPrediction(h_m, x)$ then
                $\lambda_m^{sc} \leftarrow \lambda_m^{sc} + \lambda_d$
                $\lambda_d \leftarrow \lambda_d * \frac{N}{2*\lambda_m^{sc}}$
            else
                $\lambda_m^{sw} \leftarrow \lambda_m^{sw} + \lambda_d$
                $\lambda_d \leftarrow \lambda_d * \frac{N}{2*\lambda_m^{sw}}$
            end If

        4. If $driftDetection(h_m, EDIST)$ then
                $\lambda_m^{drift} \leftarrow \lambda_m^{drift} + \lambda_d$
                $\lambda_d \leftarrow \lambda_d * \frac{TotalDrifts}{\lambda_m^{drift}}$
                reset $h_m$ using examples stored since the
            Warning level defined by $EDIST$
            end If
        5.
        $w_m \leftarrow computeWeightBoostEDIST(\lambda_m^{sc}, \lambda_m^{sw}, \lambda_m^{drift})$
    end For

$Prediction_{ensemble}(x) = argmax_{y \in Y} (\sum_{m=1}^{M} w_m [y = Prediction_m(x)])$
end For
```

```
$computeWeightBoostEDIST(\lambda_m^{sc}, \lambda_m^{sw}, \lambda_m^{drift})$
Calculate
    $\epsilon_m \leftarrow \frac{\lambda_m^{sw}}{\lambda_m^{sw} + \lambda_m^{sc} * (1 + \lambda_m^{drift})}$
    $\beta_m \leftarrow \frac{\epsilon_m}{1 - \epsilon_m}$
    $w_m \leftarrow log\left(\frac{1}{\beta_m}\right)$
Return $(w_m)$
```

5. Drift Diversity Measure

Diversity in ensemble learning algorithms is an issue that has received lots of attention in the literature. In off-line and online ensemble classifiers, the diversity could be expected to be one of the features that help to achieve good accuracy. Only some empirical studies have highlighted this hypothesis and have revealed that there could be a positive correlation between the accuracy of the ensembles and the diversity of its base classifiers (Kuncheva and Whitaker, 2003; Minku et al., 2010) However, there are still not theoretical explanations of how the diversity can enhance the performance of ensemble classifiers. Moreover, some results in the studies of Kuncheva and Whitaker (2003) have raised some doubts about the relationship between diversity and accuracy and the authors have concluded that large diversity may not always correspond to better ensemble results.

In this investigation, we intuitively ought to study the diversity of the base classifiers in BagEDIST and BoostEDIST algorithms, and see how they cope with concept drifts. For this purpose, a new diversity measure, namely Drift Diversity Measure, is proposed. Unlike the most existing diversity measures, which are often related to the accuracy of base classifiers, the Drift Diversity Measure takes into consideration the performance of each base classifier and its ability to detect drifts. This new measure may help us to have a clearer vision about the ensemble's behavior when dealing with concept drifts.

The formulation of Drift Diversity Measure (D), which is inspired from the Entropy Diversity Measure (E) (Cunningham and Carney, 2000) is as follows:

$$D = \frac{1}{N} * \sum_{i=1}^{N} \sum_{y=1}^{Y} \left(-p_{iy}^{drift} log_2 (p_{iy}^{drift})\right) \qquad (6)$$

Where N is the number of examples, Y is the number of class labels and $\mathrm{p}_{\mathrm{iy}}^{\mathrm{drift}}$ is the ratio of base classifiers that have classified the example i as class y and detected a drift. Let assume the number of base classifiers is M. If three base classifiers classify the first example as class 1 and only two classifiers have detected a drift, $p_{11}^{drift} = \frac{2}{M}$.
The Drift Diversity Measure (D) $\in [0\ ,1]$ and larger value means larger diversity of base classifiers.

6. EXPERIMENTAL EVALUATION

BagEDIST and BoostEDIST were implemented in the java programming language by extending the Massive Online Analysis (MOA) software (Bifet et al., 2010) MOA is an online learning framework for evolving data streams. It derives from the well-known WEKA framework, and supports a collection of offline and online machine learning methods for both classification and clustering.

To evaluate EDIST, we have used the Hoeffding Tree (HT) learning algorithm. HT is an incremental decision tree induction algorithm that is able to learn from a massive data streams. (for more details please refer to (Hulten et al., 2001)). We have used HT implemented in MOA with information gain split criterion and Adaptive Naive Bayes classification at leaves.

6.1. Parameter Settings

To evaluate BagEDIST and BoostEDIST algorithms, we have used the Prequential Evaluation method presented by Gama et al. (2009). This method evaluates a classifier on a stream by testing then training with each example in sequence and may use a sliding window or a fading factor forgetting mechanism. We have used the Prequential Evaluation method implemented in MOA with sliding window of size 5000.

The number of base classifiers for BagEDIST and BoostEDIST was fixed to 10, and each classifier used the Hoeffding Tree (HT) as learning algorithm and EDIST as a drift detection method.

The parameter settings of the drift detection method EDIST are as follows:

- NB represents the minimum number of examples to initialise the learning algorithm and it is set to 30
- n represents the number of errors of classification occurred in W_0 and it is set to 90
- r represents the amount of change for the warning level and it varies from 0 to 1 and,
- s represents the amount of change for the drift level and it varies from 1 to 3; with the constraint s>r.

For the comparison, we have developped BagDDM, BoostDDM, BagEDDM and BoostEDDM which use the same BagEDIST and BoostEDIST algorithms with changing EDIST by the two well-known drift detection methods DDM and EDDM. We have also compared our approaches to BagADWIN and BoostADWIN proposed by Bifet et al. (2009) and which use ADWIN (Bifet et al. 2007) as a change detector in the original version of Online Bagging and Online Boosting developed by Oza and Russell (2001).

6.2. Synthetic data sets

Synthetic data sets are primordial for studying the behaviour of the proposed algorithms where severity, width and time of change of concept drifts are known.

Rotating Hyperplane has been widely used to simulate a changing concept based on moving hyperplane (Hulten et al. 2001). A hyperplane in d-dimensional space is represented by: $\sum_{i=1}^{d} w_i x_i = w_0$. The examples which satisfy $\sum_{i=1}^{d} w_i x_i \geq w_0$ are labelled as positive, otherwise negative. The concept drift is simulated either by the orientation or the position of the hyperplane which can be changed by varying the relative size of the weights.

Agrawal Generator was introduced by Agrawal et al. (1993). It generates one of ten different pre-defined loan functions. The generator produces a stream containing nine attributes, six numeric and three categorical. This generator is based on ten functions defined for generating binary class labels from the attributes in order to determine whether the loan should be approved.

6.3. Real world datasets

Hereafter, the real world datasets used for evaluating our method.

Electricity dataset is a real world dataset from the Australian New South Wales Electricity Market. This dataset was first described by Harries (1999). In this electricity market, the prices are not fixed and may be affected by demand and supply. This dataset contains 45312 instances, dated from May 1996 to December 1998.

Forest Covertype contains the forest cover type for 30 x 30 meter cells obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS) data. It contains 581012 instances and 54 attributes, and the task is to predict the specific forest cover types.

Poker-Hand consists of 1000000 instances and 11 attributes. Each record of the Poker-Hand dataset is an instance of a hand consisting of five playing cards drawn from a standard deck of 52. Each card is described using two attributes (suit and rank), for a total of 10 predictive attributes; and one class attribute that describes the “Poker Hand”.

AirlinesDataset consists of a large amount of records, containing flight arrival and departure details for all the commercial flights within the USA, from October 1987 to April 2008. It contains 13 attributes and the task is to predict whether a given flight will be delayed or not.

7. RESULTS

In this section, the performance of BagEDIST and BoostEDIS is evaluated using data sets presenting concepts drifts of several severity, width and time change.

7.1. The diversity of base classifiers in BagEDIST and BoostEDIST algorithms

In this subsection, we use the Drift Diversity Measure to study the diversity evolution of base classifiers in BagEDIST and BoostEDIST algorithms on the Agrawal dataset of 300000 examples.

In Figure 3, the scatter plots correspond to detected drifts of every base classifier, the curves correspond to the evolution of diversity measure of the ensemble; and vertical dotted lines represent the occurrence positions of concept drifts.

From this experiment, it is noticeable that (i) some base classifiers make early detection of drifts where others detect drifts with acceptable delay. This implies that there is a good combination of base classifiers in the ensemble, in the sense that a classifier’s early detection can make up for the delay of another. This confirms that the ensemble proceeds through an effective cooperation between base classifiers in order to optimally detect drifts. (ii) When we observe the diversity measure curves, we remark that the diversity increases as soon as a drift is encountered and stabilizes when it ends. This confirms that the Drift Diversity Measure is an adequate measure for ensembles that handle drifts and can provide effective information about width and time change of drifts.

7.2. The Drift Diversity Measure Vs. accuracy

For Hyperplane datasets, we have added two concept drifts at t_0=50000 and t_0=150000 and varied the widths from 5000 to 100000 by generating 300000 examples. Results are exposed in tables 1 and 2 where columns represent the Prequential accuracies (Acc), the Drift Diversity Measure (D) and the total number of drifts detected by the ensemble (num Drifts).

As shown, BagEDIST and BoostEDIST have found the best accuracy for both abrupt and gradual drifts; and presented more robustness to noise than the others methods. In the same time, it is worth underlining that there is no clear correlation between the accuracy and the diversity for different methods, in the sense that best accuracies may not correspond to highest amounts of diversity. Another important point is that the diversity measures obtained are positively correlated with the total number of detected drifts. Hence, we can define the Drift Diversity Measure as the degree of base classifiers disagreement about detecting drifts, in the sense that the Drift Diversity Measure increases as much as classifiers disagree about the detected drifts. This definition has two interpretations. The first is that the Drift Diversity Measure is an adequate measure for drift handling ensembles because it is obvious that a set of base classifiers that identically detect drifts does not bring any advantages. The second is that when the ensemble presents high diversity and low accuracy, it may be explained by its overreacting to concept drifts or to its sensitivity to false alarms. Once again, we are facing the issue of how quantifying the ensemble diversity in order to enhance the overall accuracy.

7.3. Results on real world data sets

For the comparison, we have developped BagDDM, BoostDDM, BagEDDM and BoostEDDM which use the same BagEDIST and BoostEDIST algorithms with changing EDIST by the two well-known drift detection methods DDM and EDDM.

BagEDIST and BoostEDIST have also been tested through real world data sets widely used in similar studies. Despite the different features of each real data set, encouraging results have been found comparing to BagADWIN, BoostADWIN, and to the original version of Online Bagging and Online Boosting (OzaBag and OzaBoost).

These results reaffirm the effectiveness of the adaptation of the original version of Bagging and Boosting for drift handling with the use of any drift detection method DDM, EDDM or EDIST (see tables 3 and 4).

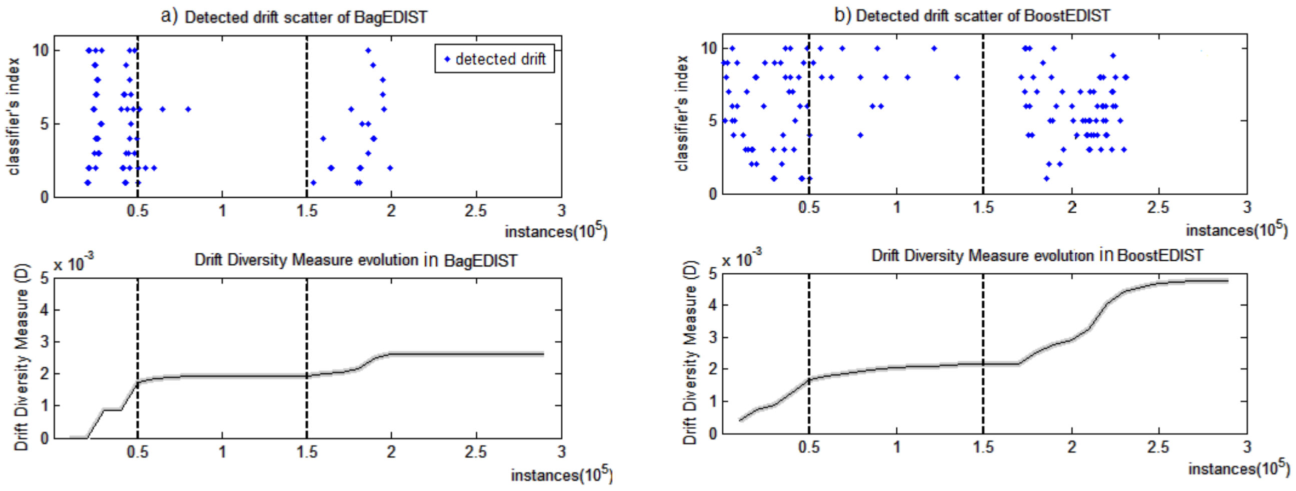


Figure 3. The diversity of base classifiers for repectively BagEDIST (a) and BoostEDIST (b) algorithms in Agrawal dataset containing 300000 examples, 2 concept drifts at t_0=50000 and t_0=150000 with width=50000.

Table 1. Prequential accuracy (Acc), the Drift Diversity Measure (D) and the total number of drifts detected by the ensemble (num Drifts) in Hyperplane datasets containing 300000 examples, 2 concept drifts at t_0=50000 and t_0=150000, 20% then 40% of severity and 30% of noise.

	Drift width											
	5000			10000			50000			100000		
	acc	(D)	num drifts	acc	(D)	num drifts	acc	(D)	num drifts	acc	(D)	num drifts
BagEDIST	68,400	8,969E-05	81	68,800	8,083E-05	73	68,700	5,979E-05	54	67,100	5,426E-05	49
BagDDM	67,800	7,308E-05	66	68,400	6,312E-05	37	68,100	6,422E-05	58	66,000	2,215E-05	20
BagEDDM	67,800	1,484E-04	134	68,300	1,440E-04	130	68,400	1,583E-04	143	67,200	1,532E-04	139

Table 2. Prequential accuracy (Acc) , the Drift Diversity Measure (D) and the total number of drifts detected by the ensemble (num Drifts) in Hyperplane datasets containing 300000 examples, 2 concept drifts at t_0=50000 and t_0=150000, 20% then 40% of severity and 30% of noise.

	Drift width											
	5000			10000			50000			100000		
	acc	(D)	num drifts	acc	(D)	num drifts	acc	(D)	num drifts	acc	(D)	num drifts
BoostEDIST	69,900	7,530E-05	68	68,400	8,083E-05	73	68,800	1,882E-05	17	66,200	3,986E-05	36
BoostDDM	68,000	3,100E-05	28	68,400	3,322E-05	30	68,000	1,661E-05	15	65,900	1,107E-05	10
BoostEDDM	67,200	8,637E-05	78	67,100	1,207E-04	109	66,900	1,008E-04	91	66,100	6,755E-05	61

Table 3. Prequential accuracy of different Bagging ensemble algorithms in real world data sets.

	ELEC2	Covtyp	Poker	Airline
BagEDIST	91,800	96,800	67,600	63,500
BagDDM	87,000	97,400	64,300	62,100
BagEDDM	81,400	97,200	78,000	63,300
BagADWIN	75,300	97,100	65,500	61,600
OzaBag	88,500	91,000	63,300	64,200

Table 4. Prequential accuracy of different Boosting ensemble algorithms in real world data sets.

	ELEC2	Covtyp	Poker	Airline
BoostEDIST	89,800	96,500	80,000	59,900
BoostDDM	89,300	96,400	81,100	60,100
BoostEDDM	88,400	97,100	90,800	62,800
BoostAdwin	88,000	97,200	85,800	58,200
OzaBoost	87,400	91,500	88,800	61,300

8. Conclusion

This paper has presented two ensembles learning algorithms BagEDIST and BoostEDIST, which respectively combine the Online Bagging and the Online Boosting with the drift detection method EDIST. EDIST is a new drift detection method which monitors the distance between two consecutive errors of classification, and tracks concept drifts through two adaptive data-windows W_G and W_0 . EDIST makes use of a statistical hypothesis test in order to compare W_G and W_0 error distance distributions and checks whether the averages differ by more than the adjusted threshold ε .

BagEDIST and BoostEDIST are two ensemble learning algorithms which explicitly handle concept drifts by providing useful description about severity, width and time of change of drift. The originality behind these approaches is that the weighting process of each base classifier depends on its prediction performance and its ability to detect drifts. Moreover, in BoostEDIST, the instances which trigger drifts are given more importance for next base classifiers. During the experiments we have noticed that there is an effective cooperation between base classifiers to optimally cope with concept drifts, in the sense that a classifier's early detection can make up for the delay of another.

This paper has also presented a new Drift Diversity Measure in order to study the diversity of base classifiers and see how they cope with concept drifts. Unlike the existing diversity measures, which are often related to the accuracy of base classifiers, the Drift Diversity Measure takes into consideration the performance of each base classifier and its ability to detect drifts. Through previous experiments, this new measure has helped us to have a clearer vision about the ensemble's behavior when dealing with concept drifts.

References

Agrawal, R., Imielinski, T., & Swami., A. (1993). Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, pp. 914–925.

Baena-García, M., Campo-Avila, J. D., Fidalgo, R., Bifet, A., Gavaldà, R., & Morales-Bueno, R. (2006). Early drift detection method. In *Proceedings of the Fourth International Workshop on Knowledge Discovery from DataStreams*, Berlin, Germany, pp. 77-86.

Bifet, A., & Gavald, R.(2007) Learning from time-changing data with adaptive windowing. In *Proceeding of 7th International Conference on Data Mining*, Minnesota, USA, pp. 443-448.

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavalda, R,. (2009). New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, Paris, France, pp 139-148.

Bifet, A., Holmes, G., & Pfahringer, B.,(2010) Leveraging bagging for evolving data streams machine learning and knowledge discovery in databases. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Barcelona, Spain, pp 135-150.

Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B.(2010). MOA: Massive Online Analysis. *Journal of Machine Learning Research*, vol. 11, pp. 1601-1604.

Brzezinski, D., & Stefanowski, J.,(2011). Accuracy Updated Ensemble for Data Streams with Concept Drift. In *Proceedings of the 6th international conference on Hybrid artificial intelligent systems*, Wroclaw, Poland, pp 155-163.

Cunningham, P., & Carney, J., (2000). Diversity versus Quality in Classification Ensembles based on Feature Selection. In *Proceedings 11th European Conference on Machine Learning*. Barcelona, Spain, pp. 109-116.

Domingos, P., & Hulten G.,(2000). Mining high-speed data streams. In *the Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining*, Boston, MA, USA, pp. 71-80.

Gama, J., Medas, P., Castillo, G., & Rodrigues. P.,(2006) Learning with local drift detection. In *Proceedings of the Second International Conference on Advanced Data Mining and Applications*, Xi'an, China. pp. 42-55.

Gama, J., Sebastião, R., & Rodrigues, P.,(2009). Issues in evaluation of stream learning algorithms. In *the Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, Paris, France, pp. 329–338.

Harries, M.,(1999). Splice-2 comparative evaluation: Electricity pricing. Technical Report, The University of South Wales, Autralia.

Hulten, G., Spencer, L., & Domingos, P.,(2001). Mining time-changing data streams. In *Proceedings of the Seventh ACM SIGKDD international conference on Knowledge Discovery and Data Mining,* California, USA, pp. 97-106.

Ikonomovska, E., Gama, J., Sebastio, R., & Gjorgjevik, D.,(2009). Regression trees from data streams with drift detection. In *Proceedings of the 12th International Conference on Discovery Science,* Berlin, Germany, pp. 121–135.

Klinkenberg, R. (2001). Learning drifting concepts: example selection vs. example weighting. *Intelligent Data Analysis,* vol. 8 , pp. 281–300.

Kolter, J., & Maloof, M., (2007). Dynamic weighted majority: a new ensemble method for tracking concept drift. *The Journal of Machine Learning Research,* vol. 8. pp. 2755-2790.

Kuncheva, L., (2004). Classifier Ensembles for Changing Environments. In *Proceedings of the 5th International Workshop on Multiple Classifier Systems,* Cagliari, Italy, pp. 1-15.

Kuncheva, L. I., & Whitaker, C. J, (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Journal of Machine Learning,* vol. 51, pp. 181–207.

Lazarescu, M., Venkateshand, S.,& Bui, H., (2004). Using multiple windows to track concept drift. *Intelligent data analysis,* vol. 8, pp. 29-59.

Lughofer, E., & Angelov, P.,(2011). Handling Drifts and Shifts in On-Line Data Streams with Evolving Fuzzy Systems. *Applied Soft Computing*, vol. 11, pp. 2057-2068.

Masud, M., Gao, J., Khan, L., Han, J., & Thuraisingham, B., (2011). Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge and Data Engineering,* vol. 23, pp. 859–874.

Mitchell, T,(1997). *Machine Learning*. McGraw Hill, New York, USA.

Minku, L., White, A., & Yao, X.,(2010). The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Transactions on Knowledge and Data Engineering,* vol. 22, pp. 730–742.

Oza, N., & Russell, S., (2001). Online bagging and boosting. In *Proceedings of the Eighth International Workshop of Artificial Intelligence and Statistics,* Florida,USA, pp. 105-112.

Sayed-Mouchaweh, M., (2010). Semi-supervised classification method for dynamic applications. *Fuzzy Sets and Systems,* vol. 4, pp. 544–563.

Schlimmer, J. C., & Granger, R. H. (1986). Incremental learning from noisy data. *Journal of Machine Learning,* vol. 3, pp. 317-354.

Sobhani, P., & Beigy, H., (2011). New drift detection method for data streams. In *Proceedings of the second international conference on Adaptive and intelligent systems,* Berlin, Germany, pp. 88-97.

Tsymbal, A., (2004). The problem of concept drift: definitions and related work. Technical Report TCD-CS-2004-15, Trinity College, Dublin, Ireland.

BIOGRAPHIES

Imen Khamassi is a Ph.D. student at the High Institute of Management of the University of Tunis in Tunisia (ISG-Tunis). She received his Master degree in Computer Science and Business Intelligence from (ISG- Tunis) in 2010. She is member in the research laboratory SOIE (Tunisia) and member in the Tunisian Association of Artificial Intelligence (ATIA-Tunisia). Her research interests include combinatorial optimization, learning control, meta-heuristics, intelligent systems and their applications to decision making.

Moamar Sayed-Mouchaweh received his Master degree from the University of Technology of Compiegne-France in 1999. Then, he received his PhD degree from the University of Reims-France in December 2002. He was nominated as Associated Professor in Computer Science, Control and Signal processing at the University of Reims-France in the Research center in Sciences and Technology of the Information and the Communication (CReSTIC). In December 2008, he obtained the Habilitation to Direct Researches (HDR) in Computer science, Control and Signal processing. Since September 2011, he is working as a Full Professor in the High National Engineering School of Mines "EcoleNationaleSupérieure des Mines de Douai" at the Department of Automatic Control and Computer Science (Informatique&Automatique IA). He supervised several defended Master and PhD thesis as well as research projects in the field of Modeling, Monitoring and Diagnosis in non-stationary environments. He published more than 100 journal and conference papers. He served as International Program Committee member for several International Conferences as well as a member in IEEE and IFAC technical committees. He also (co-) organized several special sessions and presented several tutorials. He is an associate

Editor of the Springer international journal “Evolving Systems”.

Moez Hammami has received the M.S. and Ph.D. degrees in Computer Science from the High Institute of Management of the University of Tunis in Tunisia (ISG-Tunis). In 2005, he was nominated as Associated Professor in Computer Science in (ISG- Tunis). He is member in the research laboratory SOIE (Tunisia) and member in the Tunisian Association of Artificial Intelligence (ATIA-Tunisia). His research interests include combinatorial optimization, meta-heuristics, constraint satisfaction problem and multi agent systems.

Khaled Ghédira Khaled Ghédira has received the bachelor degree in mathematics (FS-Tunisia); the Engineer degree in hydraulic (ENSEEIHT-France); the specialized Engineer degree in computer science and applied mathematics (ENSIMAG-France), both the Master and the Ph.D. degrees in Artificial Intelligence (ENSAE-France) and finally the Habilitation degree in computer science (ENSI-Tunisia). He was Research Fellow at IIIA (Switzerland 1992-96), expert consultant at BT (England 1995), the head of ENSI (2002-08) and from April 2011, the general managing director of the Tunis Science City. He is also Professor at ISG (University of Tunis), the founding president of the Tunisian Association of Artificial Intelligence (ATIA-Tunisia) and the founding director of the research laboratory SOIE (Tunisia). He is member of several international scientific networks/program committees and is often invited as keynote speaker/ visiting professor at national and international. He was also member of the think national committee for higher education (training of trainers, LMD) and member/president of several committees: evaluation of higher education institutions, research projects reviewing, teachers recruiting. His research areas include multi agent system, transport and production logistics, meta-heuristics and security in M/E-government. He has led several national and international research projects. He has supervised twenty PhD thesis and more than 50 master thesis. He has co/authored about 300 journal/conference/book research papers. He has written two text books on metaheuristics and production logistics and co-authored three others.

Distribution Free Prediction Interval for Uncertainty Quantification in Remaining Useful Life Prediction

Huimin Chen [1]

[1] *Department of Electrical Engineering, University of New Orleans, New Orleans, LA, 70148, U. S. A.*
hchen2@uno.edu

Abstract

Remaining useful life (RUL) prediction is an important component for system health monitoring and prognosis. Ideally, one expects the prediction algorithm to provide the complete distribution of the RUL prediction over time taking various uncertainties into account. However, the dynamic model being used to characterize state estimation and future loading uncertainties is often simplified through various approximations, leading to non-credible predicted distribution. Nevertheless, certain algorithm may only provide a point estimate of the RUL, making it difficult to quantify the uncertainty of the prediction. In this paper, we focus on interval prediction with high probability that guarantees finite sample validity without the knowledge of statistical distribution of the noise. The key idea is to leverage the newly proposed conformal prediction framework with non-parametric conditional density estimation. Under certain regularity conditions, the proposed interval estimator converges to an oracle band at a minimax optimal rate. In addition, we apply a data driven method to automatically select the bandwidth in the kernel density estimator. We discuss practical approximations to speed up the computation. The proposed method can be used to predict the RUL interval with physics-based model in a distribution free manner. It can also be applied to assess the validity of other prognostic algorithms from experimental data. We demonstrate the effectiveness of the RUL prediction for Li-Ion batteries using both simulated and experimental data.

1. Introduction

Remaining useful life (RUL) prediction is an important component for prognosis and system health monitoring. Ideally, one expects the prediction algorithm to provide the complete distribution of the RUL prediction over time taking various uncertainties into account. However, the dynamic model being used to characterize state estimation and future loading uncertainties is often simplified with various approximations, resulting in non-credible predicted distribution. Nevertheless, certain algorithms often provide point estimates of the RUL, making it difficult to quantify the uncertainty of the prediction. Existing efforts for uncertainty quantification can be largely classified into two categories: point-based and density-based credibility tests. In point-based tests, one wants to find the probability that the RUL is longer than the estimate so that the critical component can be replaced before its failure. In many cases, one has to ensure that the probability is above a certain confidence level, say 0.95, in order to declare that the point estimate of the RUL is valid. Among all the valid RUL predictions, it is desirable to find the least conservative one, i.e., the largest RUL prediction for decision-making. However, a prediction point with a desirable confidence level does not fully characterize the uncertainty of the RUL estimate. The complete description requires the posterior probability density function of the RUL, which is hard to be fully specified when one has the uncertainty of future loading that will affect the system dynamics during the predicted time horizon. Density-based tests address whether the posterior distribution of the RUL provided by a prognostic algorithm is valid in certain desirable notions. For example, one can check wether the whole distribution is valid with a desirable significance level by Kolmogorov-Smirnov test (Justel, Pena, & Zamar, 1997). This may require a large number of samples for any statistically meaningful result. Alternatively, one may test the statistical significance at some value of the cumulative distribution function of the RUL estimate using Fisher's exact test (Fisher, 1954). However, the method only ensures validity of a point estimate in the RUL and throws away other useful information contained in the probability density function.

It is clear that a point estimate of the RUL seems to be inadequate to quantify the uncertainty. However, the whole posterior density of the RUL is hard to come by. Nevertheless, density-based prognostic methods have been used by people engaged in various density-based state estimation techniques, e.g., particle filters (Saha & Goebel, 2011). Note that the posterior density is valid only when the assumed process noise distribution is true across the entire RUL prediction horizon.

Huimin Chen et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

In practice, the prediction interval of the RUL with high probability is often useful for monitoring a mission critical component. The lower limit serves as the valid point estimate of the RUL while the upper limit shows the full potential of the component in account for the future loading uncertainty. For a given confidence level, one seeks the smallest interval to contain the RUL with a probability no less than the desired confidence level. This boils down to the problem of interval prediction with high probability that guarantees finite sample validity without the knowledge of statistical distribution of the noise. The key idea is to leverage the newly proposed conformal prediction framework (Shafer & Vovk, 2008) with non-parametric conditional density estimation (Rosenblatt, 1956). Under certain regularity conditions, the proposed interval estimator converges to an oracle band at a minimax optimal rate. In addition, we apply a data driven method to automatically select the bandwidth in the kernel density estimator and discuss practical approximations to speed up the computation. The method can be used to predict the RUL interval with physics-based model in a distribution free manner. It can also be applied to assess the validity of any point-based prognostic algorithm from the experimental data. We demonstrate the effectiveness of the RUL prediction for Li-Ion batteries using both simulated and experimental data.

The rest of the paper is organized as follows. Section 2 formulates the RUL prediction problem. Section 3 presents the distribution free interval estimation of the component's end-of-life and discusses its asymptotic properties. Section 4 provides the experimental study on the state-of-charge estimation of Li-Ion battery using the predicted RUL intervals of a desired confidence level. Concluding summary is in Section 5.

2. The Formulation of RUL Prediction Problem

State space model has been commonly used to describe the dynamics of a component such as the aging process during repeated usage (Luo et al., 2003). In general, one can apply the principle of physics to model the dynamics of a component by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{p}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (1)$$

where $\mathbf{x}(t)$ is the state vector, $\mathbf{p}(t)$ is the parameter vector that may change over time but not governed by the differential equation as the state $\mathbf{x}(t)$, $\mathbf{u}(t)$ is the input vector, and $\mathbf{v}(t)$ is the process noise vector. The state and parameter can be observed through a generic continuous time model

$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \mathbf{p}(t), \mathbf{u}(t), \mathbf{w}(t)) \quad (2)$$

where $\mathbf{w}(t)$ is the measurement noise vector. In practice, measurements are made at discrete time time instants t_1, ..., t_k, ... and the measurement model becomes

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{p}_k, \mathbf{u}_k, \mathbf{w}_k) \quad (3)$$

where the subscript k indicates that the measurement $\mathbf{y}_k$ is obtained at time t_k. Denote by $\mathbf{Y}_k$ the measurement sequence $\mathbf{y}_1$, ..., $\mathbf{y}_k$. The state estimation problem is to obtain the posterior distribution $p(\mathbf{x}_k|\mathbf{Y}_k)$ at any time t_k using the prior knowledge $p(\mathbf{x}_0)$ and the measurements $\mathbf{Y}_k$. The desired performance of the component can be characterized by a set of constraints from the state, parameter and input space to some set that fully characterizes the normal condition (Sankararaman & Goebel, 2013). Specifically, we let $c_i(T) = 1$ if the constraint is satisfied, i.e., $\mathbf{g}_i(\mathbf{x}(t), \mathbf{p}(t), \mathbf{u}(t)) \in B_i$ for $t \in [0, T]$, where $\mathbf{g}_i$ is a known mapping and B_i is the set of acceptable values. Otherwise, $c_i(T) = 0$. The end-of-life (EOL) of a component is defined as the earliest time that one of the n-constraints is violated. Denote by t_{EOL} the end-of-life of the component given the state, parameter and input vectors, i.e.,

$$t_{\mathrm{EOL}} = \inf_t \left\{ t \,\middle|\, \prod_{i=1}^{n} c_i(t) = 0 \right\} \quad (4)$$

The remaining useful life (RUL) of the component at time t_P is given by $t_{\mathrm{EOL}} - t_P$. Owing to the uncertainty in the state dynamics, the state $\mathbf{x}(t)$ is a random process. Thus the constraints have to be assessed in a probabilistic sense. Let $\alpha \in [0, 1]$ be the significance level and we define the EOL as a function of α by

$$t_{\mathrm{EOL}}(\alpha) = \inf_t \left\{ t \,\middle|\, P\left(\prod_{i=1}^{n} c_i(t) = 0\right) \leq \alpha \right\} \quad (5)$$

where the constraint c_i is evaluated by $\mathbf{g}_i(\hat{\mathbf{x}}(t), \hat{\mathbf{p}}(t), \mathbf{u}(t))$ using the state and parameter estimates with the measurements up to time t_P. Note that one can propagate the state, parameter and input uncertainties into the future and assess the constraint satisfaction c_i probabilistically. For small α, $t_{\mathrm{EOL}}(\alpha)$ may be conservative especially when the uncertainty of the state dynamics is large initially with limited number of measurements. The RUL prediction becomes a density estimation problem if one wants to fully characterize $t_{\mathrm{EOL}}(\alpha)$ at any time t_P. It is computationally demanding and does not have a closed form expression for a problem with either nonlinear dynamics or nonlinear constraint.

3. Distribution Free Prediction Interval

Consider a generic RUL prediction problem where one observes $\mathbf{Y}_k$ and $\mathbf{U}_k$ up to time t_k and applies state estimation algorithm to obtain $\hat{\mathbf{x}}_k$ with the associated error covariance P_k. In order to estimate t_{EOL}, one has to propagate the dynamic model (1) from t_k with either known (deterministic) future input or random future input with the anticipated uncertainty given by a probabilistic model. Clearly, exact characterization of (5) is computationally challenging and one may only have a point estimate of the component's EOL, namely, $\hat{t}_{k\mathrm{EOL}}$ at time t_k. In principle, the estimate of t_{EOL} can be based on the maximum a posteriori (MAP) criterion, which does not necessarily need the complete knowledge of the posterior distribution. However, a prediction interval $[t_{k\min}, t_{k\max}]$ of the t_{EOL} conditioned on $\mathbf{Y}_k$ and $\mathbf{U}_k$ with a guaranteed confidence level α is often desirable. Existing criterion to evaluate an RUL prediction algorithm using the prediction interval such as the α-λ metric (Saxena, Celaya, Saha, Saha, & Goebel, 2009) requires the complete distribution from the prediction algorithm, which is often unavailable from the point-based estimators. Nevertheless, the performance assessment can be misleading when the predictive distribution is not credible. To circumvent the above challenge, we take an alternative route to generate distribution free prediction interval using only the point estimate $\hat{t}_{k\mathrm{EOL}}$. The main idea is to generate conformal prediction interval so that

$$P(t_{\mathrm{EOL}} \in [t_{k\min}, t_{k\max}]) \geq \alpha$$

for any distribution of the state and future input. The usual way to obtain a non-parametric prediction interval leads to the form

$$\left[\hat{t}_{k\mathrm{EOL}} - Z_{\alpha/2}\sqrt{\hat{\sigma}_k^2 + s^2}, \hat{t}_{k\mathrm{EOL}} + Z_{\alpha/2}\sqrt{\hat{\sigma}_k^2 + s^2}\right]$$

where $\hat{t}_{k\mathrm{EOL}}$ is the point estimate from the prediction algorithm, $Z_{\alpha/2}$ is the normal quantile, $\hat{\sigma}_k^2$ is the estimated conditional variance, and s^2 is the estimated error of the prediction algorithm using bootstrapping (Efron & Tibshirani, 1993). However, such a prediction interval does not have distribution free finite sample validity.

3.1. Constructing Valid Prediction Interval

We apply the recently proposed conformal prediction method (Shafer & Vovk, 2008) to ensure the finite sample validity. Specifically, let $\hat{t}_{1\mathrm{EOL}}$, ..., $\hat{t}_{k\mathrm{EOL}}$ be a random sequence following an unknown distribution. We can estimate the density of the prediction sequence and denote the estimated density by $\hat{p}_k(t_{\mathrm{EOL}})$. The p-value from k samples is given by

$$\pi_k(t) = \frac{1}{k+1}\sum_{i=1}^{k+1} 1\left(\hat{p}_i(t) \leq \hat{p}_k(t)\right) \tag{6}$$

where $1(\cdot)$ is the indicator function. The prediction interval for the $(k+1)$-th sample with confidence level α can be obtained by

$$C_{k+1}(\alpha) = \{t : \pi_k(t) \geq 1 - \alpha\} \tag{7}$$

Intuitively, we want to test the hypothesis $H_0 : \hat{t}_{k+1\mathrm{EOL}} = t$ for arbitrary value t and use the height of the density estimate as a test statistic. Since the vector $(\hat{p}_1, ..., \hat{p}_k)$ is exchangeable, $\pi_k(t)$ is uniformly distributed in $[0, 1]$ and is a valid p-value for the significance test of H_0. The set $C_k(\alpha)$ contains all values t that are not rejected by the test, thus

$$P(\hat{t}_{k+1\mathrm{EOL}} \in C_{k+1}) \geq \alpha$$

for any distribution.

Computing $C_{k+1}(\alpha)$ is expensive since one has to find the p-value $\pi_k(t)$ for every t. An approximation is made assuming that $\hat{p}_k(t)$ can be well estimated by a kernel density estimator as the number of samples increases. With the exchangeability assumption, we can order the predictions $\hat{t}_{1\mathrm{EOL}}$, ..., $\hat{t}_{k\mathrm{EOL}}$ increasingly such that $\hat{p}_k(\hat{t}_{1\mathrm{EOL}}) \leq ... \leq \hat{p}_k(\hat{t}_{k\mathrm{EOL}})$. Let $j = \lfloor k\alpha \rfloor$ and define

$$C_{k+1}^{+}(\alpha) = \left\{t : \hat{p}_k(t) \geq \hat{p}_k(t_{k\mathrm{EOL}}) - \frac{K(0)}{kh}\right\} \tag{8}$$

where $K(\cdot)$ is the kernel function used to estimate $\hat{p}_k$ and h is the bandwidth of the kernel density estimator. It can be shown that $C_{k+1}^{+}(\alpha)$ also has finite sample validity and it has the same efficiency as $C_{k+1}(\alpha)$ asymptotically if the bandwidth h is chosen appropriately (Lei, Robins, & Wasserman, 2011).

3.2. Asymptotic Properties

Let $\hat{p}_k(t)$ be the kernel density estimate conditioned on $\hat{t}_{1\mathrm{EOL}}$, ..., $\hat{t}_{k\mathrm{EOL}}$. We can recursively update the estimate by

$$\hat{p}_{k+1}(t) = \frac{k}{k+1}\hat{p}_k(t) + \frac{1}{(k+1)h}K\left(\frac{\hat{t}_{k+1\mathrm{EOL}} - t}{h}\right)$$

with some smooth kernel $K(\cdot)$. The p-value at an arbitrary time is estimated by

$$\pi_i = \frac{1}{k+1}\sum_{j=1}^{k+1} 1\left(\hat{p}_j \leq \hat{p}_k\right), 1 \leq i \leq n+1$$

where we dropped the time argument for simplicity. We have the distribution free prediction interval satisfying

$$P(t_{\mathrm{EOL}} \in C_{k+1}^{+}) \geq \alpha$$

for any chosen $\alpha \in (0, 1)$. However, such a prediction interval is not unique and we would hope to shrink the interval without losing the validity conditioned on the input and observation.

Consider a partition $\mathcal{A} = \{A_j, j \geq 1\}$ of length s_k. Let $n_j = \sum_{i=1}^{k} 1(\hat{t}_{i\mathrm{EOL}} \in A_j)$ be the histogram counts. A local marginal kernel density estimate is

$$\hat{p}(t|A_j) = \frac{1}{n_j h_k}\sum_{i=1}^{k} 1(\hat{t}_{i\mathrm{EOL}} \in A_j)K\left(\frac{\hat{t}_{i\mathrm{EOL}} - t}{h_k}\right)$$

where h_k is the kernel bandwidth. The local conditional density rank can be defined as

$$\begin{aligned}\pi_{k,j} \quad = \quad & \frac{1}{n_j+1}\sum_{i=1}^{k+1} 1(\hat{t}_{i\mathrm{EOL}} \in A_j)\cdot \\ & 1(\hat{p}(\hat{t}_{i\mathrm{EOL}}|A_j) \leq \hat{p}(\hat{t}_{k+1\mathrm{EOL}}|A_j))\end{aligned}$$

and the prediction interval with confidence α is

$$C_{k,j}(\alpha) = \{t : \pi_{k,j}(t) \leq 1-\alpha\}.$$

We can see that the density estimate changes very little when t varies inside A_j. When A_j has a small diameter and the kernel function is smooth enough, the local sample approximates independent observations drawn inside the partition A_j. Thus we can optimize the kernel bandwidth h_k adaptively to achieve the smallest valid prediction interval asymptotically. The tuning of the partition size s_k and kernel bandwidth h_k depends on the smoothness of the conditional density. It can be trained by using a subset of the samples to construct a local conformal prediction interval and then increasing the bandwidth until C_{k+1} can not be further reduced.

Next, we argue that the convergence rate of the optimized local conformal prediction interval is asymptotically minimax optimal compared with the oracle prediction interval. Assume that the density of t_{EOL} satisfies $0 < c_0 \leq p_{t_{\mathrm{EOL}}}(t) \leq c_1 \leq +\infty$ for any input and observation sequences. In addition, we assume that $p_{t_{\mathrm{EOL}}}(t)$ is β times differentiable and uniformly bounded by L. In particular, the conditional density is Lipschitz in t, i.e.,

$$||p_{t_{\mathrm{EOL}}}(\cdot|t_1) - p_{t_{\mathrm{EOL}}}(\cdot|t_2)||_\infty \leq L|t_1 - t_2|$$

to ensure that the kernel function $K(\cdot)$ is a smooth approximation of $p_{t_{\mathrm{EOL}}}(t)$ of order β. Let $C^*(\alpha)$ be the oracle prediction interval given by

$$C^*(\alpha) = \{t : p_{t_{\mathrm{EOL}}}(t|\mathbf{Y}_\infty, \mathbf{U}_\infty) \geq t(\alpha)\}$$

where $t(\alpha)$ is the appropriately chosen threshold to achieve $P(t_{\mathrm{EOL}} \in C^*(\alpha)) = \alpha$. If we choose $s_k \sim r_k$ and $h_k \sim r_k^{1/\beta}$, then the prediction interval $C_k^+(\alpha))$ satisfies

$$P\left(\sup_t \mu\left(C_k^+(\alpha)\bigcap\neg C^*(\alpha)\right) \geq c_\lambda r_k\right) \sim O(k^{-\lambda})$$

for any $\alpha \in (0,1)$, $\lambda > 0$ and some constant c_λ being independent of k. The measure μ shows the difference between the kernel density estimate and the oracle prediction intervals while the critical rate r_k is

$$r_k = \left(\frac{\log k}{k}\right)^{\frac{\beta}{2\beta+1}}.$$

As the sample size increases, the prediction interval by kernel density estimate converges to the oracle prediction interval and the rate can not be improved in the minimax sense, i.e.,

$$\inf_{C_k^+(\alpha)}\sup_{p_{t_{\mathrm{EOL}}}} E_{p_{t_{\mathrm{EOL}}}}\left[\mu\left(C_k^+(\alpha)\bigcap\neg C^*(\alpha)\right)\right] \geq c r_k$$

for some constant $c > 0$. The proof invokes generalized Fano's lemma (Tsybakov, 2009) where the supremum is over all distribution $p_{t_{\mathrm{EOL}}}$ such that $p_{t_{\mathrm{EOL}}}(\cdot|x)$ is Lipschitz in x in the sup-norm sense and $p_{t_{\mathrm{EOL}}}(t)$ is smooth enough, i.e., β times differentiable. The kernel bandwidth $h = cr_k$ with a small enough constant c will converge to the oracle prediction interval and guarantee the finite sample validity simultaneously.

4. Battery State-of-Charge Estimation

Lithium-ion battery is the core of new plug-in hybrid-electrical vehicles as well as considered in many 2nd generation hybrid electric vehicles. The lithium-ion battery performance plays an important role for the energy management of these vehicles as high-rate transient power source cycling around a relatively fixed state of charge (SOC). The estimation of state-of-charge and state-of-health of the battery cell has drawn significant attention in battery health management (Charkhgard & Farrokhi, 2010; Chiasson & Vairamohan, 2005; Kim & Cho, 2011; Klein et al., 2013; Saha, Goebel, & Christophersen, 2009).

4.1. Battery Dynamic and Measurement Models

We adopt the enhanced self-correcting cell model (Plett, 2004a) which contains both state and unknown parameters. The model is simplified from the detailed electro-chemical model (Klein et al., 2013) and it includes open-circuit voltage, internal resistance, voltage time constant, and hysteresis. Assume the sampling interval is T_s. At time k, the current i_k is nearly a constant. The state-of-charge (SoC) c_k is governed by

$$c_{k+1} = c_k - \left(\frac{\eta T_s}{C}\right) i_k$$

where η is the Coulombic efficiency factor at current level i_k and C is the cell capacity in Ampere-seconds. The time constants of the cell voltage response are captured by several internal states. Let $\mathbf{z}_k$ be the internal state vector at time k. A linear model was suggested in (Plett, 2004a) given by

$$\mathbf{z}_{k+1} = A_{\mathbf{z}}\mathbf{z}_k + B_{\mathbf{z}} i_k$$

where $A_{\mathbf{z}}$ is a diagonal matrix with real valued entries and $B_{\mathbf{z}}$ is chosen to have all 1s. The hysteresis level is modeled by

$$h_{k+1} = e^{-\left|\frac{\eta i_k \gamma T_s}{C}\right|} h_k + \left[1 - e^{-\left|\frac{\eta i_k \gamma T_s}{C}\right|}\right] \mathrm{sgn}(i_k)$$

where γ is the hysteresis rate constant. The voltage is

$$v_k = \mathrm{OCV}(c_k) + G\mathbf{z}_k - Ri_k + Mh_k$$

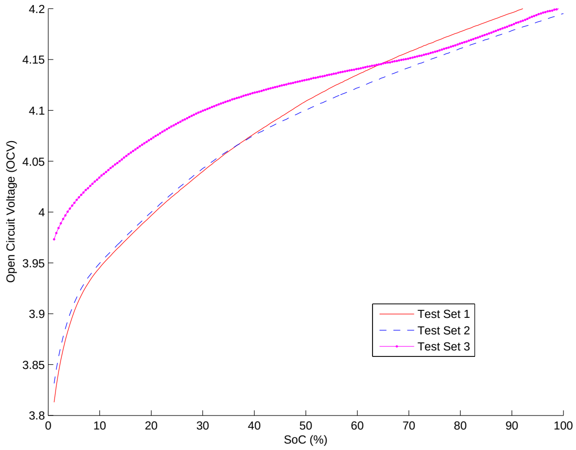


Figure 1. Empirical function of open circuit voltage vs. state-of-charge of Li-Ion battery sets

where $\mathrm{OCV}(\cdot)$ is an empirical function found by battery cell testing; G is a vector of unknown parameters related to the battery aging status; R is the internal resistance; and M is the maximum hysteresis level. Thus the overall state vector is $\mathbf{x}_k = [c_k \; \mathbf{z}_k \; h_k]^T$. The dynamic model contains unknown parameters η, γ, and G. The measurements are voltage v_k and current i_k. The primary goal is to accurately estimate the state-of-charge c_k during the battery usage with variable load.

4.2. Experimental Results

The Li-Ion battery cells used for experimental study were divided into three sets. Set 1 of two batteries was used to tune the cell model parameters (e.g., $\mathrm{OCV}(\cdot)$ function). Set 2 and Set 3 of four batteries were used in evaluating the joint state and parameter estimation to see how well the filters perform under different dynamics. The sampling interval T_s=1s. The voltage measurement accuracy is $\pm$5mV and the current measurement accuracy is $\pm$100mA.

The open-circuit voltage as a function of the state-of-charge (SoC) for three sets of the Li-Ion battery cells is plotted in Fig. 1. First, the cell was fully charged to 4.2 V with a constant current. Then, the cell was discharged at at a constant rate until 3.0V. The cell voltage as a function of the SoC under discharge and under charge were averaged to compute the OCV. This has the effect of eliminating, to the greatest extent, the presence of hysteresis and Ohmic resistance in the OCV function. In Set 1, the batteries were put in a chamber with controlled temperature of 25°C. In Set 2, the batteries were put on an open table with an electronic fan turned on. In Set 3, the batteries were put on an open table with the fan turned off. We can see that Set 2 has a closer empirical OCV function to Set 1 than Set 3 to Set 1.

The real test comprised a sequence of 40 charge and discharge cycles for each battery in three sets. The battery was connected to a potentiometer load, separated by 2A discharge pulses and 10-min rests, and spread over the 20%-90% SoC range. Set 1 was used to estimate the OCV function and calibrate the 4th order model of $A_{\mathbf{z}}$. Sets 2 and 3 were used to evaluate the SoC estimation accuracy with the model state and parameters initialized from the same conditions as in Set 1. We compare the adaptive cubature Kalman filter (CKF) (Chen, 2012) with the dual extended Kalman filter (EKF) (Plett, 2004b) in terms of the root mean square (RMS) error and maximum error over the whole duration with approximately 5000s for each cycle. The adaptive CKF and dual EKF have the same initial condition. In adaptive CKF, the forgetting factor sequence was chosen by

$$\lambda_k = 1 - 0.05 \cdot 0.95^k.$$

Table 1. Comparison of SoC Estimation Accuracy

Method	Test Set	RMS error (%)	Maximum error (%)
dual EKF	1	0.32	1.33
adaptive CKF	1	0.29	1.34
dual EKF	2	2.14	9.8
adaptive CKF	2	1.33	3.2
dual EKF	3	7.42	13.3
adaptive CKF	3	2.53	5.2

The SoC estimation results are listed in Tab. 1. The dual EKF has similar SoC estimation accuracy to the adaptive CKF on Set 1 batteries since the model parameters have been well calibrated. The slight increase of the error by the dual EKF is mainly due to linearization of the dynamic model. It is interesting to note that the adaptive CKF performs much better in Set 2 and Set 3 where the OCV functions and unknown parameters are different from those in Set 1. In Set 3, η has more than 20% of variation from the nominal value in Set 1. The dual EKF yields more than twice of the SoC estimation error made by the adaptive CKF.

Next, we apply the SoC estimation to estimate the end-of-life (EOL) and remaining useful life (RUL) of the battery during a discharge cycle based on the proposed conformal interval estimation using a Gaussian kernel. For convenience, we chose $\alpha = 0.99$ and calculated the true t_{EOF} when the cell voltage dropped below 3.0V. The adaptive CKF was used to jointly estimate the state and parameter and the resulting prediction interval of the EOL is shown in Fig. 2. We can see that more than 99% of the predicted intervals cover the true EOL, indicating the validity of the prediction. Note that the predicted EOL interval does not always decrease over time owing to the future loading uncertainty as well as the evolvement of the dynamic state. Note also that the lower and upper limit of the prediction EOL may not be symmetric around the best point estimate, indicating possibly an asymmetric posterior distribution. Nevertheless, the predicted interval reduces to less than 20s fairly quickly. Fig. 3 shows the RUL prediction interval from 100s to 700s with $\alpha = 0.99$. We can see that the prediction interval covers the true RUL nearly all the time with the upper limit being close to the true RUL. Thus the

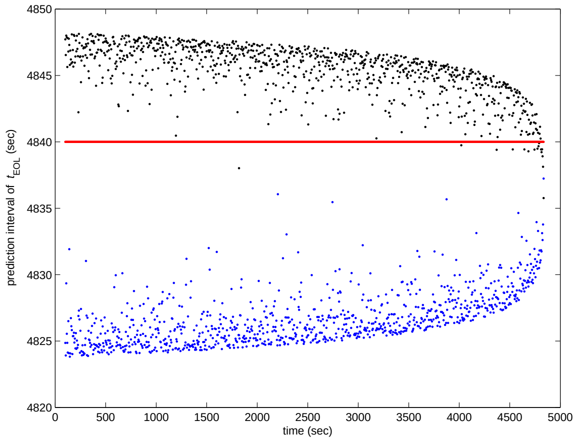


Figure 2. The end-of-life (EOL) prediction interval with $\alpha = 0.99$ using CKF for one discharge cycle of Set 2.

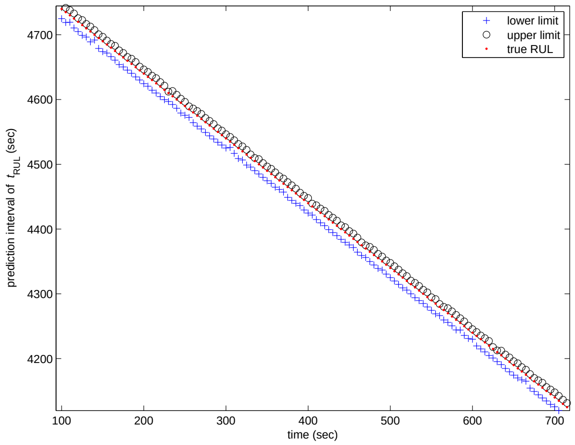


Figure 3. The remaining useful life (RUL) prediction interval with $\alpha = 0.99$ using CKF for one discharge cycle of Set 2.

algorithm seems to be practical for monitoring the battery's SoC and fully utilizing its capacity.

5. Conclusion

We presented a generic framework for distribution free interval estimation to quantify the uncertainty of the end-of-life or the remaining useful life (RUL) prediction of a system component. The method combines the conformal prediction and non-parametric density estimation to ensure the finite sample validity with arbitrarily chosen confidence level. Under certain regularity conditions, the proposed interval estimator converges to an oracle band at a minimax optimal rate. In addition, we used a data driven method to automatically select the bandwidth in the kernel density estimation and worked out a practical approximation to speed up the computation. The proposed method was used to predict the RUL interval for Li-Ion batteries with the joint state and parameter estimation using nonlinear filtering methods. The results reveal that the 99% confidence interval shrinks quickly when the dynamic model captures the discharge cycle fairly accurately. In addition, the lower and upper limit of the RUL prediction interval is often non-symmetric of the true RUL indicating the asymmetric nature of the posterior distribution.

Acknowledgment

This work was supported in part by NASA/LEQSF(2013-15)-Phase3-06 through grant NNX13AD29A and Louisiana Board of Regents through LEQSF-EPS(2012)-OPT-IN-12.

References

Charkhgard, M., & Farrokhi, M. (2010). State-of-charge estimation for lithium-ion batteries using neural networks and ekf. *Industrial Electronics, IEEE Transactions on*, *57*(12), 4178–4187.

Chen, H. (2012). Adaptive cubature kalman filter for nonlinear state and parameter estimation. In *Information fusion (fusion), 15th international conference on* (pp. 1413–1420).

Chiasson, J., & Vairamohan, B. (2005). Estimating the state of charge of a battery. *Control Systems Technology, IEEE Transactions on*, *13*(3), 465–470.

Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap* (Vol. 57). CRC press.

Fisher, R. (1954). *Statistical methods for research workers.* Oliver and Boyd.

Justel, A., Pena, D., & Zamar, R. (1997). A multivariate kolmogorov-smirnov test of goodness of fit. *Statistics & Probability Letters*, *35*(3), 251–259.

Kim, J., & Cho, B. H. (2011). State-of-charge estimation and state-of-health prediction of a li-ion degraded battery based on an ekf combined with a per-unit system. *Vehicular Technology, IEEE Transactions on*, *60*(9), 4249–4260.

Klein, R., Chaturvedi, N. A., Christensen, J., Ahmed, J., R., F., & Kojic, A. (2013). Electrochemical model based observer design for a lithium-ion battery. *Control Systems Technology, IEEE Transactions on*, *21*(2), 289–301.

Lei, J., Robins, J., & Wasserman, L. (2011). Efficient nonparametric conformal prediction regions. *Manuscript. http://arxiv.org/abs/1111.1418.*.

Luo, J., Namburu, M., Pattipati, K., Liu, Q., Kawamoto, M., & Chigusa, S. (2003). Model-based prognostic techniques. In *Ieee systems readiness technology conference* (pp. 330–340).

Plett, G. L. (2004a). Extended kalman filtering for battery management systems of lipb-based hev battery packs part 2 – modeling and identification. *Journal of Power Sources*, *134*(2), 262–276.

Plett, G. L. (2004b). Extended kalman filtering for battery management systems of lipb-based hev battery packs part 3 – state and parameter estimation. *Journal of Power Sources*, *134*(2), 277–292.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, *27*(3), 832–837.

Saha, B., & Goebel, K. (2011). Model adaptation for prognostics in a particle filtering framework. *International Journal of Prognostics and Health Management*, *2*(006).

Saha, B., Goebel, K., & Christophersen, J. (2009). Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement & Control*, *31*(3-4), 293–308.

Sankararaman, S., & Goebel, K. (2013). Uncertainty quantification in remaining useful life of aerospace components using state space models and inverse form. In *Aiaa/asme/asce/ahs/asc structures, structural dynamics and materials conference.*

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009). On applying the prognostic performance metrics. In *Annual conference of the prognostics and health management society.*

Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. *Journal of Machine Learning Research*, *9*, 371–421.

Tsybakov, A. (2009). *Introduction to nonparametric estimation*. Springer.

BIOGRAPHIES

Huimin Chen received the B.E. and M.E. degrees from Department of Automation, Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, in 2002, all in electrical engineering. He was a post doctorate research associate at Physics and Astronomy Department, University of California, Los Angeles, and a visiting researcher with the Department of Electrical and Computer Engineering, Carnegie Mellon University from July 2002 where his research focus was on weak signal detection for single electron spin microscopy. He joined the Department of Electrical Engineering, University of New Orleans in Jan. 2003 and is currently a Don E. Wilson Chevron Associate Professor. His research interests are in general areas of signal processing, estimation theory, and information theory with applications to target detection and target tracking.

Distilling the Verification Process for Prognostics Algorithms

Indranil Roychoudhury[1], Abhinav Saxena[1], Jose R. Celaya[1], and Kai Goebel[2]

[1] *Stinger Ghaffarian Technlogies, Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA*

{*indranil.roychoudhury, abhinav.saxena, jose.r.celaya*}*@nasa.gov*

[2] *NASA Ames Research Center, Moffett Field, CA 94035, USA*

kai.goebel@nasa.gov

Abstract

The goal of prognostics and health management (PHM) systems is to ensure system safety, and reduce downtime and maintenance costs. It is important that a PHM system is verified and validated before it can be successfully deployed. Prognostics algorithms are integral parts of PHM systems. This paper investigates a systematic process of verification of such prognostics algorithms. To this end, first, this paper distinguishes between technology maturation and product development. Then, the paper describes the verification process for a prognostics algorithm as it moves up to higher maturity levels. This process is shown to be an iterative process where verification activities are interleaved with validation activities at each maturation level. In this work, we adopt the concept of technology readiness levels (TRLs) to represent the different maturity levels of a prognostics algorithm. It is shown that at each TRL, the verification of a prognostics algorithm depends on verifying the different components of the algorithm according to the requirements laid out by the PHM system that adopts this prognostics algorithm. Finally, using simplified examples, the systematic process for verifying a prognostics algorithm is demonstrated as the prognostics algorithm moves up TRLs.

1. Introduction

Prognostics and health management (PHM) systems are important to ensure safe and correct operation of real-world engineered systems, reduce their downtime, and reduce maintenance costs. Integral components of PHM systems include diagnostics and prognostics algorithms, the associated diagnostics and prognostics models, sensors, and other hardware, and interfaces between these different components. Diagnostics algorithms involve fault detection, isolation, and identification capabilities; and contribute towards system safety by enabling fault mitigation steps. prognostics algorithms involve prediction of how the system will evolve in the future, thereby contributing towards system safety. Prognostics algorithms also enable reduction of downtime and maintenance costs by providing decision makers with predictions of future system behavior so that decision makers can use this information to either take preventative, fault mitigating, or maintenance actions, or modify mission operations to prolong system use, and maximize mission utility.

Before a PHM system can be deployed in real-world scenarios, it is critical that the PHM system undergoes verification and validation. At the most general level, *verification* of a product is the process where stakeholders answer the query "are we building it right?", while *validation* of a product is the process where stakeholders answer the query "are we building the right thing?" Intuitively, verification is the quality control process of evaluating whether or not a product, service, or system complies with testable constraints imposed by requirements at the start of the development process. In contrast, validation is the quality assurance process of evaluating whether or not a product, service, or a system accomplishes its intended function when fielded in the target application domain.

A PHM system may include several hardware and software components, including software implementations of diagnostics and prognostics algorithms. While many publications discuss the verification of hardware (Gupta, 1993; McMillan, 2000) and software verification (Bérard et al., 2010; Wallace & Fujii, 1989) only, this paper focuses on the verification of all the different components that constitute prognostics algorithms. To this end, first, this paper distinguishes between technology maturation and product development contexts to

Indranil Roychoudhury et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

characterize various PHM verification and validation scenarios often discussed in the literature, and then, proposes a process that identifies specific steps that can facilitate verification of prognostics algorithms. Specifically, the contributions of this paper are as follows:

1. This paper describes the verification process for a prognostics algorithm as it moves up to higher maturity levels. In this work, the concept of technology readiness levels (TRLs) is adopted to represent the different maturity levels of a prognostics algorithm.
2. Next, it is shown that at each TRL, the verification of a prognostics algorithm depends on verifying the different components of the algorithm according to the requirements laid out by the PHM system that adopts this prognostics algorithm.
3. Finally, using simplified examples, the systematic process for verifying a prognostics algorithm is demonstrated as the prognostics algorithm moves up TRLs.

2. Verification and Validation of What - A Product or a Technology?

In order to put our proposed view of the maturation process into context, first we distinguish between developing a system or a *product*[1] versus maturing a *technology*[2]. The development of a system/product is driven by the high level need to accomplish a certain goal in a specific application, whereas technology is understood to be more general and applicable to more than one system when matured.

Examples of systems or products include PHM systems, such as a health and usage monitoring system (HUMS) (Romero, Summers, & Cronkhite, 1996), battery health management system (BHMS) for an electric unmanned aerial vehicle (e-UAV) (Saha et al., 2011), health management system for a water recycling system (WRS) (Roychoudhury, Hafiychuk, & Goebel, 2013), and so on. As shown in Figure 1, a PHM system generally consists of several components, such as sensors (including data acquisition (DAQ), signal conditioner, etc.), technologies such as diagnostics and prognostics algorithms, diagnostics and prognostics models, and other hardware (e.g. communication channels, decision making, interfaces, data storage, and displays, among others). Some of these components, such as sensors, DAQ, etc., are often already matured technologies used in commercial off-the-shelf (COTS) products while others such as prognostics algorithms may be viewed as technologies that need to be matured before they can be used in the PHM systems.

An example of a prognostics algorithm or technology is the `ComputeRUL` algorithm, whose flowchart is shown in Figure 2. `ComputeRUL` consists of three main functions: (*i*) current state estimation, (*ii*) future state prediction, and (*iii*) remaining useful life (RUL) computation. The current state estimation function takes as inputs the sensor readings and the system input data and estimates the current state of the system using a particle filtering scheme (Arulampalam, Maskell, Gordon, & Clapp, 2002) that uses a prognostics model of the system. The future state prediction function takes, as inputs, estimated future operational and environmental profiles and uses a Monte Carlo technique (Kalos & Whitlock, 2008) to predict future system state using the prognostics model. Finally, the RUL computation function compares the predicted values of system state to a predefined threshold and computes RUL as the time remaining before the predicted system state values cross this threshold (Daigle & Goebel, 2011).

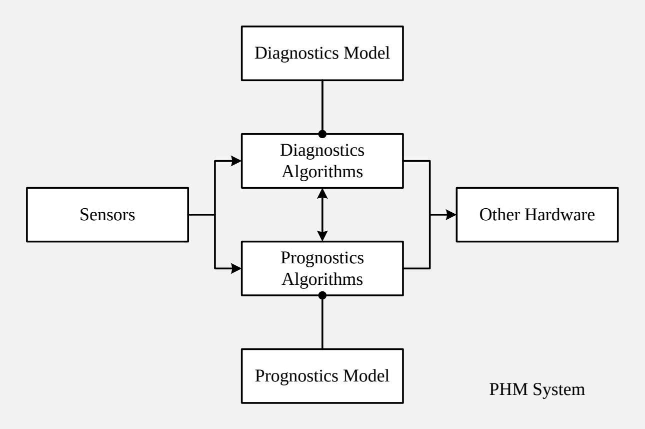


Figure 1. Typical components of a PHM System.

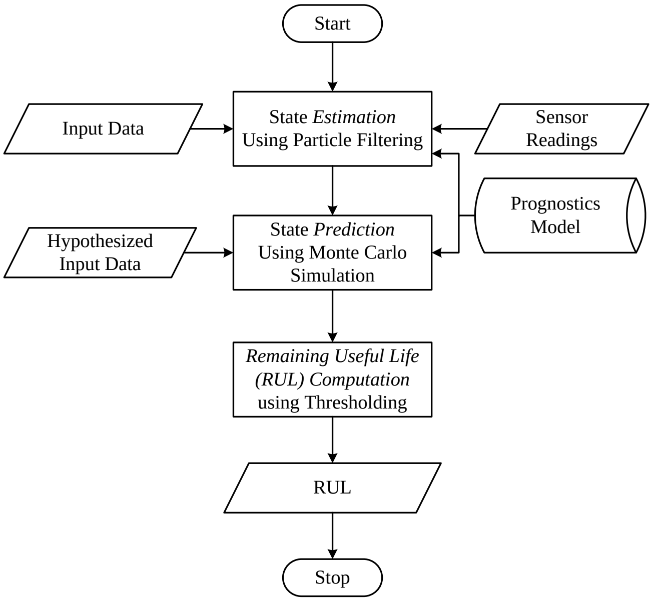

Start
Input Data
State *Estimation* Using Particle Filtering
Sensor Readings
Prognostics Model
Hypothesized Input Data
State *Prediction* Using Monte Carlo Simulation
Remaining Useful Life (RUL) Computation using Thresholding
RUL
Stop

Figure 2. Flowchart of `ComputeRUL`, an example prognostics algorithm.

Verification and validation are key steps in maturing both products and technologies; however the specifics for each

[1] In this paper, we use the terms ‘system’ and ‘product’ interchangeably.
[2] We use the terms ‘algorithm’ and ‘technology’ interchangeably.

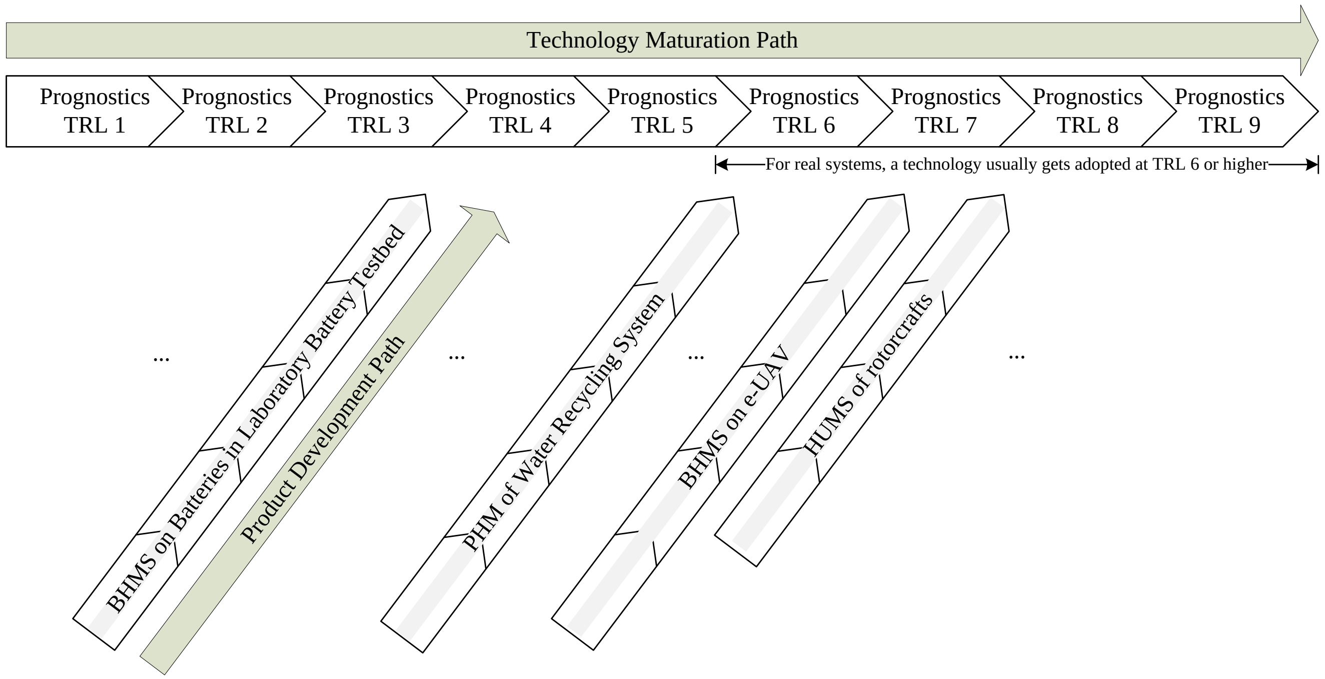


Figure 3. The technology maturation and product development paths.

case may differ. It is important to understand these differences in order to clearly identify what involves verification and validation of a prognostics technology. In the PHM literature we identified that there are several efforts that report verification and validation activities in their respective applications or products, however it was very confusing to get a consistent understanding of what activities are geared towards verification, and what activities enable validation, separately (Tang et al., 2007; Feather & Markosian, 2008). Many efforts combine verification and validation as one task (Aguilar, Luu, Santi, & Sowers, 2005; Byington, Roemer, Kalgren, & Vachtsevanos, 2005), while others use similar methods but sometime refer to them by verification and sometime by validation. Further, most of these reported developments represent different levels of technical maturity, or in other cases, different levels of system integration. Here we attempt to describe a systematic process that allows us to put most of these efforts into a common context and clearly identify the nature of distinct verification and validation activities.

But first, we distinguish between two related but different contexts that influence the nature of verification and validation activities but often get confused with each other, i.e. *product development* versus *technology maturation* (Hicks, Larsson, Culley, & Larsson, 2009). While the steps for both activities look similar there are some differences that are important to understand verification and validation for PHM system development versus for prognostics technology maturation.

Product development typically starts from a top level need for a product (such as a PHM system) for which several ideas may be evaluated at the concept stage. Based on a selection process some ideas move forward with development. At that point a system gets broken into its subsystems and components and requirements *flow down* (Saxena et al., 2012) for individual component development and system integration. At the lowest level some of these components may already exist as COTS components based on mature technologies. But if there are gaps identified, new components may be developed using new technologies. It must be noted that in this scenario the new technology is developed and matured with an end product in mind and, therefore, most of the testing is driven by the requirements flowed down from the top level product. Once developed, these components are first tested individually (quality control process) and then integrated into a subsystem, which undergoes quality control again at the integrated level. At each level tests are designed to help fulfill higher level requirements. This process continues iteratively until the entire system has been integrated and tested as a whole. The product can then be further certified for specific use by domain-relevant certifying agencies.

The *technology maturation*, on the other hand, typically starts at the very low level where a technological concept is considered potentially useful. Prototypes and simulations are developed and tested on simpler cases. Feedback is used to refine the implementation and retest. It is desirable to apply and demonstrate the technology to a variety of applications for establishing its generality. This is often accomplished by proof-of-concept developments for various use cases. Note that each of these proof-of-concept use cases can be considered as a *product* with its own product development cycle, and a successful development of each of these products helps in placing increased trust in the new technology as a whole (see Figure 3), consequently also increasing the maturity level. Conversely, the technology can also be matured without any specific product in mind, or rather with several potential products in mind. As a technology matures through demonstrations and testing it may be adopted for a specific use case for which a directed and dedicated product development cycle is usually followed. Specific verification and validation activities, may be pursued in order to integrate this technology into that product. Note that although the technology at this point can be claimed as matured to be used in that particular product, the generality of technology may allow it to be usable for other products, often with required customization.

2.1. Technology Maturation

We realize that from our research-perspective, maturation of a prognostics algorithm as a technology falls under the general technology maturation category. There are several efforts, currently undergoing, to integrate prognostics algorithms into specific PHM system products such as BHMS for an e-UAV, health management of a WRS, and so on. Therefore, from here on we will describe what prognostics technology maturation would look like and what the verification and validation specific steps are for this maturation. In this work, we adopt NASA's Technology Readiness Level (TRL) concept (Mankins, 1995) to describe various maturity levels for a prognostics technology with no particular preference. Other similar concepts can be used just as well. With this understanding, a technology moves up the TRL as it matures, whereas a product moves up a system integration ladder as it gets developed.

NASA TRLs are defined from TRL 1 through 9 (Mankins, 1995). TRL 1 describes a technology at its very concept or first level of maturation, where only basic principles are observed and reported. TRL 2 describes the stage when a technology concept and/or application is formulated. At TRL 3, analytical and experimental critical function and/or characteristic proof of concept of the technology has been performed. Component and/or breadboard validation in laboratory environment is performed in TRL 4. TRL 5 represents the stage when the component and/or breadboard validation is performed in a relevant environment, while TRL 6 indicates the maturation stage when the system/subsystem model or prototype demonstration is performed in a relevant environment (either ground or space). When the technology reaches TRL 7, the system prototype demonstrations are performed in a space environment. TRL 8 represents the stage when the actual system gets completed and flight qualified through test and demonstration (either ground or space), and finally, TRL 9 represents the stage when the actual system is 'flight proven' through successful mission operations. Note that while an OEM component has reached TRL 8 or 9, it may be integrated into a larger system (product) which is at a lower integration level, i.e. not a full system on its own.

From the technology's point of view, Figure 4 illustrates that at each TRL of a technology, such as prognostics, both verification *and* validation activities must be performed. It is expected that at low TRLs (TRL 1-2), more effort would be on validation of the concepts than verification because the goal at these TRLs is to ensure that the prognostics technology is indeed useful in accomplishing system level performance, safety, and cost goals. In these stages, the technology is still being developed and is adopted in less mature prototypes and products. At middle TRLs (TRL 3-7), more effort is expected on verification activities than validation, since at these TRLs, the emphasis is on adopting and implementing a particular prognostics technology (already verified and validated at lower TRLs) in different PHM systems (at different maturation levels). At high TRLs (TRL 8-9), relatively more effort is again on validation than verification, since by now it is established that the implementations of the prognostics technology (in middle TRLs) are verified and validated to be 'working', and the emphasis at higher TRLs is to ensure that the intended functions of the target PHM system that adopts this prognostics algorithm is fulfilled successfully. As we can see in the above description, it is clear that verification and validation of a prognostics technology at any TRL assumes completion of verification and validation at previous TRLs.

Figure 4 also points out that the scope of the products (e.g. PHM systems) that adopt this technology gets more focused as the prognostics technology development proceeds from low TRLs to high TRLs, e.g., from less-mature PHM systems implemented on breadboards to mature BHMS for the particular Lithium-Ion batteries used in the e-UAV. Moreover, as the prognostics technology matures to higher TRLs, they get integrated into PHM systems that are part of progressively larger systems.

2.2. Product Development

From a products point of view, verification and validation steps are performed for the product (PHM system in our case) by verifying and validating each of its components, the interfaces between these components, and their interactions. The individual components of a product, however, follow their own maturation cycle and integrate into the main product life-cycle when they have matured to a certain degree within their own maturation scale. For example, the prognostics algorithm, like other components of the PHM system, follows its own maturation (TRL development) stages and gets integrated into a product when a minimum TRL is achieved. Typ-

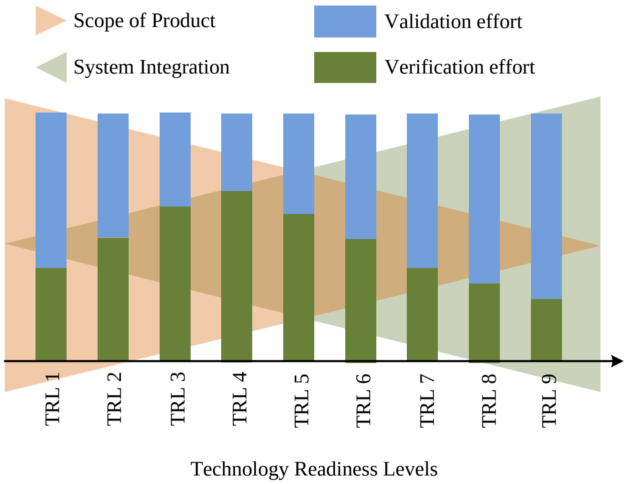


Figure 4. Verification and validation activities across different TRLs.

ically technologies demonstrated to be at TRL 6 or higher are considered a candidate to be integrated into a product ready for use. Once integrated into a product, the prognostics algorithm gets verified and validated together with other integrated components, as a subsystem for that product. Note that other components of the PHM system, e.g. original equipment manufacturer (OEM) components such as sensors may be at very high maturation level, but the product, i.e., the PHM system, as a whole may not be considered fully mature.

In the remainder of this paper, we focus our attention specifically on the verification of the prognostics algorithms, which is the main topic in this paper. The validation of prognostics algorithms is beyond the scope of this work, and will be investigated as part of future work. However, for the sake of highlighting how both verification and validation activities are performed at different TRLs (shown in Figure 4), we will describe some specific validation activities while discussing the case study (in Section 4). The illustrated examples of validation activities also help in drawing a contrast with specific verification activities at various TRLs, especially since, in literature, these validation activities are often included as verification that leads to confusion.

3. Verification of Prognostics Algorithms

Before we describe the verification process for prognostics algorithms, we first have to define what constitutes a prognostics algorithm. Figure 5 shows the different components of a prognostics algorithm when adopted by a product, such as a PHM system. The components of a prognostics algorithm, according to our understanding, are:

- The core prognostics algorithm (CPA) is a high-level abstraction of the prognostics approach which can be represented in terms of a system block diagram, a flowchart, or pseudocode. It is not implemented code. Figure 2 presents an example of the CPA for the `ComputeRUL` prognostics algorithm.
- The implementation specific aspects (ISA) relate to a particular implementation of the core prognostics algorithm (denoted by CPA) in a particular coding language and a particular computational processing architecture and hardware.
- The domain specific entities (DSEs) of a prognostics algorithm when the prognostics algorithm gets adopted in a particular product. The DSE will typically include domain-specific models. We note that every diagnostics and prognostics algorithm is based on a corresponding underlying model. For instance, in classical model-based prognostics algorithms, the models may be state-space models or some other mathematical construct or abstraction that represent or describe physical behavior of the system under consideration. These models can be built upon the use of physics first principles or empirically by observing the physical behavior. For data-driven prognostics algorithms, DSEs consists of domain specific feature extraction methods along with structures for different mathematical abstractions. These abstractions are typically built by observing and extracting the information available in the data often without explicit use of physical phenomena knowledge. As a result, in the data-driven prognostics context, features and abstractions as part of the DSEs are typically equivalent to the concept of features and models in statistical learning. For example, in case of data driven diagnostics and prognostics algorithms, mathematical constructs such as Artificial Neural Network (Yegnanarayana, 2004), Gaussian Process Regression (GPR) (Seo, Wallat, Graepel, & Obermayer, 2000), etc. are trained using data by learning parameters and fixing a structure (topology, covariance structure, etc.) to develop models that can be regressed to make predictions.
- The data sources (DS) consist of sensor measurements of physical variables. These data are typically assumed to be part of a modern instrumentation system in which a transducer is used to measure a physical quantity and its output is processed through a signal conditioner and DAQ in order to obtain a digital representation of such measurement that can be logged for future usage, or use immediately by the algorithm. There are cases in which a physical quantity is not directly measured by a physical sensor but it is estimated from other physical sensors.

By making distinctions between the different components, we identify separate pieces of a prognostics technology that can be verified in parts. This decomposition, to our understanding, makes it easier to verify if something changes, since, in this way, whoever makes a change to a particular component

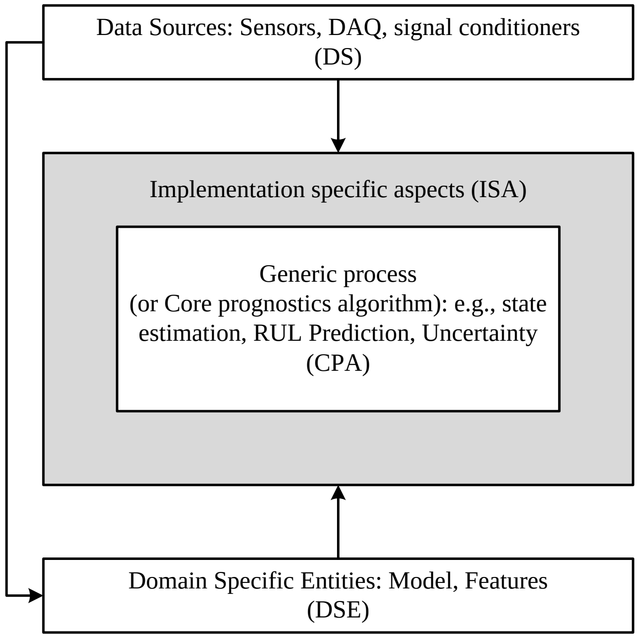

Data Sources: Sensors, DAQ, signal conditioners
(DS)
Implementation specific aspects (ISA)
Generic process
(or Core prognostics algorithm): e.g., state estimation, RUL Prediction, Uncertainty
(CPA)
Domain Specific Entities: Model, Features
(DSE)

Figure 5. Components of a prognostics algorithm adopted by a PHM System.

must also verify that (updated) component at that level. Typically, low TRL technologies get developed by different contractors and do not get adopted in a product until verified and validated at that level. So from a high TRL, system integrators' point of view, they are not tasked with the verification of low-level components. But, if a component changes at a high TRL, the component will have to undergo verification at that high TRL again.

As an example, consider the `ComputeRUL` algorithm whose CPA is described using the flowchart in Figure 2. It is possible that ISA, DSE, and DS may all change at different maturation levels of `ComputeRUL`, but the CPA may remain the same. The ISA, DSE, and DS change because through the course of maturation of the CPA, several systems or products employing such technology are going to be developed at different points in time and likely by different parties with different target application domains.

For example, at TRL 4, the ISA could be in Matlab on a laboratory computer, while at a higher TRL (say, TRL 8), perhaps the ISA would be an assembly language implementation running on embedded processors onboard the e-UAV. Similarly, at TRL 3, DS could be a simulated data, while at TRL 8, the DS could be the actual system sensors onboard the e-UAV as its data sources. Similarly the DSE for TRLs 3 and 8 could be the model of a generic COTS battery cell, and a high fidelity model of the specific Lithium-Ion battery used on board the e-UAV, respectively.

At each TRL, the verification of the prognostics technology implies the verification of ISA with respect to requirements defined using the corresponding DSE and DS. As the prognostics technology moves from one TRL to the next, if any of the DSE, DS, or ISA of the prognostics algorithm at the higher TRL differs from those at the lower TRL, all three components need to be verified again at the higher TRL. Typically, at higher TRLs, the ISA and corresponding DSE may be the same from one TRL to the other, but the DS usually changes.

Recall that verification is the quality control process of evaluating whether or not a product, service, or system complies with testable constraints imposed by *requirements* at the start of the development process. Therefore, requirements play an integral part in verification efforts, and, verification, in a way, can only be as good as the requirements (Saxena et al., 2012; Rajamani et al., 2013; Saxena, Roychoudhury, Lin, & Goebel, 2013). Several publications list various attributes that characterize the goodness of individual requirements, as well as the set of requirements (Firesmith, 2003; Sommerville & Sawyer, 1997). For the sake of brevity, we will discuss here only the most important of these attributes good requirements must fulfill to enable verification:

1. Each requirement must be *verifiable*, i.e., a finite, cost-effective process has been defined to check that the requirement has been attained.
2. Each requirement must be *attainable* (or, *feasible*), i.e, solutions exist within performance, cost, and schedule constraints and the requirement can be satisfied within the constraints of the project.
3. Each requirement must be *unambiguous* (or, *understandable*), i.e., it expresses objective facts, not subjective opinions, and it is subject to one and only one interpretation.
4. Each requirement must be *design independent*, i.e., each requirement does not specify a particular solution or a portion of a particular solution. Stating implementation instead of requirements can lead to major issues, such as forcing a design where it is not intended, or leading the authors of these requirements to believe that all requirements are covered.
5. Each requirement must be *traceable* to an originating high-level requirement. Traceability refers to relationships between parent and child requirements, and between requirements and other design goals. Every requirement should be traceable to the needs, goals, objectives, and constraints of the target application.
6. The set of requirements must be *complete*, i.e., everything the system is required to do throughout the systems life cycle is included. Completeness is a desired property but cannot be proven at the time of requirements development, or perhaps ever.

The property of *traceability* is very important. This is because, a PHM system by design is almost always a part of a larger target system, and typically, PHM system requirements are derived from high level performance, cost, and schedule requirements of these target systems (Saxena et al., 2012). Such high level requirements are typically generated by the customer (the stakeholder who concerns with getting the system built), and often times, the vendor (the stakeholder who concerns with building the system to the customer's satisfaction) must flow down the requirements from the high level customer-requirements to low-level testable requirements.

As part of previous work, in (Saxena et al., 2012), we developed a process to flow down high level functional requirements to low level prognostics performance metrics parameters and illustrated this process using an e-UAV scenario. The low level prognostics metrics take into account several performance factors such as precision, timeliness, accuracy, and prediction confidence, e.g. the α-λ and β metrics developed in (Saxena et al., 2012).

4. Case Study: Verification of ComputeRUL

This section presents a procedure for verification of the prognostics algorithm `ComputeRUL`. As mentioned in Section 3 above, a prognostics algorithm consists of four distinct components, namely CPA, ISA, DS, and DSE based on the product that has adopted the prognostics technology at a particular TRL. For this particular example, as the prognostics algorithm moves to a higher TRL, the CPA is assumed to remain unchanged, although this is not always the case. However, the other three components, i.e., ISA, DS, and DSE, may change as the prognostics algorithm moves to higher TRLs, requiring that the prognostics algorithm is verified again.

Table 1 presents the four components of a prognostics algorithm at different TRLs along with a list of verification and validation testing activities at each TRL. At TRL 1, the prognostics algorithm `ComputeRUL` is in a concept form, and exists as a flowchart (shown in Figure 2). Recall that verification tests involve checking the implementation correctness while validation tests involve checking for functional correctness. Since there is no 'real' implementation, there are no DSE and DS for this algorithm at this TRL, and the testing activities involve evaluation of the applicability of the `ComputeRUL` towards predictive life estimation towards health management. At such a low TRL, therefore, the nature of the testing of this algorithm is more of validation than verification.

In TRL 2, `ComputeRUL` is implemented on paper using the detailed mathematical abstractions for particle filter (Arulampalam et al., 2002) and Monte Carlo methods (Kalos & Whitlock, 2008). The DSE at this stage involves representative nonlinear state-space equations of batteries, and the DS involved 'made-up' synthesized data from general battery dataset. The goal of testing activities at this stage is still more validation-oriented, and involves determining if the `ComputeRUL` algorithm can be applied to battery discharge prediction using current and voltage data. It is also important to study the battery data and ensure that features are available that correlate monotonically to measure fault growth in batteries.

At TRL 3, `ComputeRUL` is implemented using C++ on a generic computer. The DSE include equations of battery of arbitrary chemistry, and the DS used involve damage progression battery data obtained from simulations. The test activities at this stage include both verification and validation activities. The verification activities involve ensuring that uncertainty quantification error, modeling and discretization errors are within allowed limits. Validation activities involve ensuring that α-λ performance, prediction horizon, convergence, confidence interval, statistical hypothesis testing, reliability metric etc. are within allowed limits.

At TRL 4, `ComputeRUL` is implemented in C++ on the computer in the battery testbed in the laboratory. The DSE includes equations of Lithium-Ion batteries similar to those on-board the e-UAV. The DS at this TRL involves data from Lithium-Ion batteries in the environmental chamber (in laboratory setting) with constant load profiles. Both verification and validation activities make up the test activities at this TRL. The verification activities involve ensuring that measurement errors are within allowed limits; the algorithm works correctly in the presence of manufacturing variability; the channel biases are kept at a minimum; and that the algorithm works for constant load profiles. Validation activities involve ensuring that α-λ performance, prediction horizon, convergence, confidence interval, statistical hypothesis testing, reliability metric etc. are within allowed limits.

At TRL 5, the ISA and DSE of `ComputeRUL` remains the same as in TRL 4. However the DS now involves data from Lithium-Ion batteries in the environmental chamber (in laboratory setting) with varying load profiles, and hence the prognostics technology will have to be verified and validated again. The verification and validation testing activities at TRL 5 are similar to that of TRL 4.

The ISA for `ComputeRUL` at TRL 6 involves MATLAB implementation of the CPA running on computers similar to those on-board the e-UAV. The DSE include equations of exact type of Lithium-Ion batteries used on-board e-UAV. The DS at this TRL includes played-back data from actual Lithium-Ion batteries onboard the e-UAV from multiple ground tests. Verification tests at this TRL include ensuring that the no coding errors are made; discretization and sampling rate errors are avoided; and no communication errors occur. Validation tests include ensuring that prognostic horizon, computation time, α-λ performance, robustness to system noise, prediction update rate, etc. are within requirements.

At TRL 7, ComputeRUL is implemented in MATLAB running on the actual computers on-board the e-UAV. The DSE are equations of exact type of Lithium-Ion batteries used onboard the e-UAV and the DS consists of real-time data from actual Lithium-Ion battery sensors onboard the e-UAV from multiple flight tests with simplistic (safe) flight profiles.

Since the CPA, ISA and DSE of ComputeRUL does not change from TRL 7 - 9, once the ISA, DSE, and the interfaces are verified in TRL 7, they do not need to be re-verified in TRL 8 and 9. But, since DS changes from real-time data from actual Lithium-Ion batteries onboard the e-UAV from multiple flight tests with complex flight profiles in TRL 8 to real-time data from actual Lithium-Ion batteries onboard the e-UAV from actual science flight missions, validation activities are performed again at both TRL 8 and 9, and involve ensuring that prognostic horizon, computation time, α-λ performance, robustness to system noise, prediction update rate, etc. are still within requirements.

In our case study, we use an example that used BHMS products at each TRL to demonstrate how the prognostics algorithm matures to higher TRLs. But, as is shown in Figure 3, maturation can also be done through different products or systems (e.g., PHM of Water Recycling System, HUMS of rotorcrafts, etc.) with necessary customization and testing.

5. Conclusions and Future Work

In this paper, we presented a systematic process of verification of prognostics algorithms. We distinguished between technology maturation and product development processes, and described the systematic process of verification of a prognostics algorithm as it moves up to higher maturity levels. This process is iterative where verification activities are interleaved with validation activities at each maturation level. It was shown that at each maturation level, verification of a prognostics algorithm depends on verifying the different components of the algorithm according to the requirements laid out by the PHM system that adopts this prognostics algorithm. Finally, using simplified examples (mostly from the battery health management domain), the systematic process for verifying a prognostics algorithm was demonstrated.

In reality, verification and validation of prognostics technology is not trivial. These challenges arise from use of nondeterministic approaches to account for uncertainty in prognostics and the self-evolving nature of these algorithms exhibiting learning behaviors both of which result in an infinite testing space from an exhaustive verification point of view, which is practically impossible to cover. Apart from mathematical or theoretical limitations, prognostics methods suffer from acausality limitations towards their validation as they require ground truth information about actual time of failure for failures that have not happened yet. As part of future work, we will investigate how to address these challenges. We will also investigate the process for validation of PHM systems.

Acknowledgment

The funding for this work was provided by the NASA System-wide Safety and Assurance Technologies (SSAT) Project.

References

Aguilar, R., Luu, C., Santi, L. M., & Sowers, T. S. (2005). Real-time simulation for verification and validation of diagnostic and prognostic algorithms. In *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit* (pp. 1–8).

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.

Bérard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., & Schnoebelen, P. (2010). *Systems and software verification: Model-checking techniques and tools*. Springer Publishing Company, Incorporated.

Byington, C. S., Roemer, M., Kalgren, P., & Vachtsevanos, G. (2005). Verification and validation of diagnostic/prognostic algorithms. In *Machinery Failure Prevention Technology Conference.*

Daigle, M., & Goebel, K. (2011). A model-based prognostics approach applied to pneumatic valves. *International Journal of Prognostics and Health Management*, *2*(2), 008.

Feather, M. S., & Markosian, L. Z. (2008). Towards certification of a space system application of fault detection and isolation. In *Proceedings of the 2008 International Conference on Prognostics and health management* (pp. 6–9).

Firesmith, D. (2003). Specifying good requirements. *Journal of Object Technology*, *2*(4), 77–87.

Gupta, A. (1993). Formal hardware verification methods: A survey. In *Computer-Aided Verification* (pp. 5–92).

Hicks, B., Larsson, A., Culley, S., & Larsson, T. (2009). A methodology for evaluating technology readiness during product development. In *Proceedings of the International Conference on Engineering Design* (pp. 157–168).

Kalos, M. H., & Whitlock, P. A. (2008). *Monte carlo methods*. John Wiley & Sons.

Mankins, J. C. (1995). Technology readiness levels. *White Paper, April*, *6*.

McMillan, K. L. (2000). A methodology for hardware verification using compositional model checking. *Science of Computer Programming*, *37*(1), 279–309.

Rajamani, R., Saxena, A., Kramer, F., Augustine, M., Schroeder, J. B., Goebel, K., ... Lin, W. (2013).

Guidelines for writing ivhm requirements for aerospace systems. In *Proceedings of the SAE 2013 AeroTech Congress & Exhibition.*

Romero, R., Summers, H., & Cronkhite, J. (1996). *Feasibility study of a rotorcraft health and usage monitoring system (hums): Results of operator's evaluation.* (Tech. Rep.). DTIC Document.

Roychoudhury, I., Hafiychuk, V., & Goebel, K. (2013). Model-based diagnosis and prognosis of a water recycling system. In *IEEE Aerospace Conference* (pp. 1–9).

Saha, B., Koshimoto, E., Quach, C. C., Hogge, E. F., Strom, T. H., Hill, B. L., ... Goebel, K. (2011). Battery health management system for electric uavs. In *IEEE Aerospace Conference* (pp. 1–9).

Saxena, A., Roychoudhury, I., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2012). Requirement flowdown for prognostics health management. In *Proceedings of the AIAA Infotech @ Aerospace.*

Saxena, A., Roychoudhury, I., Lin, W., & Goebel, K. (2013). Towards requirements in systems engineering for aerospace ivhm design. In *Proceedings of the AIAA Infotech @ Aerospace.*

Seo, S., Wallat, M., Graepel, T., & Obermayer, K. (2000). Gaussian process regression: Active data selection and test point rejection. In *Mustererkennung 2000* (pp. 27–34). Springer.

Sommerville, I., & Sawyer, P. (1997). *Requirements engineering: A good practices guide*. John Wiley & Sons.

Tang, L., Saxena, A., Orchard, M. E., Kacprzynski, G. J., Vachtsevanos, G., & Patterson-Hine, A. (2007). Simulation-based design and validation of automated contingency management for propulsion systems. In *IEEE Aerospace Conference* (pp. 1–11).

Wallace, D. R., & Fujii, R. U. (1989). Software verification and validation: an overview. *Software, IEEE*, *6*(3), 10–17.

Yegnanarayana, B. (2004). *Artificial neural networks*. PHI Learning Pvt. Ltd.

BIOGRAPHIES

Indranil Roychoudhury received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, Rajasthan, India in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, Tennessee, USA, in 2006 and 2009, respectively. Since August 2009, he has been with SGT, Inc., at NASA Ames Research Center as a Computer Scientist. Dr. Roychoudhury is a member of the Prognostics and Health Management Society and the IEEE. His research interests include hybrid systems modeling, model-based diagnostics and prognostics, distributed diagnostics and prognostics, and Bayesian diagnostics of complex physical systems.

Abhinav Saxena is a Research Scientist with SGT Inc. at the Prognostics Center of Excellence NASA Ames Research Center, Moffett Field CA. His research focus lies in developing and evaluating prognostics algorithms for engineering systems using soft computing techniques. He is a PhD in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta. He earned his B.Tech in 2001 from Indian Institute of Technology (IIT) Delhi, and Masters Degree in 2003 from Georgia Tech. Dr. Saxena has been a GM manufacturing scholar and is also a member of IEEE, Prognostics and Health Management Society, AIAA, and ASME.

Jose R Celaya is a research scientist with SGT Inc. at the Prognostics Center of Excellence, NASA Ames Research Center. He received a Ph.D. degree in Decision Sciences and Engineering Systems in 2008, a M. E. degree in Operations Research and Statistics in 2008, a M. S. degree in Electrical Engineering in 2003, all from Rensselaer Polytechnic Institute, Troy New York; and a B. S. in Cybernetics Engineering in 2001 from CETYS University, Mxico.

Kai Goebel is Deputy Area Lead of the Discovery and Systems Health Technology Area at NASA Ames Research Center. He also coordinates the Prognostics Center of Excellence. Prior to joining NASA in 2006, he was a senior research scientist at General Electric Corporate Research and Development center since 1997. Dr. Goebel received his Ph.D at the University of California at Berkeley in 1996. He has carried out applied research in the areas of real time monitoring, diagnostics, and prognostics and he has fielded numerous applications for aircraft engines, transportation systems, medical systems, and manufacturing systems. He holds 17 patents and has co-authored more than 250 technical papers in the field of IVHM. Dr. Goebel was an adjunct professor of the CS Department at Rensselaer Polytechnic Institute (RPI), Troy, NY, between 1998 and 2005 where he taught classes in Soft Computing and Applied Intelligent Reasoning Systems. He has been the co-advisor of 6 Ph.D. students. Dr. Goebel is a member of several professional societies, including ASME, AAAI, AIAA, IEEE, VDI, SAE, and ISO. He was the General Chair of the Annual Conference of the PHM Society, 2009, has given numerous invited and keynote talks and held many chair positions at the PHM conference and the AAAI Annual meetings series. He is currently member of the board of directors of the PHM Society and associate editor of the International Journal of PHM.

Table 1. Verification of `ComputeRUL` prognostics technology: an example.

TRL	CPA	ISA	DSE	DS	Testing Activities
1	`ComputeRUL`	Flowchart	N/A	N/A	Evaluate applicability of predictive life estimation towards health management **(Validation)**
2	`ComputeRUL`	Mathematically instantiating the different components in CPA and simulating these analytically	Representative nonlinear equations	Synthesized data	1. Determine that battery discharge can be predicted using available current and voltage data. **(Validation)** 2. Verify that features are available that correlate monotonically to measure fault growth in batteries **(Validation)** 3. Quantify errors and confidence in computed features correlated to fault ground truth data **(Validation)**
3	`ComputeRUL`	CPA implemented in C++ on a generic laptop	Equations of battery of any arbitrary chemistry	Damage progression data obtained from simulations	1. Ensure that uncertainty quantification error, modeling error, discretization error are within allowed limits **(Verification)** 2. Ensure that α-λ Performance, prediction horizon, convergence, etc. metrics from the ISA are within allowed limits **(Validation)**
4	`ComputeRUL`	CPA implemented in C++ on the computer in the battery testbed in the laboratory	Equations of battery of Lithium-Ion chemistry similar to those onboard the e-UAV	Data from Lithium-Ion batteries in the environmental chamber (in laboratory setting) with constant load profiles	1. Ensure that measurement errors, manufacturing variability, channel biases, load profiles are all within allowed limits. **(Verification)** 2. Ensure that α-λ Performance, prediction horizon, convergence, confidence interval, statistical hypothesis testing, reliability metric etc. are within allowed limits. **(Validation)**
5	`ComputeRUL`	CPA implemented in C++ on the computer in the battery testbed in the laboratory	Equations of battery of Lithium-Ion chemistry similar to those onboard the e-UAV	Data from Lithium-Ion batteries in the environmental chamber (in laboratory setting) with varying load profiles	1. Ensure that measurement errors, manufacturing variability, channel biases, load profiles are all within allowed limits. **(Verification)** 2. Ensure that α-λ performance, prediction horizon, convergence, confidence interval, statistical hypothesis testing, reliability metric etc. are within allowed limits. **(Validation)**
6	`ComputeRUL`	CPA implemented in MATLAB running on the computers similar to that on-board the e-UAV	Equations of exact type of batteries of Lithium-Ion chemistry on-board the e-UAV	Played-back data from actual Lithium-Ion battery sensors on-board the e-UAV from multiple ground tests	1. Ensure no coding errors; discretization and sampling rate errors; and communication errors occur. **(Verification)** 2. Ensure that prognostic horizon, computation time, α-λ performance, robustness to system noise, prediction update rate, etc. are within requirements. **(Validation)**
7	`ComputeRUL`	CPA implemented in MATLAB running on the actual computers on-board the e-UAV	Equations of exact type of batteries of Lithium-Ion chemistry on-board the e-UAV	Real-time data from actual Lithium-Ion battery sensors on-board the e-UAV from multiple flight tests with simplistic (safe) flight profiles	1. Ensure that communication errors and delays, code verification, race conditions are all within allowed limits. **(Verification)** 2. Ensure that prognostic horizon, computation time, α-λ performance, robustness to system noise, prediction update rate, etc. are within requirements. **(Validation)**
8	`ComputeRUL`	CPA implemented in MATLAB running on the actual computers on-board the e-UAV	Equations of exact type of batteries of Lithium-Ion chemistry on-board the e-UAV	Real-time data from actual Lithium-Ion battery sensors on-board the e-UAV from multiple flight tests with complex flight profiles and different operating conditions	1. Ensure that prognostic horizon, computation time, α-λ performance, robustness to system noise, prediction update rate, etc. are within requirements. **(Validation)**
9	`ComputeRUL`	CPA implemented in MATLAB running on the actual computers on-board the e-UAV	Equations of exact type of batteries of Lithium-Ion chemistry on-board the e-UAV	Real-time data from actual Lithium-Ion battery sensors on-board the e-UAV during multiple actual science missions	1. Ensure that prognostic horizon, computation time, α-λ performance, robustness to system noise, prediction update rate, etc. are within requirements. **(Validation)**

A Method for Anomaly Detection for Non-stationary Vibration Signatures

Renata Klein[1]

[1]*R.K. Diagnostics, P.O.Box 101, Gilon, D. N. Misgav 20103, Israel*
Renata.Klein@rkdiagnostics.co.il

ABSTRACT

Vibration signatures contain information regarding the health status of the machine components. One approach to assess the health of the components is to search systematically for a list of specific failure patterns, based on the physical specifications of the known components (e.g. the physical specifications of the bearings, the gearwheels or the shafts). It is possible to do so, since the manifestation of the possible failures in the vibration signature is known a priory. The problem is that such a list is not comprehensive, and may not cover all possible failures. The manifestation of some failure modes in the vibration signature may be less investigated or even unknown. In addition, when more than one component is malfunctioning, unexpected patterns may be generated. Anomaly detection tackles the more general problem: How can one determine that the vibration signatures indicate abnormal functioning when the specifics of the abnormal functioning or its manifestation in the vibration signatures are not known a priori? In essence, anomaly detection completes the diagnostics of the predefined failure modes. In many complex machines (e.g. turbofan engines), the task of anomaly detection is further complicated by the fact that changes in operating conditions influence the vibration sources and change the frequency and amplitude characteristics of the signals, making them non-stationary. Because of that, joint time-frequency representations of the signals are desired. This is different from other vibration based diagnostic techniques, which are designated for stationary signals, and often focus on either the time domain or the frequency domain.

For the purpose of this article, we will refer as TFR (time-frequency representation) to all 3D representations which employ on one axis either time, or cycles, or RPM, and on the other axis either frequency, or order. The proposed method suggests a solution for anomaly detection by analysis of various TFRs of the vibration signals (primarily the RPM-order domain).

In the first stage, TFRs of healthy machines are used to create a baseline. The TFRs can be obtained using various methods (Wigner-Ville, wavelets, STFT, etc). In the next stage, the distance TFR between the inspected recording and the baseline is computed. In the third stage, the distance TFR is analyzed and the exceptional regions in the TFR are found and characterized. A basic classification of the anomaly type is suggested. The different stages of analysis: creating baselines, computing the distance TFR, identifying the exception regions, are illustrated with actual data.

1. INTRODUCTION

Monitoring of vibrations can be used to detect machine faults, including roller bearing degradation, gearwheels degradation, eccentricity, mechanical looseness, unbalance, misalignment, oil film bearing instabilities, structural resonance, and cracked rotors. In most methods, the detection is based on comparison of vibration levels at specific frequencies to reference or “baseline” values, representing the healthy cases. The specific frequencies used for tracking are defined separately for each failure mode of each component. Detection of all the possible failure modes of a machine implies definition of all the possible failure modes of all components including all the relevant combinations of failure modes such that all the frequencies of interest will be covered. In spite of the fact that many failure modes can be pre-defined with their associated patterns, the definition and listing of all the frequencies of interest is a very complex task, often impossible. In order to complete the diagnostic process when only a part of the frequencies of interest can be predefined, an anomaly detection algorithm is required.

Diagnostics of rotating machinery during regular operation involves in many cases analysis of non-stationary signals. This is because rotating speeds, loads, and environmental conditions vary (in some cases rapidly) with time. Often, even the assumption of quasi-stationarity may not be appropriate. In such an environment, an efficient way to evaluate condition indicators may be based on time-

Renata Klein. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

frequency or time-order representations that reveal the evolution of the spectra with time. The time-frequency or time-order representations (TFR) can be computed using different techniques of signal processing such as Short Time Fourier Transform (STFT), Wavelets decomposition or Wigner-Ville representations (see Polyshchuk et al 2002, Juluri & Swarnamani 2003, Yang & Ren 2004, Bradford 2006, Klein et al 2011).

Usually the TFRs are representations of the vibration signal or its derivatives (synchronic average, envelope, pre-whitened signals, etc.) in the RPM-frequency or RPM-order domains (see Antoni & Randall 2002, Antoni et al 2004, Sawalhi & Randall 2008, Klein et al 2012). TFRs are widely used in scientific and industrial applications for visual inspection of vibrations. The primary problem of the visual inspection is that in complex machinery, the TFR contains a huge amount of information and it is difficult to sort out and focus on the relevant information manually.

Some methods of anomaly detection in TFRs have been proposed using different approaches (see below). In general, a statistical analysis of the spectrogram values or the over threshold values is used for detection of anomalies. This requires a definition of the probability density function (PDF) and an evaluation of the PDF parameters (differently for different zones of the TFR). There were different assumptions regarding the nature of the probability density function for spectrum or spectrogram values; Huillery et al (2008) show that for spectrograms and STFTs, when using Hanning windowing, the χ^2 PDF (central or non-central for deterministic peaks) is adequate for detection of exceptions. Bechhoefer et al (2011) discussed Rayleigh PDFs for spectrum values and Nakagami PDFs for sums of Rayleigh distributed values. Clifton & Tarassenko (2009) showed that the PDF in spectrogram bins is approximately Gamma and that its tail can be described by a Gumbel distribution representing extremum values distribution. Hazan et al (2012 and 2013) proposed Peak Over Threshold (POT) and Frequency Dependent Peak Over Threshold (FDPOT) methods which were based on the assumption that the values exceeding a threshold can be approximated by a Generalized Pareto distribution.

The current paper proposes an automatic procedure for anomaly detection which is adequate for all types of TFRs. The analysis algorithm emphasizes only the exceptions relative to the "baseline" or the reference TFR, allowing effective masking of huge amounts of less relevant information.

The "baseline" is a statistical characterization of the TFRs derived from a set of healthy machines. The exceptions relative to the baseline are then examined to detect relevant regions corresponding to significant anomalies. In the first section of the article, we will describe the statistical characterization stage, or the baseline generation. Then we will show the algorithm for emphasizing the exceptions in the analyzed TFR (relative to the baseline). Next, we will explain the algorithm for automatic detection and classification of the exceptional regions. The algorithm is demonstrated with an example of a seeded test data, in which the presented algorithm was able to detect the fault without using any prior knowledge on the nature of the fault or the physical dimensions of the faulty part.

2. Baseline generation

The baseline is generated from a set of TFRs recorded in a set of healthy machines. In essence, the baseline is a statistical characterization of the data distribution in each cell of the TFR matrix:

$$\mu_{i,j} = \frac{1}{N}\sum_{n=1}^{N} P_{i,j,n} \qquad \sigma_{i,j} = \frac{1}{N}\sqrt{\sum_{n=1}^{N} P_{i,j,n}^2} \tag{1}$$

where: $\mu_{i,j}$ is the average of values in cell i,j, $\sigma_{i,j}$ is the standard deviation of the values in cell i,j, N is the number of TFRs in the baseline, and $P_{i,j,n}$ is the value of the spectrum n in cell i,j.

It is highly advisable to use similar operating conditions for baseline generation. This allows a better representation of the healthy population, hence a higher reliability in detecting anomalies. To illustrate that, let us consider slow acceleration versus fast acceleration in a jet engine. In our experience, the two cases differ significantly in their vibration patterns even at the same RPM. Evidently, loads vary significantly, some of the resonances that are excited during a slow acceleration may not be present at a fast acceleration, and there are also differences in the amplitude of peaks at characteristic frequencies. Combining both cases of fast and slow accelerations in the same baseline model may lead to a significant reduction in discrimination abilities of the condition indicators.

Thus, it is essential to decide which operating conditions can be combined in the same baseline. This can be achieved by a relatively simple statistical hypothesis testing procedure, combined with a physical understanding of the load variations in the different operating conditions.

Various other technical issues should be addressed during the implementation of the baseline algorithm.

First, all the TFRs need to have the same scale. This can be achieved either by interpolation of the existing TFRs to a new common scale, or by calculation of the TFRs using a predefined common scale. The predefined common scale is achieved by calculating the TFRs at predefined ranges of rotating speeds and similar frequency/order resolution.

If interpolation is used, one should be careful not to introduce artifacts to the data when the time scale does not fit the variation rate of the load. For example, when the time resolution or RPM resolution is too low compared to the acceleration rate, and adjacent spectra differ abruptly in amplitude, the interpolated spectrum may generate an erroneous baseline TFR with high variances.

Another issue is how to set a correct scale. A higher resolution in time or RPM will provide better detection capabilities, but setting the resolution too high may leave some time segments of the TFR too short for a reliable spectrum calculation. The scale should be adapted to the operating modes of the inspected machinery so that most of the TFR will be calculated correctly.

3. Distance TFR

When a new data is available, the TFR is interpolated to obtain the same scale as the scale that was used during baseline generation. A new representation, the distance TFR, is calculated, where each cell represents the corresponding distance from the model of healthy machines.

The distance TFR emphasizes the cells that deviate from the distribution of healthy machines (see Figure 1 and Figure 2).

Mahalanobis distance is used for comparison (Eq. 2).

$$D_{i,j} = \frac{P_{i,j} - \mu_{i,j}}{\sigma_{i,j}} \tag{2}$$

where: $D_{i,j}$ is cell *i,j* in the distance TFR, $P_{i,j}$ is the corresponding cell in the TFR of the new data, $\mu_{i,j}$ is the mean of *i,j* cell in the baseline, and $\sigma_{i,j}$ is the baseline standard deviation of the corresponding cell.

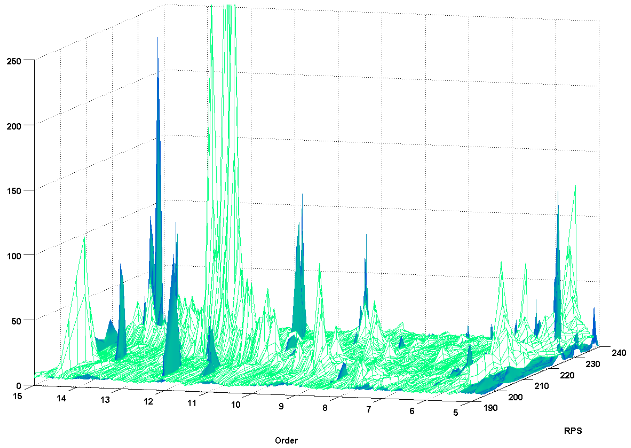


Figure 1. Comparison between baseline and a newly obtained TFR in the RPS-order domain (RPS – Rotation Per Second). The white surface represents the baseline (μ+3σ) and the dark green surface represents the inspected TFR.

The distance TFR represents the distance of the actual TFR from the healthy population in terms/units of standard deviations, i.e. it contains data that is statistically normalized. The way the distance is calculated does not imply a specific probability distribution.

Faults of mechanical components generate specific known vibration patterns such as characteristic frequencies with sidebands due to modulation. Appropriate algorithms allowing diagnostics of components based on TFRs can recognize these patterns automatically (such an algorithm operating on the distance TFR was proposed for detection of faulty bearings in Klein et al 2012).

In other cases where the exceptions do not follow a specific pattern it will not be possible to associate the failure with one of the mechanical components. Nevertheless, automatic diagnostics of abnormal behavior can be performed with good reliability and detection capabilities.

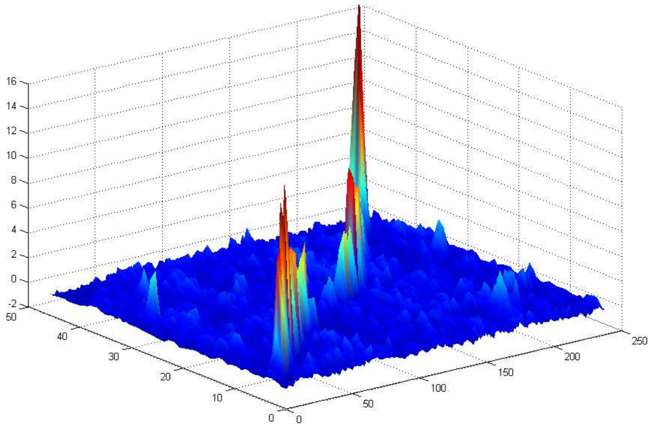


Figure 2. Illustration of the distance TFR

4. Detection of exceptional regions

The goal of the algorithm is to identify continuous regions of exceptional cells. The algorithm flowchart is described schematically in Figure 3.

First a surface defining the threshold for each cell is defined. Then the exceptional cells exceeding the local threshold are found. The exceptional cells are grouped into continuous regions. The number of cells and volume of each exceptional region are calculated and compared to the criteria defined for identification of anomalies.

Searching for over threshold values as an only criterion was found to be insufficient. To avoid false alarms, there was a need to screen out noise phenomena in single cells, and highlight exceptions only if they belong to continuous and sizeable regions. To accomplish that, the algorithm is searching for exceptional regions satisfying the following additional criteria:

- The number of cells N_k, in a continuous region k, should exceed a minimum value – to avoid consideration of spurious peaks.
- The volume V_k ($V_k = \sum_{\forall i,j \in k} D_{i,j}$) of an exceptional region k should exceed a minimum value. The volume represents a measure of the number of cells and their

values. We want to guarantee that at least one of these is large enough to be considered as significant.

- The total volume of exceptions $\sum_k V_k$ should exceed a minimum value. The total volume represents the number of exceptional regions and their mean volume giving the option to define at what level we will consider the TFR as exceptional.

These criteria allow sufficient flexibility to tune the detection algorithm and adapt it to different needs. For example, if we suspect that the distance of peaks (from the baseline of healthy machines) maybe of low amplitude, we may want to set the threshold to a low value (e.g. $T_{i,j} = 3\sigma$), and to compensate it by setting a large threshold for the number of cells in a region and/or the minimum volume of a region.

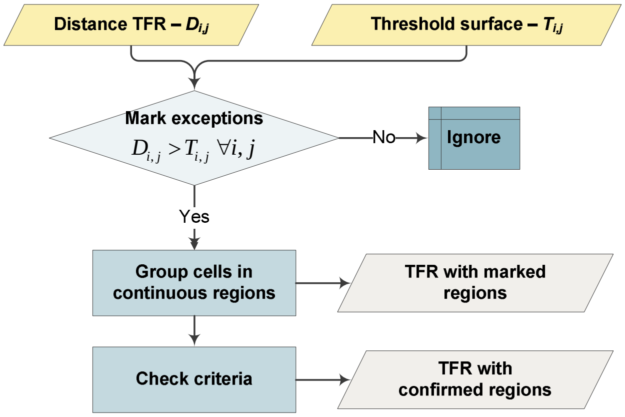

Distance TFR – $D_{i,j}$
Threshold surface – $T_{i,j}$
Mark exceptions
$D_{i,j} > T_{i,j} \ \forall i, j$
No
Ignore
Yes
Group cells in continuous regions
TFR with marked regions
Check criteria
TFR with confirmed regions

Figure 3. Algorithm flowchart

It is important to note that the proposed solution for general anomaly detection should attempt to cover several types of TFRs. For example, when a fault exists in a rotating component like a bearing or a gearwheel, we would expect to detect several exceptional regions, each related to a specific harmonic or a sideband (each region will contain several cells covering different rotating speeds). The exceptional regions can be detected in the order domain, but they can be better emphasized in the order domain of the dephased signal or in the order domain of the envelope (see Antoni & Randall 2002, Antoni et al 2004, Sawalhi & Randall 2008, Klein et al 2009, Bechhoefer et al 2011).

4.1. Definition of the threshold surface

The selection of threshold values is important and influences the reliability of anomaly detection. The threshold can be constant for all the cells of the TFR, defining a plane parallel to the time frequency plane, or can vary defining any positive surface.

Several considerations affect the selection of the threshold surface:

First, it is possible to use the threshold surface for masking out effects of faults discoverable by the direct search algorithms. For example, faults in specific gearwheels are discoverable in some specific frequencies/orders. We may want to set very high threshold values to the corresponding frequencies/orders to mask out these effects.

The second consideration for selection of the threshold surface is the type of the probability distribution function (PDF) of the healthy population belonging to the baseline. It is also possible that the PDF parameters differ from one cell to another. The determination of these parameters for each cell may require a large data set of healthy TFRs.

Because we are using the distance TFR, which is already normalized, the threshold for several types of PDFs can be constant and generic. This is true for Rayleigh, χ^2, and Gamma PDFs.

One last word on threshold selection: Because thresholds are not the sole parameter used (the algorithm also uses the criteria of area and volume, i.e. number of exception cells and accumulated sum of values), the proposed method is relatively tolerant to imperfections in selecting the thresholds. The algorithm was applied on several TFRs of healthy machines, using relatively low thresholds, without triggering false alarms.

4.2. Classification of anomalies

The algorithm for anomaly detection targets faults which are not covered in the direct search algorithms. It can be used as a start point to learn about and define new patterns to search for, thus enlarging the knowledge about faults in a specific machine. The detection of anomalies should be amended with an examination of experts and field feedback on the status of the machine.

The classification of anomalies should allow as much as possible hints on their origin and nature. The hints can be based on the type of TFR in which the anomaly was detected (e.g. TFR of the raw vibration signal, TFR of the synchronous average, TFR of the resampled signal, TFR of the dephased signal, etc.), as well as the range of frequencies or orders.

The simplicity of the algorithm and the fact that only the threshold surface depends on the assumed PDF makes it useful and easy to apply in different TFRs and different configurations.

5. Example of Algorithm Performance

The example is based on data recorded during a seeded fault back to back test in a turbofan engine. The fault was introduced on the outer race of a bearing in which the inner race rotates at a speed proportional to shaft A rotating speed. The example demonstrates that the algorithm was able to detect the fault, without using any prior knowledge on the nature of the fault or the physical dimensions of the faulty bearing.

The presented method was applied to TFRs consisting of RPM-orders spectrograms. The spectrograms used were based on PSDs. The PSDs were calculated in consecutive periods of fixed length, during accelerations and decelerations of a turbofan engine with similar rotating speed gradients.

The presented results were based on spectrograms calculated with similar order resolution and varying RPM resolution. A study of the variations of the spectra levels in the healthy records revealed that the variations of the peak levels did not exceed the random error of the PSDs in bins of 5 Hz. Therefore, the periods for each PSD calculation were defined such that the rotating speed variations will be of maximum 5 Hz and the interpolation of the RPM axis was applied in bins of 3Hz.

The statistics of the baseline were calculated on 28 spectrograms from healthy runs.

Figure 4 shows a part of the RPM-order spectrogram of the vibration signal from a run with the faulty bearing. Some energetic ridges corresponding to the shaft A harmonics and background noise can be observed (the highest ridges can be observed at orders above 35). As well, some harmonics of shaft B rotating speed can be observed.

Figure 5 shows the distance TFR (based on the RPM-order spectrogram and corresponding baseline) calculated on the same data as in Figure 4. A pattern that was not visible in the regular spectrogram becomes evident after distance calculation. The shaft harmonics that are clearly observable at orders above 35 in Figure 4 are not seen in Figure 5. This means that the vibration levels corresponding to both shafts harmonics were close to the baseline of healthy systems and not exceptional.

As one can see from comparison of Figure 4 and Figure 5, the distance TFR is a helpful tool for visual inspection of TFRs. It emphasizes only the suspicious locations and allows a significant reduction of information for manual scan.

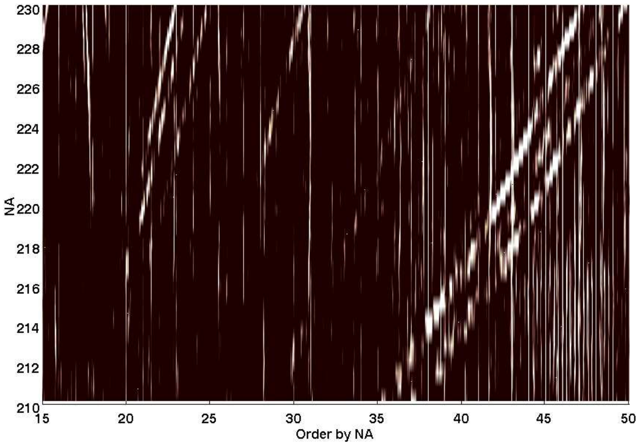


Figure 4. RPM-Order spectrogram

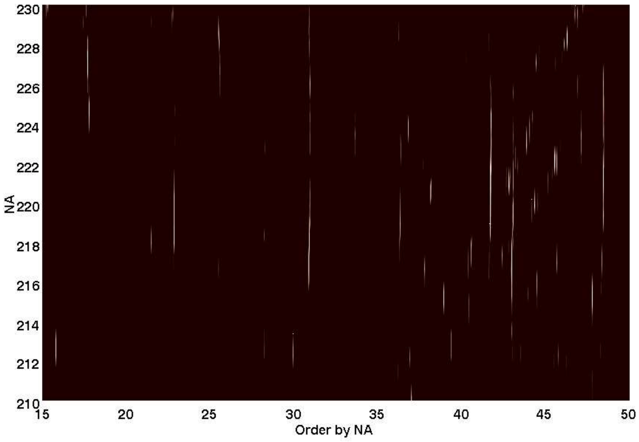


Figure 5. Distance TFR of the RPM-Order spectrogram

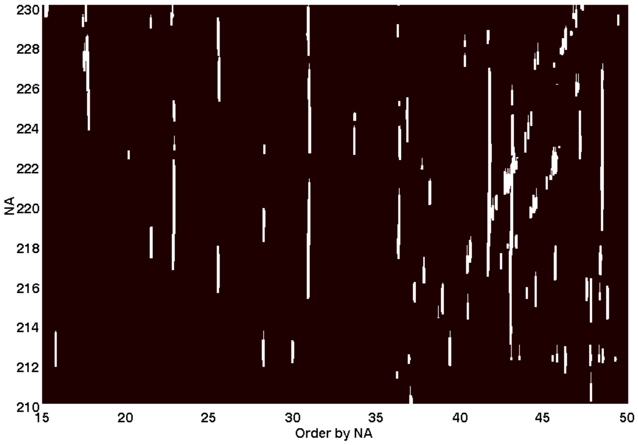


Figure 6. Exceptions found in the distance-TFR of the RPM-Order spectrogram

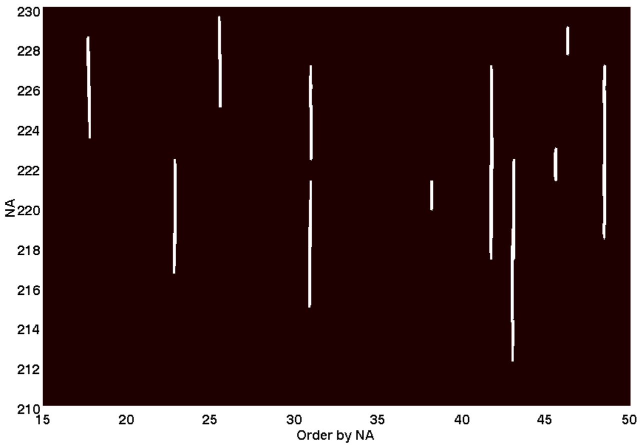


Figure 7. Exceptions confirmed in the distance-TFR of the RPM-Order spectrogram

Figure 6 shows the contours of the exceptional continuous regions found after the comparison with the threshold surface. Figure 7 shows the contours of the confirmed regions after application of all the criteria (i.e. the number of cells and the volume). It can be easily observed that in Figure 7 only the peaks related to the faulty bearing remain, and that their location indicates a very clear pattern that is easy to identify and diagnose.

6. Summary and Conclusions

A method for analysis and diagnosis of non-stationary TFRs of vibro-acoustic data was proposed.

The method can be applied on any type of TFR, regardless of the analyzed signal or the method of the TFR calculation. The key idea of the method is the detection of the exceptional regions in the distance TFR (deviation from the baseline TFR).

The method was demonstrated, using RPM-order spectrograms, for diagnosis of a damaged bearing in a seeded fault test of a turbofan engine (without relying on the specific physical information of the bearing). The method of extracting the exceptional regions was described. It was shown that the proposed method is effective for detection of the abnormal behavior resulting from a faulty bearing.

It seems that the distance TFR is a powerful tool in detecting anomalies and emphasizing abnormal behavior.

The distance TFR proved effective in emphasizing exceptions in a noisy environment, including unknown damage or anomalies of any kind.

Last, the distance TFR can be used efficiently by experts for visual inspection and diagnostics of exceptions. Since the distance TFR emphasizes only the ridges or regions that deviate from the baseline population statistics, masking out the irrelevant data, it is easier and a more focused tool for visual inspection compared to the original time frequency representation.

Abbreviations

RPM	Rotations Per Minute
TFR	Time-Frequency Representation
STFT	Short Time Fourier Transform
PDF	Probability Density Function
PSD	Power Spectral Density
POT	Peak Over Threshold
FDPOT	Frequency Dependent Peak Over Threshold

References

Antoni, J. & Randall, R. B., (2002). Differential Diagnosis of Gear and Bearing Faults, *Journal of Vibration and Acoustics*, Vol. 124 pp165-171, April 2002.

Polyshchuk, V. V., Choy, F. K. & Braun, M. J., (2002). Gear Fault Detection with Time-Frequency Based Parameter NP4, *International Journal of Rotating Machinery*, 8(1), 57-70, 2002.

Juluri, N. & Swarnamani, S., (2003). Improved accuracy of Fault Diagnosis of Rotating Machinery using Wavelet De-noising and Feature Selection, *Proceedings of ASME Turbo Expo 2003*, Power for Land, Sea and Air, June 16-19, 2003, Atlanta, Georgia, USA.

Antoni, J., Bonnardot, F., Raad, A. & El Badaoui, M., (2004). Cyclostationary modeling of rotating machine vibration signals, *Mechanical Systems and Signal Processing* 18 (6), pp. 1285-1314, 2004.

Yang, W. X. & Ren, X. M., (2004)., Detecting Impulses in Mechanical Signals by Wavelets, *EURASIP journal on Applied Signal Processing*, pp 1156-1162, 2004.

Bradford, S. C., (2006). Time-Frequency Analysis of Systems with Changing Dynamic Properties, Ph.D. Thesis, *California Institute of Technology, Pasadena, California,* 2006.

Huillery, J., Millioz, F. & Martin, N., (2008). On the description of spectrogram Probabilities with a Chi-Squared Law, *IEEE Transactions on Signal Processing* 56, 6 (2008) pp. 2249-2258.

Sawalhi, N. & Randall, R. B., (2008). Localised fault diagnosis in rolling element bearings in gearboxes, *Proceedings of The Fifth International Conference on Condition Monitoring and Machinery Failure Prevention Technologies – CM/MFPT*, 2008.

Clifton, D. A. & Tarassenko, L., (2009). Novelty detection in jet engine vibration spectra, *Proceedings of the 6th International Conference on Condition Monitoring and Machine Failure Prevention Technologies*, 2009.

Klein, R., Rudyk, E., Masad, E. & Issacharoff, M., (2009). Emphasizing bearing tones for prognostics, *The Sixth International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, pp. 578-587, 2009.

Bechhoefer, E., He, D. & Dempsey, P., (2011). Gear health threshold setting based on probability of false alarm, *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, 2011.

Klein, R., Masad, E., Rudyk, E. & Winkler, I. (2011). Bearing diagnostics using image processing methods, *Proceedings of Surveillance 6*, Compiegne, October 2011.

Klein, R., Rudyk, E. & Masad, E. (2012). Bearing diagnostics in non-stationary environment, *International Journal of Condition Monitoring*, March 2012.

Hazan, A., Verleysen, M., Cottrell, M., Lacaille, J. & Madani, K. (2012). Probabilistic outlier detection in vibration spectra with small learning dataset, *Mechanical Systems and Signal Processing*, 2012.

Hazan, A. & Madani, K., (2013). Frequency-Dependent Peak-Over-Threshold algorithm for fault detection in the spectral domain, *ESANN 2013*, Bruges: Belgium, 2013.

BIOGRAPHY

Renata Klein received her B.Sc. in Physics and Ph.D. in the field of Signal Processing from the Technion, Israel Institute of Technology. In the first 17 years of her professional career, she worked in ADA-Rafael, the Israeli Armament Development Authority, where she managed the Vibration Analysis department. In the decade that followed, she focused on development of vibration based health management systems for machinery. She invented and managed the development of vibration based diagnostics and prognostics systems that are used successfully in combat helicopters of the Israeli Air Force, in UAV's and in jet engines. Renata is a lecturer in the faculty of Aerospace Engineering of the Technion, and in the faculty of Mechanical Engineering in Ben Gurion University of the Negev. In the recent years, Renata is the CEO and owner of "R.K. Diagnostics", providing R&D services and algorithms to companies who wish to integrate Machinery health management and prognostics capabilities in their products.

Online Monitoring of Plant Assets in the Nuclear Industry

Vivek Agarwal[1], Nancy J. Lybeck[1], Binh T. Pham[1], Richard Rusaw[2], and Randall Bickford[3]

[1]*Idaho National Laboratory, Idaho Falls, ID, 83415, USA*
vivek.agarwal@inl.gov
nancy.lybeck@inl.gov
binh.pham@inl.gov

[2]*Electric Power Research Institute, Charlotte, NC, 28262, USA*
rrusaw@epri.com

[3]*Expert Microsystems, Orangevale, CA, 95662, USA*
rbickford@expmicrosys.com

ABSTRACT

Today's online monitoring technologies provide opportunities to perform predictive and proactive health management of assets within many different industries, in particular the defense and aerospace industries. The nuclear industry can leverage these technologies to enhance safety, productivity, and reliability of the aging fleet of existing nuclear power plants. The U.S. Department of Energy's Light Water Reactor Sustainability Program is collaborating with the Electric Power Research Institute's (EPRI's) Long-Term Operations program to implement online monitoring in existing nuclear power plants.

Proactive online monitoring in the nuclear industry is being explored using EPRI's Fleet-Wide Prognostic and Health Management (FW-PHM) Suite software, a set of web-based diagnostic and prognostic tools and databases that serves as an integrated health monitoring architecture. This paper focuses on development of asset fault signatures used to assess the health status of generator step-up transformers and emergency diesel generators in nuclear power plants. Asset fault signatures describe the distinctive features based on technical examinations that can be used to detect a specific fault type. Fault signatures are developed based on the results of detailed technical research and on the knowledge and experience of technical experts. The Diagnostic Advisor of the FW-PHM Suite software matches developed fault signatures with operational data to provide early identification of critical faults and troubleshooting advice that could be used to distinguish between faults with similar symptoms. This research is important as it will support the automation of predictive online monitoring techniques in nuclear power plants to diagnose incipient faults, perform proactive maintenance, and estimate the remaining useful life of assets.

1. INTRODUCTION

The average age of existing commercial nuclear power plants (NPPs) in the United States is around 34 years. As these plants continue to age and their components degrade, it is important to understand their condition and be proactive in maintenance and replacement. The current periodic and condition-based maintenance practices at NPPs result in high maintenance costs and increased likelihood of human error. Additionally, the inability to identify developing faults can lead to either disabling component failure or forced plant outage. Implementation of advanced predictive online monitoring would minimize these limitations and enhance plant safety by enabling systems and maintenance engineers to diagnose incipient faults and estimate the remaining useful life (RUL) of their assets, thereby reducing operational costs by optimizing maintenance activities. Predictive online monitoring techniques include advanced diagnostic and prognostic techniques.

The U.S. Department of Energy, Office of Nuclear Energy funds the Light Water Reactor Sustainability (LWRS) program to develop the scientific basis to extend the operation of commercial light water reactors beyond the current 60-year licensing period. The Advanced Instrumentation, Information, and Control Systems pathway under the LWRS program is collaborating with the Electric Power Research Institute's (EPRI's) Long-Term Operations (LTO) program to conduct research and development on technologies that can be used to enhance long-term reliability, productivity, and safety of aging light water reactors. One of the primary areas of focus for the LWRS

Agarwal et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

and LTO programs is *online monitoring (OLM) of active assets* in the nuclear industry.

An important objective of the online monitoring of active assets research is to implement predictive online monitoring for the existing fleet of NPPs. EPRI's Fleet-Wide Prognostic and Health Management (FW-PHM) Suite software was selected for use as a demonstration platform. The FW-PHM Suite was specifically developed by EPRI for use in the commercial power industry (both nuclear and fossil fuel). EPRI and Idaho National Laboratory (INL) are working with nuclear utility partners to develop diagnostic models in the FW-PHM Suite software for Generator Step-Up Transformers (GSUs) and Emergency Diesel Generators (EDGs). The nuclear utility partners include Shearon Harris Nuclear Generating Station (owned by Duke Progress Energy) for GSUs and Braidwood Generating Station (owned by Exelon Nuclear) for EDGs.

This paper presents the research and development performed to date with the GSU and EDG content development in the FW-PHM Suite and initial diagnoses. The paper is organized as follows. Section 2 summarizes some of the relevant works associated with fleet-wide monitoring. The FW-PHM Suite software architecture is briefly described in Section 3. Section 4 discusses the content development for GSUs and EDGs. Assessments of the diagnoses based on fault signatures entered in the FW-PHM Suite software are presented in Section 5. Finally, conclusions and future research are presented in Section 6.

2. Related Works

Fleet-wide diagnosis, prognosis, and knowledge management have gained significant interest across different industries. Review of some of the myriad of fleet-wide diagnosis and prognosis architectures, knowledge structures, and associated issues reported in literature is presented.

Deployment of a fleet-wide health management solution is challenging and requires a systematic approach (Johnson, 2012). The approach suggested in (Johnson, 2012) includes identifying assets and business needs, identifying critical components within assets, selecting sensory sources, and selecting prognostic methods.

Managing relevant knowledge arising both from modeling and monitoring of a fleet is essential (Monnin, Voisin, Leger, & Lung, 2011). A knowledge-structuring scheme based on ontologies for fleet-wide application of PHM in a marine domain was presented in (Medina-Oliva et al., 2012). In the case of fleets with heterogeneous assets, the knowledge-structure based on ontologies was utilized to search for assets based on similar characteristics (Monnin et al., 2011). Similarly, predictive maintenance of a fleet with homogeneous assets using an ontology based modeling approach was suggested by (Umiliacchi, Lane, & Romano, 2011).

In (Patrick et al., 2010) authors demonstrate that the threshold values indicating different fault conditions for a homogeneous fleet could be derived from statistical studies of fleet-wide behaviors of identical assets and known cases of faults. A similarity-based approach for estimating the remaining useful life of a fleet composed of similar assets was proposed by (Wang et al., 2008).

3. Fleet-Wide Prognostic and Health Management Suite

The FW-PHM Suite software is an integrated suite of web-based diagnostic and prognostic tools and databases, developed for EPRI by Expert Microsystems, specifically designed for use in the commercial power industry (for both nuclear and fossil fuel generating plants). The FW-PHM Suite serves as an integrated health management framework, as shown in Figure 1, managing the functionality needed for a complete implementation of diagnostics and prognostics (Electrical Power Research Institute, 2012). The FW-PHM Suite consists of four main modules: the Diagnostic Advisor, the Asset Fault Signature (AFS) Database, the Remaining Useful Life Advisor, and the Remaining Useful Life Database. The FW-PHM Suite has the capability to perform diagnosis and prognosis at different hierarchical levels, from the component level to the plant level, across a fleet of power units.

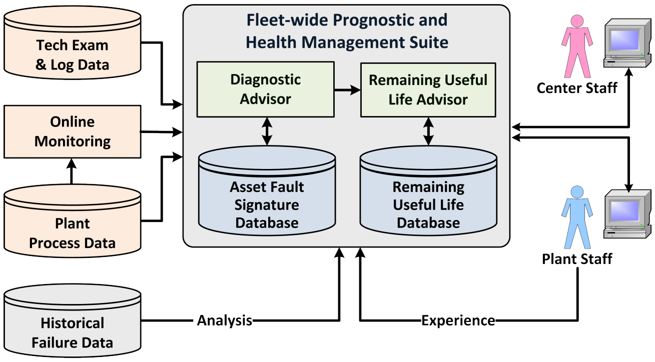

Tech Exam & Log Data
Online Monitoring
Plant Process Data
Historical Failure Data
Fleet-wide Prognostic and Health Management Suite
Diagnostic Advisor
Remaining Useful Life Advisor
Asset Fault Signature Database
Remaining Useful Life Database
Center Staff
Plant Staff
Analysis
Experience

Figure 1. Data flow in the EPRI FW-PHM Suite software architecture (Electrical Power Research Institute, 2012).

The FW-PHM Suite uses fault signatures as a structured representation of the information that an expert would use to first detect and then verify the occurrence of a specific type of fault (Electrical Power Research Institute, 2012). A fault describes a particular mode of degradation that can be detected by analysis of plant information before the asset condition reaches the point of failure to meet a service requirement. Implied is an assumption that the fault is detectable by analysis of plant information and that the analysis can be performed in time to prevent or otherwise remedy the fault condition before it becomes a failure.

Fault signatures are developed for application to a specific type of asset and are therefore organized with reference to that type of asset. However, it is desirable to specify fault signatures as broadly as possible to be used in the entire industry. Many of the fault signatures defined in this paper can be applied to comparable assets used in similar service environments.

3.1. Asset Fault Signature Database

The Asset Fault Signature database organizes fault signatures collected from the many EPRI member utilities. At the most basic level, fault signatures are comprised of an asset type, a fault type, and a set of one or more fault features (symptoms) that are indicative of the specified fault. Each installation of the software has two separate database schemas: the master database maintained and distributed by EPRI, and a local database containing data developed at the plants or fleet monitoring center. Locally developed information can be exported and sent to EPRI for evaluation and possible inclusion in the master database that is shared amongst EPRI members. The AFS Database is populated via a content development exercise that is described in the following section.

3.2. Diagnostic Advisor

The Diagnostic Advisor identifies possible faults by comparing asset fault signatures with operating data. The Diagnostic Advisor is expected to be used on a daily or other periodic basis by technicians who are monitoring the health of a specific asset in the plant. Using either online data sources or information that is input manually, the Diagnostic Advisor presents the likely faults (if any), and, when appropriate, recommends additional tests that might be used to discriminate amongst the possible faults. The Diagnostic Advisor is expected to streamline the diagnosis process by helping the technician focus his/her efforts on the most likely faults and possible causes based on the operating behavior of the system.

3.3. Remaining Useful Life Advisor

The Remaining Useful Life Advisor calculates remaining useful life for an asset based on the model type, model parameters, input process parameters, and diagnostic information (from the Diagnostic Advisor). The RUL Advisor is expected to be used on a periodic basis by technicians who are monitoring the health of a specific asset in the plant.

3.4. Remaining Useful Life Database

The RUL Database organizes asset RUL signatures (i.e., models) collected from across the industry. At the most basic level, a RUL signature is comprised of an asset type, a model type, and model calibration parameters. The model type definition includes the definition of the input variables needed to run the model. Subject matter experts from the power industry, EPRI, and EPRI's partners/subcontractors will most likely develop RUL signatures. Figure 2 shows the modules available in the FW-PHM Suite software.

4. Content Development for the Asset Fault Signature Database

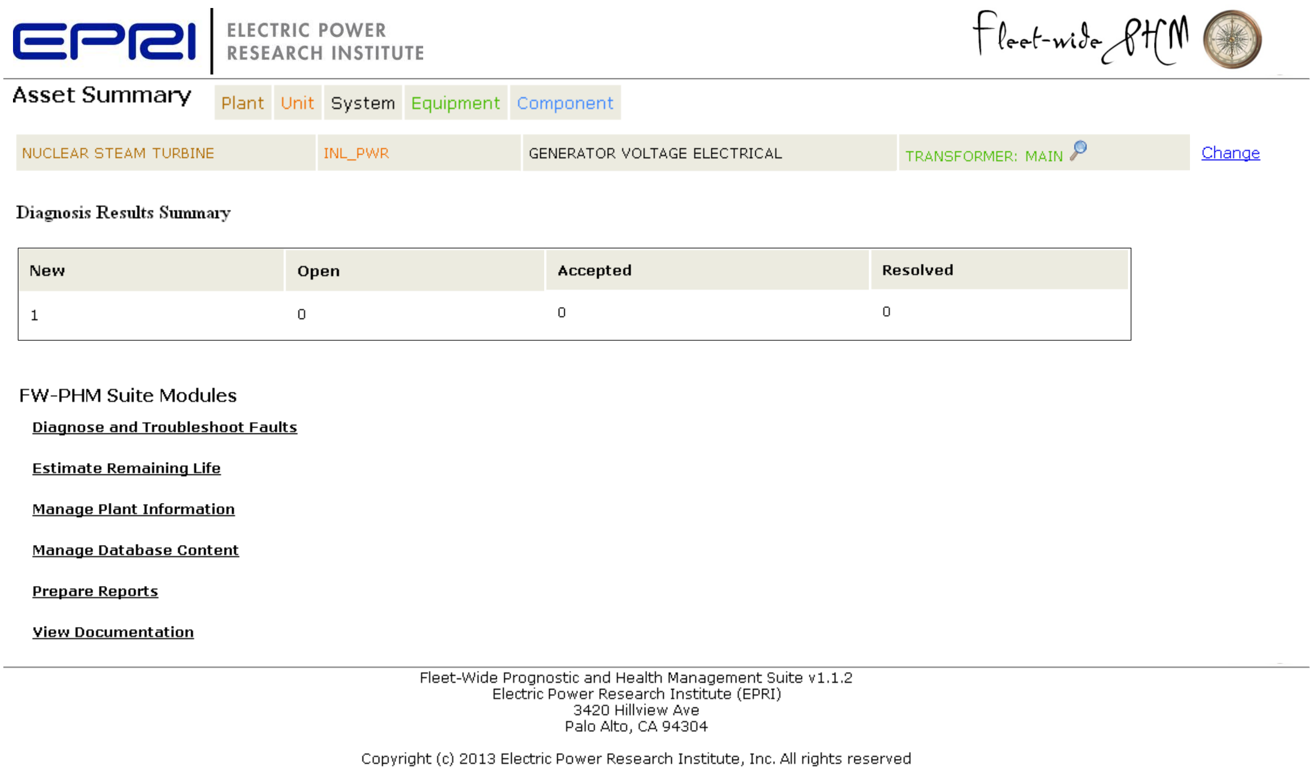
INL is working with subject matter experts from industry, EPRI, and EPRI's partners/subcontractors to develop content in the AFS Database for GSUs and EDGs. Currently, troubleshooting is typically a manual process that predominantly relies on expert knowledge and written documentation. The goal of content development is to capture this rich operating knowledge, creating a set of asset fault signatures organized in a standardized structure. The Diagnostic Advisor for automatic asset health monitoring uses these fault signatures and associated fault features. Content management for the asset fault signatures will be provided by EPRI.

Fault signatures for a specified asset must include, at a minimum, a fault description and associated fault features. Fault features represent a unique state of one or more parameters indicating a faulty condition; these parameters come from technical examinations of the asset. Therefore, identification of different technical examinations for a target asset is a critical step in the development of fault signatures. Some of the most common technical examination for GSUs and EDGs are presented in the following sections.

4.1. Technical Examinations of Generator Step-Up Transformers

The technical examinations listed below are commonly used to monitor the operation of GSUs (Lybeck et al., 2011). These examinations allow assessment of winding insulation degradation, loss of dielectric strength of insulating oil, cooling system effectiveness, and bushing degradation.

1. *Temperature analysis*. The top oil temperature reflects the effectiveness of a cooling system.
2. *Insulating Oil Analysis (online or offline).* Performing oil analysis allows assessment of the electrical property (e.g. dielectric strength), chemical properties (e.g., water content, acidity), and the physical property (e.g. interfacial tension (IFT)). Insulating oil loses its dielectric strength either due to increase in the moisture content or due to thermal aging. Acidity of insulating oil needs to be monitored, as an increase in acidity is harmful because if the oil becomes acidic, the solubility of the water in the oil increases and also deteriorates the winding paper insulation strength. Acidity is measured in terms of milligrams of potassium hydroxide present in one gram of oil. IFT measures molecular attractive forces between oil and water. A decrease in the IFT indicates the presence of contaminants in the oil and is measured in Dyne/cm.



Diagnosis Results Summary

New	Open	Accepted	Resolved
1	0	0	0

FW-PHM Suite Modules

Fleet-Wide Prognostic and Health Management Suite v1.1.2
Electric Power Research Institute (EPRI)
3420 Hillview Ave
Palo Alto, CA 94304

Copyright (c) 2013 Electric Power Research Institute, Inc. All rights reserved

Figure 2. The FW-PHM Suite main page.

3. *Dissolved gas analysis*. Another important chemical property of transformer insulating oil to be analyzed is the dissolved gas concentration. Thermal or electrical faults occurring inside the transformer decompose the hydrocarbon bonds, resulting in generation of gases within the transformer. One of the most important aspects of oil analysis is to measure the concentration of key dissolved gases, which include hydrogen (H_2), methane (CH_4), acetylene (C_2H_2), ethylene (C_2H_4), and ethane (C_2H_6). In addition to the key gases, carbon monoxide (CO), carbon dioxide (CO_2), oxygen (O_2), and nitrogen (N_2) are also generated, even under normal operating conditions. The gas ratios such as O_2/N_2, CO2/CO, C_2H_2/H_2, Doernenberg Ratios, Duval triangle (Duval, 2002), and Rogers Ratios indicate different types of degradation inside transformers.
4. *Doble Capacitance Test.* Measuring bushing capacitance is a standard technique used to determine bushing condition. The main capacitance (C1) test is conducted in the ungrounded specimen mode, i.e., the ground lead is not used for measurement, but instead the selected low voltage leads are used. The outcome of the C1 test allows assessment of contamination in the main body of the bushing. The tap capacitance (C2) test is conducted in the grounded specimen mode, i.e., at least part of the test current is measured through the grounded lead; the rest is measured through the low voltage leads, if used and not guarded. The outcome of the C2 test allows assessment of contamination in the oil and tap area.
5. *Sweep Frequency Response Test.* The frequency responses measured during a sweep frequency response test reflect the measured winding capacitance. These responses are compared to the reference sweep frequency responses. Any deviation in the measured responses from the reference indicates capacitance change that might be due to winding movement or core displacement.

4.2. Technical Examinations for Emergency Diesel Generators

The technical exams listed below are commonly used to monitor diesel engine operation (Pham, Lybeck, & Agarwal, 2012). These examinations allow assessment of degradation of valves, fuel injectors, seals, and piston rings as well as the overall health of the EDG engine.

1. *Temperature analysis.* Component temperatures (e.g., engine cylinders or exhaust manifold) indicate engine performance. Cooling water temperatures and lubricating oil temperatures (inlet, outlet, and their difference) are used to monitor the thermodynamic efficiency of the engine (Banks, 2001).
2. *Pressure and flow rate analysis.* Deviation of engine cylinder pressure (measured by a pressure transducer mounted on the cylinder head) from baseline pressure-time curves for each of the cylinders indicates a variety

of abnormal engine operating conditions. The key reference points in time are peak firing pressure, peak firing pressure crank angle, maximum pressure rise rate, start of injection, and start of combustion (Banks et al., 2001). Pressure and fluid volume measurements (inlet and outlet) from engine support systems such as the fuel oil system, lubricating oil system, cooling water system, and starting air system are used to identify leakage and component failure of the corresponding system. Abnormally low pressure in these systems usually indicates either system leakage of fuel, oil, water, or air or pump failure.

3. *Vibration analysis.* Vibration data from various engine components as a function of run time (or crankshaft angle) can be used to assess the condition of the bearing, the crankshaft, and other moving parts without physical examinations. The existence of peaks at frequencies higher than 2-times the line frequency in the engine vibration spectrum can indicate liner scuffing and blow by.

4. *Engine oil analysis (or lubricating oil analysis, online or offline)*: Oil analysis is used to detect metal particles (e.g., particle count according to size), fuel oil, water, or combustion products in the lubricating oil, indicating problems in the diesel engine, including mechanical wear of components, bearing failure, and leaking seals (Banks et al., 2001). There are three basic technical examination methods for oil analysis.

   Ferrography (International Standards Organization, 1999)*:* Ferromagnetic particles in the lubricating oil are counted using a magnetic field to separate the particles according to size. The ferrography oil analysis includes the following operations: collection of wear particles according to size on a transparent substrate; selection and separation of significant particles; inspection and evaluation of the particles and their morphology and nature; and identification of particles (i.e., type of material).

   Spectroscopy (American Society for Test and Materials, 2009): The frequency and intensity of light emitted from electrically excited particles are measured using a spectrometer to detect particles in the lubricating oil.

   Particle count: Particles are counted in the engine lube oil using a particle counter. The nature of particle counting is based on light scattering, light obscuration, or direct imaging when the particle passes through a high-energy light beam.

5. *Power analysis.* Voltage and frequency measurements at the outlets of an EDG potential transformer can be used to assess its performance and detect faults when these parameters are not within specified ranges.

4.3. Fault Signatures

A step-by-step procedure for developing and implementing an asset fault signature in the AFS Database is described in EPRI (2012) and shown in Figure 3. Several fault signatures have been developed and implemented in the AFS Database as part of a knowledge transfer exercise with utility partners for GSUs and EDGs. Two representative fault signatures are described below.

Primary winding paper insulation degradation is one of the common faults in transformers (Bartley, 2003). The two most common modes of primary winding paper insulation degradation are electrical and thermal.

Paper insulation degradation due to electrical discharges represents the occurrence of either a partial discharge phenomenon or an arcing phenomenon. A steep increase in the H_2 concentration level compared to other dissolved gases in the transformer insulating oil is an indication of partial discharge. Similarly, an increase in the H_2 and C_2H_2 concentration levels combined compared to other dissolved gases is an indication of arcing.

Paper insulation degradation due to thermal phenomenon can be diagnosed when a steep increase in the carbon monoxide concentration level is observed. As a result, a decrease in the CO_2/CO ratio is observed. Therefore, a decrease in the CO_2/CO ratio also indicates thermal degradation of primary winding paper insulation.

EDGs are safety-related assets that are required to operate reliably if the external grid power supply to a plant is interrupted. It is required to start, run, and take the basic load that is essential for safe shutdown of the plant. There are many faults that could lead to EDG failure. One of the faults is *improper valve timing* (alternately referred as ignition timing) for the diesel engine cylinder. An increase or decrease or both in cylinder temperature may be due to improper value timing. Lower and upper threshold limits are defined to monitor the change in cylinder temperature.

In Appendix A, the currently implemented fault signatures for GSUs and EDGs are summarized in Tables A1 and A2 respectively. The last column of Tables A1 and A2 (i.e., the 5th column) lists the effectiveness of each technical examination. Effectiveness is used in ranking possible diagnosis of the Diagnostic Advisor, especially in a situation where the same technical examination is used to diagnose different fault types. These fault signatures represent the initial effort to create useful fault signatures for GSUs and EDGs, but do not create a complete diagnostic system for GSUs and EDGs.

5. Diagnosis Process

In this section, the ability of the Diagnostic Advisor to diagnose a developing fault in a GSU and an EDG is presented. The Diagnostic Advisor compares the simulated

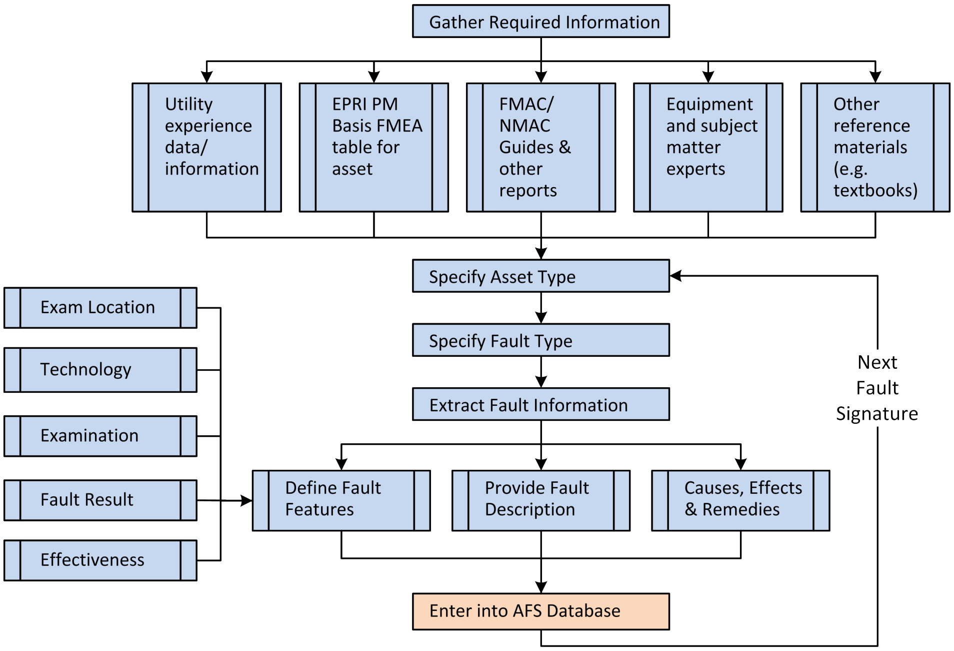

Gather Required Information
Utility experience data/ information
EPRI PM Basis FMEA table for asset
FMAC/ NMAC Guides & other reports
Equipment and subject matter experts
Other reference materials (e.g. textbooks)
Specify Asset Type
Specify Fault Type
Extract Fault Information
Next Fault Signature
Exam Location
Technology
Examination
Fault Result
Effectiveness
Define Fault Features
Provide Fault Description
Causes, Effects & Remedies
Enter into AFS Database

Figure 3. Steps involved in gathering asset fault signatures (EPRI, 2012).

operating data with the established fault signatures in the AFS Database, to assess the health of a plant asset. The diagnosis of primary winding paper insulation degradation in a GSU and of improper valve timing in an EDG is discussed.

5.1. GSU Primary Winding Paper Insulation Diagnosis

The Diagnostic Advisor's ability to diagnose the primary winding paper insulation degradation is discussed in detail in Agarwal et al. (2013). For the purpose of testing, the Diagnostic Advisor is connected to a simulated data stream in which an initial increase in the acetylene concentration level is followed by an increase in the carbon monoxide concentration level and a decrease in the CO_2/CO ratio. The threshold limits in the Diagnostic Advisor for each monitored gas level are mapped to 3-out-of-4 classification criteria (Condition 1 through Condition 3) developed by the Institute of Electrical and Electronics Engineers (IEEE) to classify risk to transformers (IEEE, 2008). Table 1 in IEEE (2008) lists the dissolved gas concentration for the individual gases for Condition 1 through Condition 4.

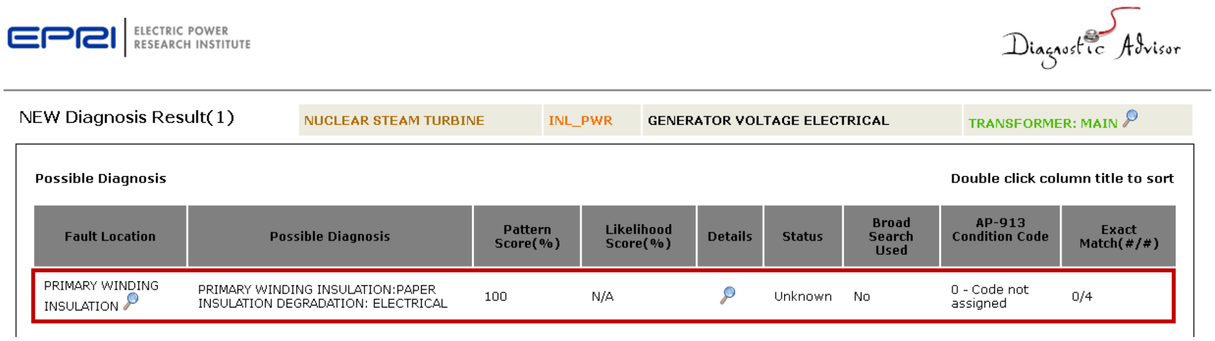
During simulation, when the acetylene concentration enters Condition 2, the Diagnostic Advisor identifies the change and generates a possible diagnosis result. It identifies primary winding paper insulation degradation due to electrical phenomena, as shown in Figure 4, as the most likely fault (as expected based on technical examinations and fault signatures implemented in the AFS Database). As the simulation continues, the carbon monoxide gas level enters Condition 2 and soon enters Condition 3. As the carbon monoxide gas level increases to Condition 3, the CO_2/CO ratio reduces below 3 (i.e., below the recommended level (IEEE, 2008)). The Diagnostic Advisor records these changes and updates its previous possible diagnosis results. Based on the updated diagnoses, as shown in Figure 5, primary winding paper insulation degradation due to thermal phenomena is now the most likely fault.

The key observation is that the Diagnostic Advisor updates its possible diagnosis outcomes as new information becomes available.

5.2. EDG Improper Valve Timing Diagnosis

A 20 cylinder EDG is considered in this diagnosis example. As stated earlier, improper valve timing in a diesel engine can lead to either an increase or decrease in cylinder temperature, or both. To mimic improper valve timing, a positive drift and a negative drift are simulated in two randomly selected cylinders. During simulation, when the cylinder temperature and the cylinder temperature differential (i.e., difference between the maximum and the minimum temperature) exceed the user-defined threshold limits, the Diagnostic Advisor recognizes the change and diagnoses the possible fault. Figure 6 shows the possible diagnosis result. Observe that the two possible diagnoses are close in terms of pattern score (i.e., the percentage indicating the relative likelihood of the fault based on the current information) and further evidence is required to identify the most likely fault type.

One of the additional evidences that can be used to update the diagnosis result is the information on recent maintenance. The Diagnostic Advisor provides an option to manually update the previous diagnosis result. By utilizing the option, recent maintenance information was manually



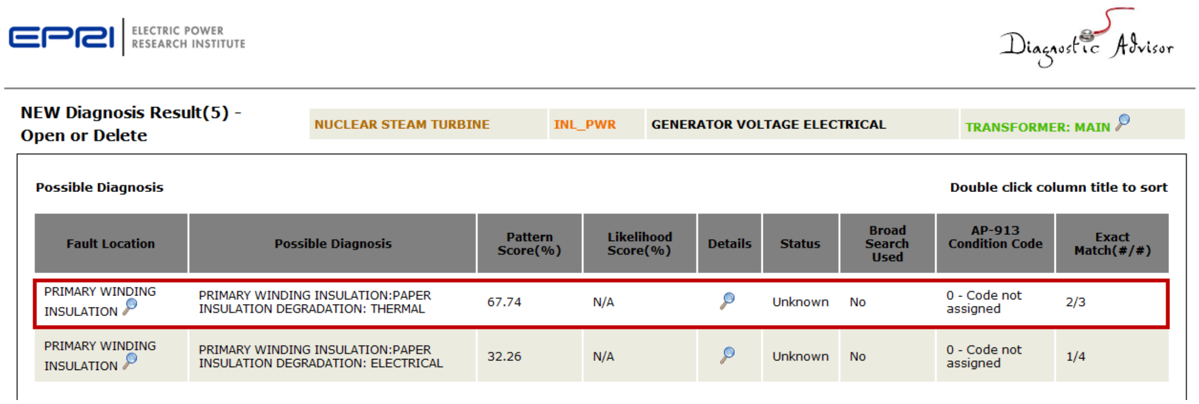
NEW Diagnosis Result(1) | NUCLEAR STEAM TURBINE | INL_PWR | GENERATOR VOLTAGE ELECTRICAL | TRANSFORMER: MAIN

Possible Diagnosis

Double click column title to sort

Fault Location	Possible Diagnosis	Pattern Score(%)	Likelihood Score(%)	Details	Status	Broad Search Used	AP-913 Condition Code	Exact Match(#/#)
PRIMARY WINDING INSULATION	PRIMARY WINDING INSULATION:PAPER INSULATION DEGRADATION: ELECTRICAL	100	N/A		Unknown	No	0 - Code not assigned	0/4

Figure 4. New Diagnosis Result page created by the Diagnostic Advisor when the acetylene level reaches IEEE Condition 2.



NEW Diagnosis Result(5) - Open or Delete | NUCLEAR STEAM TURBINE | INL_PWR | GENERATOR VOLTAGE ELECTRICAL | TRANSFORMER: MAIN

Possible Diagnosis

Double click column title to sort

Fault Location	Possible Diagnosis	Pattern Score(%)	Likelihood Score(%)	Details	Status	Broad Search Used	AP-913 Condition Code	Exact Match(#/#)
PRIMARY WINDING INSULATION	PRIMARY WINDING INSULATION:PAPER INSULATION DEGRADATION: THERMAL	67.74	N/A		Unknown	No	0 - Code not assigned	2/3
PRIMARY WINDING INSULATION	PRIMARY WINDING INSULATION:PAPER INSULATION DEGRADATION: ELECTRICAL	32.26	N/A		Unknown	No	0 - Code not assigned	1/4

Figure 5. Updated New Diagnosis Result page indicating paper insulation degradation: thermal as the most likely fault.

entered and the previous diagnosis result was updated. Based on the updated result, as shown in Figure 7, it can be observed that the pattern score for improper valve timing has increased to 56.52% compared to the previous score of 52.94% (Figure 6).

This example demonstrates that same technical examination can be used to diagnose different fault types. In such situation, additional evidence can be included manually (if available) to clearly identify the most likely fault type.

6. Conclusions and Future Research

The paper presented research and development performed to date towards implementation of predictive online monitoring for existing nuclear power plants using the FW-PHM Suite software. Several fault signatures were developed for common faults observed in GSUs and EDGs based on the results of technical research and on the knowledge and experience of technical experts. After a thorough verification and validation process, these fault signatures will serve as a foundation for implementation of automated online monitoring for GSUs and EDGs in the nuclear industry. The discussion of diagnosis results highlights the ability of the Diagnostic Advisor to identify the most likely fault type based on the currently implemented fault signatures.

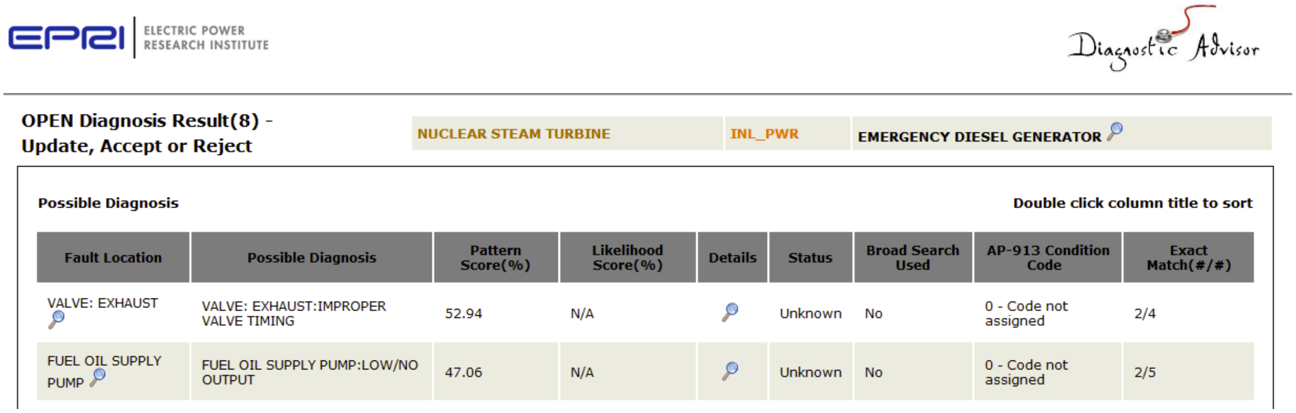
In the future, EPRI and INL will continue to work with nuclear utility partners to develop and verify a full set of fault signatures covering a wide range of recognized faults for GSUs and EDGs, enabling implementation in NPPs. Ultimately, prognostic models based on physics of failure will be developed and implemented in the FW-PHM Suite to predict the remaining useful life of GSUs and EDGs.

Acknowledgement

The authors would like to thank Kirk Fitzgerald (INL) and Rahul Palnitkar (Expert Microsystems) for their substantial support. The research work was supported by the Light Water Reactor Sustainability Program at Idaho National Laboratory, funded by the U.S. Department of Energy under DOE Idaho Operations Office Contract DE-AC07-05ID14517.

Disclaimer

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.



OPEN Diagnosis Result(8) - Update, Accept or Reject

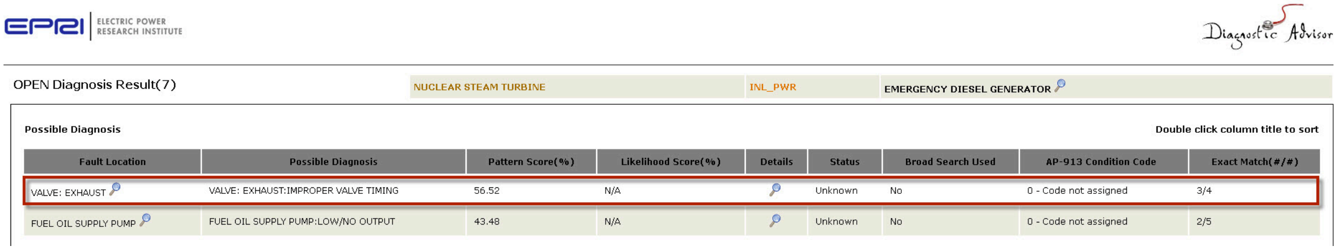
NUCLEAR STEAM TURBINE | INL_PWR | EMERGENCY DIESEL GENERATOR

Possible Diagnosis

Double click column title to sort

Fault Location	Possible Diagnosis	Pattern Score(%)	Likelihood Score(%)	Details	Status	Broad Search Used	AP-913 Condition Code	Exact Match(#/#)
VALVE: EXHAUST	VALVE: EXHAUST:IMPROPER VALVE TIMING	52.94	N/A		Unknown	No	0 - Code not assigned	2/4
FUEL OIL SUPPLY PUMP	FUEL OIL SUPPLY PUMP:LOW/NO OUTPUT	47.06	N/A		Unknown	No	0 - Code not assigned	2/5

Figure 6. New Diagnosis Result page created by the Diagnostic Advisor when the cylinder temperature and the cylinder temperature differential exceed threshold limits.



OPEN Diagnosis Result(7)

NUCLEAR STEAM TURBINE | INL_PWR | EMERGENCY DIESEL GENERATOR

Possible Diagnosis

Double click column title to sort

Fault Location	Possible Diagnosis	Pattern Score(%)	Likelihood Score(%)	Details	Status	Broad Search Used	AP-913 Condition Code	Exact Match(#/#)
VALVE: EXHAUST	VALVE: EXHAUST:IMPROPER VALVE TIMING	56.52	N/A		Unknown	No	0 - Code not assigned	3/4
FUEL OIL SUPPLY PUMP	FUEL OIL SUPPLY PUMP:LOW/NO OUTPUT	43.48	N/A		Unknown	No	0 - Code not assigned	2/5

Figure 7. Updated Diagnosis Result page after manual entry of recent maintenance information.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

REFERENCES

Agarwal, V., Lybeck, N., Matacia, L., & Pham, B. (2013). *Demonstration of Online Monitoring for Generator Step-Up Transformers and Emergency Diesel Generators*. Report INL/EXT-13-30155. Idaho National Laboratory, Idaho Falls, ID.

American Society for Test and Materials (ASTM) (2009). Standard Test Method for In-Service Monitoring of Lubricating Oil for Auxiliary Power Plant Equipment. In ASTM*, ASTM D6224-09*, West Conshohocken, PA, USA: ASTM International.

Banks, J., Hines, J., Lebold, M., & Campbell, R. (2001). Failure modes and predictive diagnostics considerations for diesel engines. *Proceedings of the 55th Meeting of the Society for Machinery Failure Prevention Technology*, April 2–5, Virginia Beach, VA.

Bartley, W. H. (2003). Analysis of Transformer Failures. *Proceedings of the 36th Annual Conference on International Association of Engineering Insurers*.

Duval, M. (2002). A review of fault detectable by gas-in-oil analysis in transformers. *IEEE Electrical Insulation Magazine*, vol. 18, no. 3, pp. 8-17.

Electric Power Research Institute (EPRI) (2012). *Fleet-Wide Prognostics and Health Management Application Research*. Report EPRI 1026712. Electric Power Research Institute, Charlotte, NC.

Electrical Power Research Institute (EPRI) (2012). *Asset Fault Signature Requirements*. Software manual. Electric Power Research Institute, Charlotte, NC.

Institute of Electrical and Electronics Engineers (IEEE) (2008). IEEE Guide for Interpretation of Gases Generated in Oil-Immersed Transformers. In IEEE*, IEEE Std C57.104: 2008*. New York, USA.

International Standards Organization (ISO) (1999). Hydraulic fluid power—Fluids—Method for coding the level of contamination by solid particles. *In ISO, ISO4406:1999*. Genève, Switzerland: International Standards Organization.

Johnson, P. (2012). Fleet wide asset monitoring: Sensory Data to Signal Processing to Prognostics. *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, September 23-27, Minneapolis, MN. ISBN-978-1-036263-05-9.

Lybeck, N., Agarwal, V., Pham, B., Medema, H., & Fitzgerald, K., (2012). *Online Monitoring Technical Basis and Analysis Framework for Large Power Transformers: Interim Report for FY 2012*. Report

INL/EXT-12-27181. Idaho National Laboratory, Idaho Falls, ID.

Monnin, M., Voisin, A., Leger, J., & Lung, B. (2011). Fleet-Wide Health Management Architecture. *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, September 25-29, Montreal, Quebec, Canada. ISBN-978-1-936263-03-5.

Monnin, M., Abichou, B., Voisin, A., & Mozzati, C. (2011). Fleet Historical Case for Predictive Maintenance. *Proceedings of the International Conference on Surveillance* 6, October 25-26, Compiegne, France.

Medina-Oliva, G., Voisin, A., Monnin, M., Peysson, F., & Leger, J. B. (2012). Prognostic Assessment using Fleet-Wide Ontology. *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, September 23-27, Minneapolis, MN. ISBN-978-1-036263-05-9

Patrick, R., Smith, M. J., Byington, C. S., Vachtsevanos, G. J., Tom, K., & Ly, C. (2010). Integrated Software Platform for Fleet Data Analysis, Enhanced Diagnostics, and Safe Transition to Prognostics for Helicopter Component CBM. *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, October 13-16, Portland, OR. ISBN-978-1-936263-01-1.

Pham, B., Lybeck, N., & Agarwal, V. (2012). *Online Monitoring Technical Basis and Analysis Framework for Emergency Diesel Generators: Interim Report for FY 2013*. Report INL/EXT-12-27754. Idaho National Laboratory, Idaho Falls, ID.

Umiliacchi, P., Lane, D., & Romano, F. (2011). Predictive Maintenance of Railway Subsystem using an Ontology based Modeling Approach. *Proceedings of 9th World Conference on Railway Research*, May 22-26, Lille, France.

Wang, T., Yu, J., Siegel, D., & Lee, J. (2008). A Similarity-based Prognostics Approach for Remaining Useful Life Estimation of Engineered Systems. *Proceedings of the International Conference on Prognostics and Health Management*, October 06-09, Denver, CO.

BIOGRAPHIES

Vivek Agarwal is a Research Scientist at the Idaho National Laboratory, Idaho Falls, ID. He holds a Ph.D. in nuclear engineering from Purdue University. He also worked as a research associate at Hewlett Packard Laboratories, Palo Alto, CA. His research interests include signal processing, machine learning, battery modeling, wireless sensor networks, instrumentation and controls, diagnostics and prognostics. He is a Section Editor for the Journal of Pattern Recognition Research.

Nancy J. Lybeck is a Data Analyst at the National Laboratory, Idaho Falls, ID. She holds a Ph.D. in mathematics from Montana State University. Her research interests include numerical analysis, applied mathematics, diagnostics and prognostics.

Binh T. Pham is a Research Scientist at the Idaho National Laboratory, Idaho Falls, ID. She received her Ph.D. and M.S. in electronics engineering from Moscow Power Engineering Institute, Russia in 1993 and 1986 respectively. Her research interests include quantitative modeling, statistical modeling, programming and computer simulation.

Richard Rusaw is a Senior Project Manager in the nuclear sector at the Electric Power Research Institute. He received his B.S. in nuclear engineering from University of Michigan in 1979, his MBA from University of North Carolina – Charlotte in 1983. He is a registered PE in the state of South Carolina. At EPRI, his responsibilities are focused on instrumentation and control (I&C) with a high degree of specialization in monitoring technologies and system reliability. Prior to joining EPRI, he spent 25 years at Duke Power as an I&C engineer with a wide range of responsibilities supporting Oconee, McGuire, and Catawba nuclear power plants. His current research interests include diagnostic and prognostics software, development of advanced sensors for nuclear applications, advanced information technologies for plant productivity, reliability, and Long-Term Operation.

Randall Bickford is Expert Microsystems' President and Chief Technology Officer. He holds a B. S. in chemical engineering from University of California, Davis. He is a recognized worldwide expert in Prognostic and Health Management technology. He is one of the industry's pioneers and holds multiple patents in the areas of pattern recognition, fault detection, diagnostics, and prognostics. Prior to founding Expert Microsystems, he worked in the aerospace industry where he developed advanced diagnostic and digital control technologies for space propulsion systems.

APPENDIX A: TECHNICAL EXAMS AND FAULT SIGNATURES ADDED TO AFS DATABASE FOR GSUS AND EDGS

Table A1. GSU fault signatures developed and implemented into the AFS Database (Lybeck et al., 2012).

Equipment or Component	**Fault Type**	**Technical Exam and Location**	**Fault Feature**	**Effectiveness**
Winding insulation	Paper insulation degradation	*Dissolved gas analysis:* levels of H_2 and C_2H_2 gas in insulating oil	High levels of H_2 and C_2H_2	High
		Dissolved gas analysis: levels of C_2H_2 gas in insulating oil	High levels of C_2H_2	Very High
		Dissolved gas analysis: CO_2/CO ratio in insulating oil	High ratio of CO_2/CO	High
		Temperature analysis: time at excess temperature at transformer winding	Long time at excessive temperature	Medium
		Insulating oil analysis: Acid number of insulating oil	High acid number	Medium
		Dissolved gas analysis: levels of H_2 gas in insulating oil	High levels of H_2	Very High
Insulating Oil	High acidity	*Insulating oil analysis*: level of KOH per gram of oil	High value of KOH	Very High
	Contamination	*Insulating oil analysis*: the value of IFT measured in Dynes/cm or in mN/m	Low IFT value	Very High
		Insulating oil analysis: color variation	Distinct color change	High
	Low dielectric strength	*Insulating oil analysis*: to measure the dielectric breakdown voltage (ASTM D877-02)	Low value of breakdown voltage	Very High
		Insulating oil analysis: measure the moisture content in the oil	High value of moisture	High
		Insulating oil analysis: the value of IFT measured in Dynes/cm or in mN/m	High value of KOH	Medium
		Dielectric strength: measure the power factor of the oil in the bushing	High value of power factor	Very High
Bushing	Low dielectric strength	*Dielectric strength*: measure the main capacitance	High value of main capacitance	High
		Dielectric strength: measure the tap capacitance	High value of tap capacitance	High
		Dielectric strength: measure the power factor of the oil in the bushing	High value of power factor	High
		Inspection: measure the oil level	Low level of oil	High
Core	Displaced winding core	*Sweep frequency response analysis:* captures the core capacitance across different frequencies	Change in the capacitance across different frequencies	Very High

Equipment or Component	**Fault Type**	**Technical Exam and Location**	**Fault Feature**	**Effectiveness**
Insulating oil pump motor	Loss of performance	*Insulating oil motor pump:* measure electric voltage	High value of electric voltage	High
		Insulating oil motor pump: measure electric current	High value of electric current	Very High
		Insulating oil motor pump: measure electric resistance	High value of electric resistance	High

Table A2. EDG fault signatures developed and implemented into the AFS Database (Pham, Lybeck, & Agarwal, 2012).

Equipment or Component	**Fault Type**	**Technical Exam and Location**	**Fault Feature**	**Effectiveness**
Diesel engine fuel injector	Improper fuel injection	*Temperature analysis:* temperature at the exhaust manifold	Abnormal temperature (too high or too low)	High
		Temperature analysis: temperature at the exhaust manifolds	High temperature differential between exhaust manifolds	High
		Inspection: unpleasant odor of fuel	Unpleasant smell of fuel	Very High
Diesel engine piston	Excessive wear	*Lubricating oil analysis:* evaluate chromium and aluminum content in lube oil sampled from the sump	High value of chromium and aluminum	Very High
		Temperature analysis: temperature at the cylinder neck	Low value of temperature	High
Fuel oil supply pump	Fuel pump failure	*Temperature analysis:* temperature at the cylinder neck	High value of temperature	Medium
		Temperature analysis: temperature at the cylinder neck	Low value of the temperature	Medium
		Temperature analysis: temperature at the cylinder neck	High temperature differential between cylinder necks	High
		Inspection: position of fuel metering rod	Displacement in the position	High
		Pressure analysis: of fuel line	Abnormal value of pressure	Very High
Diesel engine exhaust valve	Improper valve timing	*Temperature analysis:* temperature at the cylinder neck	High value of temperature	High
		Temperature differential: between the cylinder necks	High temperature differential	Very High
		Maintenance activity: on the diesel engine	Time from maintenance	High

Equipment or Component	**Fault Type**	**Technical Exam and Location**	**Fault Feature**	**Effectiveness**
Governor	Unresponsive governor	*Power analysis:* power (KW) at Potential Transformer inlets	KW power unchanged in response to demand change	High
		Power analysis: power frequency at potential Transformer inlets	Frequency fluctuating in response to demand change	High
		Inspection: position of fuel rack linkage	Position unchanged in response to demand change	High
		Power analysis: Voltage across governor outlets	Voltage unchanged in response to demand change	High
		Power Analysis: Voltage at Magnetic Pickup Unit	Voltage changed with demand change	Medium
		Temperature analysis: Temperature at the Exhaust Manifold	Temperature does not change in response to demand change	Very High
	Intermittent control signal	*Power analysis:* power (KW) at Potential Transformer inlets	KW power profile fluctuating	High
		Power analysis: power frequency at potential Transformer inlets	Frequency profile fluctuating	High
		Power analysis: Voltage across governor outlets	Voltage profile fluctuating	Very High
		Temperature analysis: temperature at the exhaust manifold	Temperature profile fluctuating	High
		Inspection: position of fuel rack linkage	Position profile fluctuating	High
		Power Analysis: Voltage at Magnetic Pickup Unit	Voltage profile fluctuating	High
	Intermittent MPU signal	*Power analysis:* power (KW) at Potential Transformer inlets	KW power profile fluctuating	Medium
		Power analysis: power frequency at potential Transformer inlets	Frequency profile fluctuating	Low
		Power analysis: Voltage across governor outlets	Voltage profile fluctuating	High
		Temperature analysis: temperature at the exhaust manifold	Temperature profile fluctuating	High
		Inspection: position of fuel rack linkage	Position profile fluctuating	High
		Power Analysis: Voltage at Magnetic Pickup Unit	Voltage profile fluctuating	Very High

Electrochemistry-based Battery Modeling for Prognostics

Matthew Daigle[1] and Chetan S. Kulkarni[2]

[1] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
matthew.j.daigle@nasa.gov

[2] *SGT, Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA*
chetan.s.kulkarni@nasa.gov

ABSTRACT

Batteries are used in a wide variety of applications. In recent years, they have become popular as a source of power for electric vehicles such as cars, unmanned aerial vehicles, and commericial passenger aircraft. In such application domains, it becomes crucial to both monitor battery health and performance and to predict end of discharge (EOD) and end of useful life (EOL) events. To implement such technologies, it is crucial to understand how batteries work and to capture that knowledge in the form of models that can be used by monitoring, diagnosis, and prognosis algorithms. In this work, we develop electrochemistry-based models of lithium-ion batteries that capture the significant electrochemical processes, are computationally efficient, capture the effects of aging, and are of suitable accuracy for reliable EOD prediction in a variety of usage profiles. This paper reports on the progress of such a model, with results demonstrating the model validity and accurate EOD predictions.

1. INTRODUCTION

With electric unmanned aerial vehicles, electric/hybrid cars, and commercial passenger aircraft, we are witnessing a drastic increase in the usage of batteries to power vehicles. These vehicles are being deployed in military, civilian and scientific applications all over the world. However, for battery-powered vehicles to operate at maximum efficiency and reliability, it becomes crucial to both monitor battery health and performance and to predict end of discharge (EOD) and end of useful life (EOL) events.

For example, NASA's Mars Global Surveyor stopped operating in late 2006 due to battery overheating caused by the raditor being oriented directly towards the sun, resulting in a significant loss of capacity (Saha & Goebel, 2009). Both the AFRL ARGOS satellite and the Viking 2 Mars Lander also suffered battery failures. More recently, problems have arisen in the Boeing Dreamliner aircraft, which utilize lithium-ion (Li-ion) batteries for auxiliary power (Ross, 2013). Battery monitoring failed to capture thermal runaway conditions, leading to fire. With advanced health monitoring and predictive capabilities, such problems can be prevented.

To implement such technologies, it is crucial to understand how batteries work and to capture that knowledge in the form of models that can be used by monitoring, diagnosis, and prognosis algorithms. Several battery modeling methodologies have been developed in the literature. The models can generally be classified as *empirical models*, *electrochemical engineering models*, *multi-physics models*, and *molecular/atomist models* (Ramadesigan et al., 2012). Emprical models (e.g., (Saha & Goebel, 2009)), are based on fitting certain functions to past experimental data, without making use of any physicochemical principles. Electrical circuit equivalent models are popular forms of empirical models, e.g., (Ceraolo, 2000; Chen & Rincon-Mora, 2006; Daigle, Saxena, & Goebel, 2012). Electrochemical engineering models are typically continuum models that include electrochemical kinetics and transport phenomena, e.g. (Karthikeyan, Sikha, & White, 2008; Rong & Pedram, 2006; Ning & Popov, 2004; Doyle, Fuller, & Newman, 1993; Newman & Tiedemann, 1975). The former type of model has the advantage of being computationally efficient, but the disadvantage of limited accuracy and robustness, due to the approximations in battery behavior that are made, and as a result of such approximations, cannot represent aging well. The latter type of model has the advantage of being very accurate, but are often computationally inefficient, having to solve complex sets of partial differential equations, and thus being ill-suited to online application of prognostics. Similarly, multi-physics (Lee, Smith, Pesaran, & Kim, 2013) and atomist models are even less suited to online application.

In this work, we develop an electrochemistry-based model of Li-ion batteries, that (*i*) capture the significant electrochemical processes, (*ii*) are computationally efficient, (*iii*) capture

Matthew Daigle et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the effects of aging, and (*iv*) are of suitable accuracy for reliable EOD prediction in a variety of usage profiles. The model can be considered an electrochemical engineering model, but unlike most such models found in the literature, we make certain approximations that allow us to retain computational efficiency. Although we focus here on Li-ion batteries, the model is quite general and can be applied to different chemistries through a change of model parameter values. This paper reports on the progress of such a model, providing model validation results and EOD prediction results.

The paper is organized as follows. Section 2 provides background material on batteries. Section 3 describes the model-based prognostics framework. Section 4 develops the electrochemistry-based battery model. Section 5 provides model validation and battery prognosis results. Section 6 concludes the paper.

2. BACKGROUND

A battery converts chemical energy into electrical energy, and often consists of many cells. A cell consists of a positive electrode and a negative electrode with electrolyte in which the ions can migrate. For Li-ion, a common chemistry is a positive electrode consisting of lithium cobalt oxide (Li_xCoO_2) and negative electrode of lithiated carbon (Li_xC). These active materials are bonded to metal-foil current collectors at both ends of the cell and electrically isolated by a microporous polymer separator film that is permeable to Li ions. The electrolyte enables lithium ions (Li^+) to diffuse between the positive and negative electrodes. The lithium ions insert or deinsert from the active material (known as intercalation and deintercalation) depending upon the electrode and whether the active process is charging or discharging.

A schematic of the battery during the discharge process is shown in Fig. 1. When fully charged, the active (mobile) Li ions reside in the negative electrode. Upon connecting a load to the battery, current is allowed to flow from the positive to the negative electrode. This is supported by the oxidation reaction (loss of electrons) in the negative electrode (acting as the anode):

$$\mathrm{Li}_x\mathrm{C} \xrightarrow{\text{discharge}} \mathrm{C} + x\mathrm{Li}^+ + x\mathrm{e}^-,$$

which results in the liberation of Li ions and electrons. The electrons move through the load from the negative to positive electrode, and the Li ions move in the same direction from the negative to the positive electrode through the separator. In the positive electrode (acting as the cathode), the reduction reaction (gain of electrons) takes place:

$$\mathrm{Li}_{1\,-\,x}\mathrm{CoO}_2 + x\mathrm{Li}^+ + x\mathrm{e}^- \xrightarrow{\text{discharge}} \mathrm{LiCoO}_2$$

During charging, a current source forces current to move from the negative to the positive electrode, as shown in Fig. 2. The active material in the positive electrode (acting as the anode) is oxidized and Li ions are de-intercalated with the corresponding reaction:

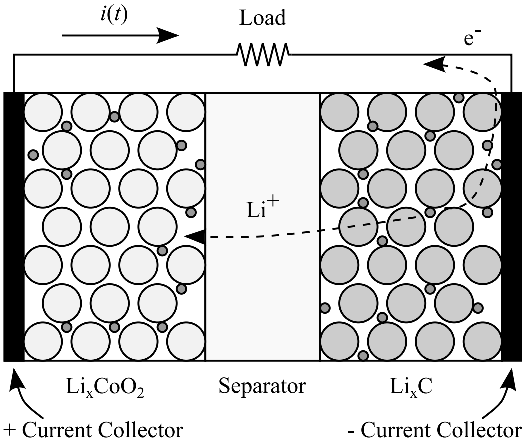


Figure 1. Li-ion battery during discharge.

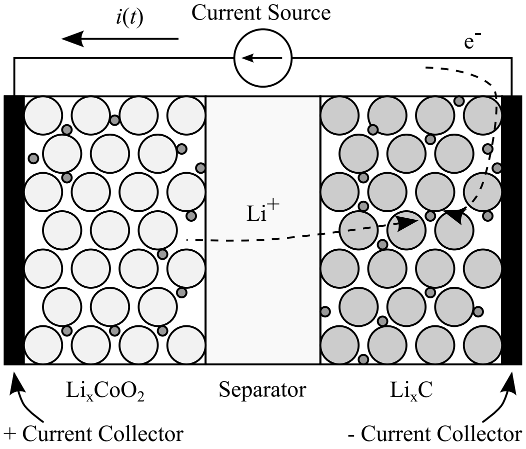


Figure 2. Li-ion battery during charge.

$$\mathrm{LiCoO}_2 \xrightarrow{\text{charge}} \mathrm{Li}_{1\,-\,x}\mathrm{CoO}_2 + x\mathrm{Li}^+ + x\mathrm{e}^-$$

that results in the loss of Li ions and electrons, which can then move to the negative electrode (acting as the cathode). The reduction reaction takes place in the negative electrode:

$$\mathrm{C} + x\mathrm{Li}^+ + x\mathrm{e}^- \xrightarrow{\text{charge}} \mathrm{Li}_x\mathrm{C}$$

Theoretically the cell has a voltage of around 4.2 V when fully charged. The terminal voltage of the battery rises/falls with a charge/discharge cycle, respectively. At the end of the charging/discharging period the battery voltage settles to a steady-state value that is a function of its state of charge (SOC). SOC is conventionally defined to be 1 when the battery is fully charged and 0 when the battery is fully discharged. Hence even in a degraded battery the SOC for maximum charge is always 1 by definition.

The charge/discharge dynamics of batteries can be characterized by taking voltage measurements under constant current charge/discharge conditions. The rate of charge/discharge is measured relative to the battery capacity C. For example, a $0.1C$ discharge rate for a 5 Ah battery is 0.5 A.

Charging typically consists of periods of constant current (CC) and/or constant voltage (CV) charging. At lower SOC, a CC charge is applied to bring the battery voltage to the CV level. Once the desired voltage level is achieved, the charger switches to CV mode and current decreases until 100% SOC is reached. The charger is not operated in CV mode at lower SOC levels since this will increase the amount of current flowing throught the battery while charging, leading to additional heat generation and side reactions taking place that may affect the battery life. The charge acceptance is the maximum permissible rate at which batteries can be charged. For Li-ion batteries the recommended charge acceptance is $C/3$.

The end of life of a battery depends on the chemistry, discharge-charge cycling, temperature, and storage conditions, among other factors. Atmospheric temperature affects the operating performance of the batteries. At low temperatures, ionic diffusion and migration could be hindered and also damaging side reactions like lithium plating may take place. High temperatures cause corrosion and generation of gases leading to an increase in internal pressure. As the battery ages, degradation results in the decrease in the observed battery capacity. This is primarily due to a loss of mobile ions due to parasitic or side reactions and an increase in the internal resistance. Internal resistance leads to ohmic losses that generate heat and accelerate the aging process. Some relevant physical aging mechanisms in the electrodes are:

1. Solid-electrolyte interface (SEI) layer growth: The negative electrode degrades with the growth of the SEI layer leading to an increase in the impedance. The layers are formed during cycling and storage at high temperatures and entrains the lithium.
2. Lithium corrosion: Lithium in the active carbon material of the negative electrode corrodes over time leading to degradation. This causes a decrease in the capacity due to irreversible loss of mobile lithium ions.
3. Lithium plating: At low temperatures, high charge rates and low cell voltages forms a plating layer on the negative electrode which leads to irreversible loss of lithium.
4. Contact loss: SEI layer disconnects from the negative electrode which leads to contact loss and an increase in impedance.

3. Battery Prognostics

In this section we describe the general model-based prognostics problem and an implementation architecture, and describe how it applies to batteries.

3.1. Problem Formulation

We assume the system model may be generally defined as

$$\mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{v}(k)), \tag{1}$$

$$\mathbf{y}(k) = \mathbf{h}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{n}(k)), \tag{2}$$

where k is the discrete time variable, $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\theta}(k) \in \mathbb{R}^{n_\theta}$ is the unknown parameter vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(k) \in \mathbb{R}^{n_v}$ is the process noise vector, $\mathbf{f}$ is the state equation, $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ is the output vector, $\mathbf{n}(k) \in \mathbb{R}^{n_n}$ is the measurement noise vector, and $\mathbf{h}$ is the output equation.[1]

In prognostics, we are interested in predicting the occurrence of some event E that is defined with respect to the states, parameters, and inputs of the system. We define the event as the earliest instant that some event threshold $T_E : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_u} \to \mathbb{B}$, where $\mathbb{B} \triangleq \{0, 1\}$ changes from the value 0 to 1. That is, the time of the event k_E at some time of prediction k_P is defined as

$$k_E(k_P) \triangleq \inf\{k \in \mathbb{N} \colon k \geq k_P \wedge T_E(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k)) = 1\}. \tag{3}$$

The time remaining until that event, Δk_E, is defined as

$$\Delta k_E(k_P) \triangleq k_E(k_P) - k_P. \tag{4}$$

In the context of systems health management, T_E is defined via a set of performance constraints that define what the acceptable states of the system are, based on $\mathbf{x}(k)$, $\boldsymbol{\theta}(k)$, and $\mathbf{u}(k)$ (Daigle & Goebel, 2013). For batteries, there are two types of predictions that we are interested in. The first is EOD, defined by a voltage threshold V_{EOD}. In this case, T_E is defined by $V < V_{EOD}$, that is, when the battery voltage is less than the cutoff voltage, EOD is reached and T_E evaluates to 1. The second type of prediction is EOL, which is typically defined by a lower bound on the effective battery capacity (Saha & Goebel, 2009). In this case, T_E is defined by $C < C_{EOL}$, where C is the measured battery capacity and C_{EOL} is the lower bound on capacity.

3.2. Prognostics Architecture

We adopt a model-based prognostics architecture (Daigle & Goebel, 2013), in which there are two sequential problems, (*i*) the *estimation* problem, which requires determining a joint state-parameter estimate $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k))$ based on the history of observations up to time k, $\mathbf{y}(k_0{:}k)$, and (*ii*) the *prediction* problem, which determines at k_P, using $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k))$, a probability distribution

[1] Here, we use bold typeface to denote vectors, and use n_a to denote the length of a vector $\mathbf{a}$.

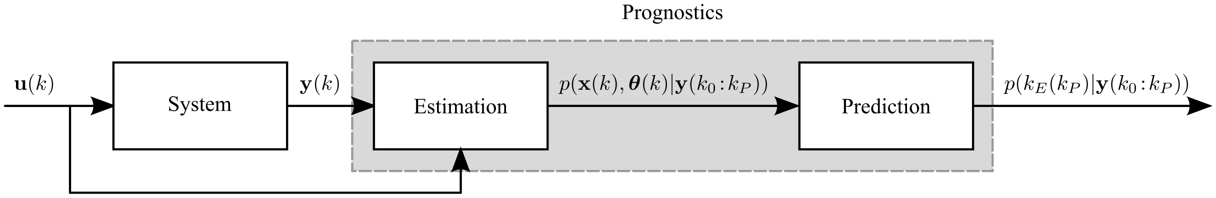

Prognostics
$\mathbf{u}(k)$
System
$\mathbf{y}(k)$
Estimation
$p(\mathbf{x}(k),\boldsymbol{\theta}(k)|\mathbf{y}(k_0:k_P))$
Prediction
$p(k_E(k_P)|\mathbf{y}(k_0:k_P))$

Figure 3. Prognostics architecture.

$p(k_E(k_P)|\mathbf{y}(k_0:k_P))$. The distribution for Δk_E can be trivially computed from $p(k_E(k_P)|\mathbf{y}(k_0:k_P))$ by subtracting k_P. Both steps require a dynamic model in the form of equations 1 and 2.

The prognostics architecture is shown in Fig. 3. In discrete time k, the system is provided with inputs $\mathbf{u}_k$ and provides measured outputs $\mathbf{y}_k$. The estimation module uses this information, along with the system model, to compute an estimate $p(\mathbf{x}(k),\boldsymbol{\theta}(k)|\mathbf{y}(k_0:k))$. The prediction module uses the joint state-parameter distribution and the system model, along with hypothesized future inputs, to compute the probability distribution $p(k_E(k_P)|\mathbf{y}(k_0:k_P))$ at given prediction times k_P.

In the application of prognostics in this example, we do not account for any uncertainty except for that provided in the state estimate, as our focus is on determining how accurate the model can predict EOD given precise information about the future. A more general prognostics architecure that accounts for these additional sources of uncertainty is described in (Daigle, Saxena, & Goebel, 2012; Sankararaman, Daigle, Saxena, & Goebel, 2013).

4. Battery Modeling

In order to predict end-of-discharge as defined by a voltage cutoff, the battery model must compute the voltage as a function of time given the current drawn from the battery. There are several electrochemical processes that contribute to the cell's potential that make this a difficult problem. We focus here on a lumped-parameter ordinary differential equations form, so it is efficient and usable for on-line prognostics, yet still considers the main electrochemical processes. We focus here specifically on Li-ion 18650 batteries with an average nominal voltage of 3.7V and nominal capacity of 2200mAh, however, the model is still general enough that with some modifications it may be applied to different battery chemistries.

The voltages of a battery are summarized in Fig. 4 (adapted from (Rahn & Wang, 2013)). The overall battery voltage $V(t)$ is the difference between the potential at the positive current collector, $\phi_s(0,t)$, and the negative current collector, $\phi_s(L,t)$, minus resistance losses at the current collectors (not shown in the diagram). As shown in the figure, the potentials vary with the distance $d \in [0,L]$, because the loss varies with distance from the current collectors.

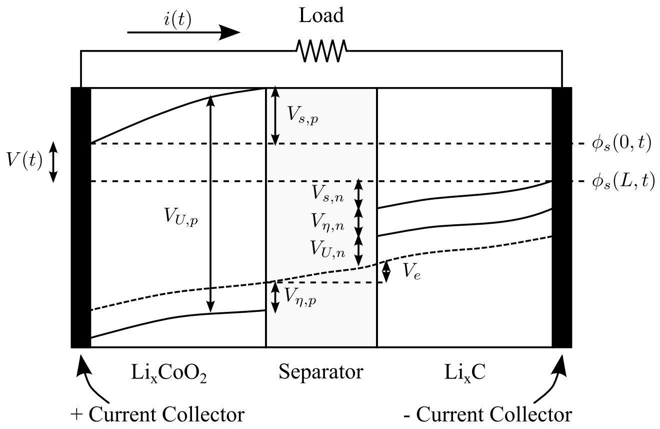


Figure 4. Battery voltages.

The potentials at the current collectors are described by several voltage terms. At the positive current collector is the equilibrium potential $V_{U,p}$. This voltage is then reduced by $V_{s,p}$, due to the solid-phase ohmic resistance, and $V_{\eta,p}$, the surface overpotential. The electrolyte ohmic resistance then causes another drop V_e. At the negative electrode, there is a drop $V_{\eta,n}$ due to the surface overpotential, and a drop $V_{s,n}$ due to the solid-phase resistance. The voltage drops again due to the equilibrium potential at the negative current collector $V_{U,n}$. We describe each of these voltage terms in turn.

4.1. Equilibrium Potential

In an ideal battery, the equilibrium potential of an electrode is described by the Nernst equation:

$$V_{U,i} = U_0 + \frac{RT}{nF}\ln\left(\frac{\gamma_{\beta,i}x_{\beta,i}}{\gamma_{\alpha,i}x_{\alpha,i}}\right), \tag{5}$$

or, equivalently,

$$V_{U,i} = U_0 + \frac{RT}{nF}\ln\left(\frac{x_{\beta,i}}{x_{\alpha,i}}\right) + \frac{RT}{nF}\ln\left(\frac{\gamma_{\beta,i}}{\gamma_{\alpha,i}}\right), \tag{6}$$

where i refers to the electrode (n for negative or p for positive), U_0 is a reference potential, R is the universal gas constant, T is the electrode temperature, n is the number of electrons transferred in the reaction ($n = 1$ for Li-ion), F is Faraday's constant, x is the mole fraction, where α refers to the lithium-intercalated host material and β to the unoccupied host material, and γ is the activity coefficient (Karthikeyan et al., 2008). When discharging, Li ions move out of the negative electrode and into the positive electrode, so $x_{\alpha,n}$ and $x_{\beta,p}$ decrease while $x_{\beta,n}$ and $x_{\alpha,p}$ increase. As a result, $V_{U,p} - V_{U,n}$ will decrease. In an ideal condition the activity coefficients are unity and so the last term disappears, resulting in the classical form of the Nernst equation.

In the case of a Li-ion battery, $x_{\beta,i} = 1 - x_{\alpha,i}$, so we can rewrite the equation making this substitution. We then define x_i as $x_{\alpha,i}$ and drop the α subscript:

$$V_{U,i} = U_0 + \frac{RT}{nF} \ln \left(\frac{1 - x_i}{x_i} \right) + V_{\text{INT},i}, \tag{7}$$

where $V_{\text{INT},i}$ is the activity correction term (0 in the ideal condition). For Li_xCoO_2, x must be at least 0.4; Li cannot be reversibly removed beyond that (Karthikeyan et al., 2008).

We let q_i represent the amount of Li ions in electrode i, as measured in Coulombs. Recall that the flow of Li ions moves opposite to the flow of current. Therefore q_i changes in the same direction as x_i, and we may define x_i based on q_i using

$$x_i = \frac{q_i}{q^{\max}}, \tag{8}$$

where $q^{\max} = q_p + q_n$ refers to the total amount of available (i.e., mobile) Li ions. It follows then that $x_p + x_n = 1$. When fully charged, $x_p = 0.4$ and $x_n = 0.6$. When fully discharged, $x_p = 1$ and $x_n = 0$.

Batteries are nonideal and therefore the assumption of unity activity coefficients is not applicable. As a result, battery models often resort to curve-fitting for modeling the equilibrium potential. The equilibrium potential can be obtained by discharging the battery at a very slow rate, such that a concentration gradient will not build up, temperature is steady, and voltage drops due to internal resistances and other overpotentials are negligible. Fig. 5 shows the measured voltage ($V_{U,p} - V_{U,n}$) as a function of mole fraction against that predicted by the Nernst equation. Clearly, the Nernst equation cannot fit this case so it must be nonideal.

The activity coefficient terms are related to excess Gibbs free energy and can be captured using the Redlich-Kister expansion (Karthikeyan et al., 2008). We then have

$$\begin{aligned} V_{\text{INT},i} &= \frac{RT}{nF} \ln \left(\frac{\gamma_{\beta,i}}{\gamma_{\alpha,i}} \right) && (9) \\ &= \frac{1}{nF} \left(\sum_{k=0}^{N_i} A_{i,k} \left((2x_i - 1)^{k+1} - \frac{2x_i k(1 - x_i)}{(2x_i - 1)^{1-k}} \right) \right). && (10) \end{aligned}$$

The number of terms N_i in the above expansion and the fitting parameters $A_{i,k}$ must be identified. Fig. 6 shows the modified equation fitted to the measured equilibrium potential for $N_p = 12$ and $N_n = 0$. Clearly, the modified equation much more realistically reflects the true voltage curve. The identified parameters are given in Table 1 and were determined using the Nelder-Mead simplex method. The identified parameters represent those achieving a least-squares (local) error minimum, which is satisfactory for our purposes.

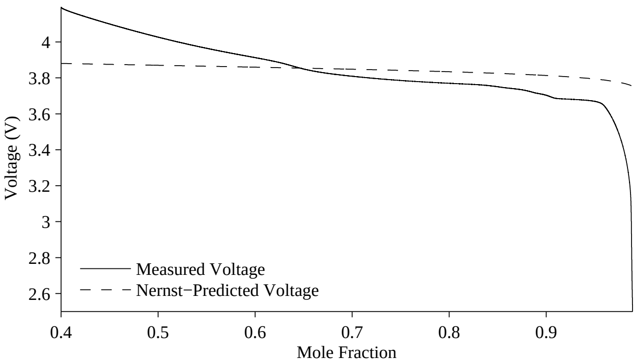


Figure 5. Measured and Nernst-predicted open-circuit potential.

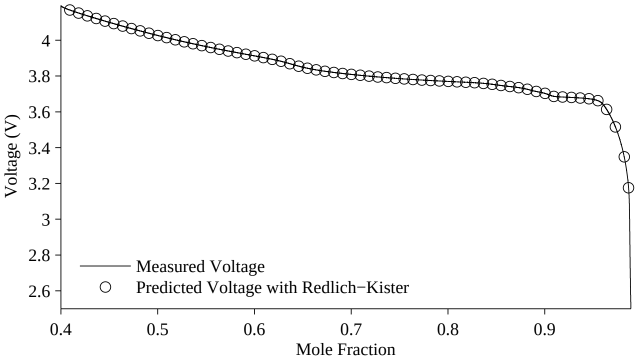


Figure 6. Measured and Redlich-Kister-predicted open-circuit potential.

4.2. Concentration Overpotential

When a battery is discharged, the reactions take place at the surface of the electrode and this results in a concentration gradient. The concept is shown in Fig. 7. In the bulk volume, the concentration is nearly even, but close to the electrode the concentration changes drastically. To accommodate this, one may split the total volume into two individual control volumes (CVs), one for the bulk (with subscript b) and one for the surface (with subscript s).

For the volumes, the concentration of Li ions is computed as

$$c_{b,i} = \frac{q_{b,i}}{v_{b,i}} \tag{11}$$

$$c_{s,i} = \frac{q_{s,i}}{v_{s,i}}, \tag{12}$$

Table 1. Battery Model Parameters

Parameter	Value
$U_{0,p}$	4.03 V
$A_{p,0}$	-33642.23 J/mol
$A_{p,1}$	0.11 J/mol
$A_{p,2}$	23506.89 J/mol
$A_{p,3}$	-74679.26 J/mol
$A_{p,4}$	14359.34 J/mol
$A_{p,5}$	307849.79 J/mol
$A_{p,6}$	85053.13 J/mol
$A_{p,7}$	-1075148.06 J/mol
$A_{p,8}$	2173.62 J/mol
$A_{p,9}$	991586.68 J/mol
$A_{p,10}$	283423.47 J/mol
$A_{p,11}$	-163020.34 J/mol
$A_{p,12}$	-470297.35 J/mol
$U_{0,n}$	0.01 V
$A_{n,0}$	86.19 J/mol

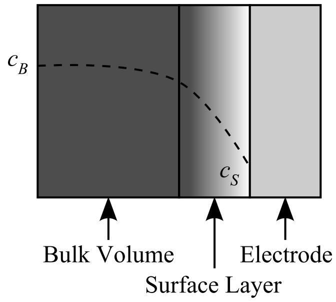


Figure 7. Concentration gradient.

where, for CV v in electrode i, $c_{v,i}$ is the concentration and $v_{v,i}$ is the volume. We define $v_i = v_{b,i} + v_{s,i}$. Note now that the following relations hold:

$$q_p = q_{s,p} + q_{b,p} \tag{13}$$

$$q_n = q_{s,n} + q_{b,n} \tag{14}$$

$$q^{\max} = q_{s,p} + q_{b,p} + q_{s,n} + q_{b,n}. \tag{15}$$

As the battery discharges, Li ions must move out of the surface layer at the negative electrode, through the bulk, and to the surface layer at the positive electrode, in order to match the flow of electrons. Li ions also move from the bulk CV to the surface due to the concentration gradient. The diffusion rate from the bulk to the surface is expressed as

$$\dot{q}_{bs,i} = \frac{1}{D}(c_{b,i} - c_{s,i}), \tag{16}$$

where D is the diffusion constant.

The q variables are described as

$$\dot{q}_{s,p} = i_{app} + \dot{q}_{bs,p} \tag{17}$$

$$\dot{q}_{b,p} = -\dot{q}_{bs,p} + i_{app} - i_{app} \tag{18}$$

$$\dot{q}_{b,n} = -\dot{q}_{bs,n} + i_{app} - i_{app} \tag{19}$$

$$\dot{q}_{s,n} = -i_{app} + \dot{q}_{bs,n}, \tag{20}$$

where i_{app} is the applied electric current. Initially, $c_{s,i} = c_{b,i}$ and so diffusion is zero. Once discharge begins, ions move from the surface layer of the negative electrode to the bulk volume of the negative electrode, through the separator to the bulk volume of the positive electrode and then on to the surface layer of the positive electrode. If the applied current is larger than the diffusion rate, then a concentration gradient will build up between the surface and bulk volumes. Once discharge stops, then only diffusion is active, and the concentrations will even out, causing a rise in voltage (because the concentration lowers in the surface layer of the positive electrode, raising the voltage, and the concentration rises in the surface layer of the negative electrode, raising the voltage).

The concentration overpotential is the difference in voltage between the two CVs due to the difference in concentration. Using the expression for equilibrium potential, we can compute the potential for the bulk volume and the potential for the surface layer; the difference between them is the concentration overpotential. We can explicitly account for the concentration overpotential simply by using as the expression for equilibrium potential, the equilibrium potential of the surface layer, i.e.,

$$V_{U,i} = U_0 + \frac{RT}{nF} \ln\left(\frac{1 - x_{s,i}}{x_{s,i}}\right) + V_{\text{INT},i}, \tag{21}$$

where $x_{s,i}$ is computed using

$$x_{s,i} = \frac{q_{s,i}}{q_{s,i}^{\max}}, \tag{22}$$

and

$$q_{s,i}^{\max} = q^{\max} \frac{v_{s,i}}{v_i} \tag{23}$$

The mole fraction in the surface changes faster than the mole fraction at the bulk, causing a larger concentration gradient for larger applied current and smaller diffusion rates. The observed voltage depends only on what is happening in the surface layer, not the bulk. When discharge stops, the surface layer becomes like the bulk, accounting for the recovery in voltage. The battery may then be discharged further since the surface layers were replenished with ions (from the bulk) in the negative electrode.

4.3. Ohmic Overpotential

The voltage drops due to the solid-phase ohmic resistances, the electrolyte ohmic resistance, and the resistances at the current collectors can be treated as constant resistances and lumped together:

$$V_o = V_{s,p} + V_{s,n} + V_e + V_{cc,p} + V_{cc,n} \tag{24}$$

$$= i_{app}(R_{s,p} + R_{s,n} + R_e + R_{cc,p} + R_{cc,n}) \tag{25}$$

$$= i_{app}R_o, \tag{26}$$

where V_{cc} is the voltage drop over the current collectors, $R_{cc,i}$ is the resistance at the current collector for electrode i.

4.4. Surface Overpotential

The overpotentials due to charge transfer resistance and SEI kinetics are described by the Butler-Volmer equation. For small currents the overpotential is linear with current and for larger currents it is proportional to the log of the current, and these two characteristics are combined in the Butler-Volmer equation. The equation is given as

$$J_i = J_{i0}\left(\exp\left(\frac{(1-\alpha)F}{RT}V_{\eta,i}\right) - \exp\left(-\frac{\alpha F}{RT}V_{\eta,i}\right)\right), \tag{27}$$

where α is the symmetry factor, J_i is the current density, and J_{i0} is the exchange current density. The current densities are defined as

$$J_i = \frac{i}{S_i} \tag{28}$$

$$J_{i0} = k_i(1 - x_{s,i})^{\alpha}(x_{s,i})^{1-\alpha}, \tag{29}$$

where k_i is a lumped parameter of several constants including a rate coefficient, electrolyte concentration, and maximum ion concentration. The voltage drop takes place at the electrode surface, so $x_{s,i}$ is used in computing the exchange current density.

In the case of Li ions, the symmetry factor is 0.5, so the Butler-Volmer equation can be simplified and written in terms of voltage as

$$V_{\eta,i} = \frac{RT}{F\alpha}\text{arcsinh}\left(\frac{J_i}{2J_{i0}}\right). \tag{30}$$

4.5. State of Charge

As discussed in Section 2, the SOC of a battery is conventionally defined to be 1 when the battery is fully charged and 0 when the battery is fully discharged. In this model, it is analogous to the mole fraction x_n, but scaled from 0 to 1. There is a difference here between nominal SOC and *apparent* SOC. Nominal SOC would be computed based on the combination of the bulk and surface layer CVs in the negative electrode, whereas apparent SOC would be computed based only on the surface layer. That is, a battery can be discharged at a given rate, and reach the voltage cutoff, i.e., apparent SOC is then 0. But, once the concentration gradient settles out, the surface layer will be partially replenished and the battery can be discharged further, i.e, apparent SOC increases whereas nominal SOC remains the same.

Nominal (n) and apparent (a) SOC can then be defined using

$$SOC_n = \frac{q_n}{0.6q^{\max}} \tag{31}$$

$$SOC_a = \frac{q_{s,n}}{0.6q^{\max_{s,n}}}, \tag{32}$$

where $q^{\max_{s,n}} = q^{\max}\frac{v_{s,n}}{v_n}$. The factor $1/0.6$ comes from the fact that the mole fraction at the positive electrode cannot go below 0.4 (as described in Section 4.1), therefore SOC of 1 corresponds to the point where $q_n = 0.6q^{\max_{s,n}}$.

4.6. Battery Voltage

Now that each of the voltage drops in Fig. 4 have been defined, battery voltage can be expressed as follows.

$$V = V_{U,p} - V_{U,n} - V_o - V_{\eta,p} - V_{\eta,n}. \tag{33}$$

Voltages in the battery are not observed to change instantaneously, i.e., the voltage changes occur smoothly. When discharge completes, for example, the voltage rises slowly as the surface layers move to the concentrations of the bulk volumes, as caused by diffusion. In addition to this, there are transients associated with V_o and the $V_{\eta,i}$ terms. To take this into account in a simple way, we compute voltage using

$$V = V_{U,p} - V_{U,n} - V'_o - V'_{\eta,p} - V'_{\eta,n}, \tag{34}$$

where

$$\dot{V}'_o = (V_o - V'_o)/\tau_o \tag{35}$$

$$\dot{V}'_{\eta,p} = (V_{\eta,p} - V'_{\eta,p})/\tau_{\eta,p} \tag{36}$$

$$\dot{V}'_{\eta,n} = (V_{\eta,n} - V'_{\eta,n})/\tau_{\eta,n}, \tag{37}$$

where the τ parameters are empirical time constants.

The model contains as states $\mathbf{x}$, $q_{s,p}$, $q_{b,p}$, $q_{b,n}$, $q_{s,n}$, V'_o, $V'_{\eta,p}$, and $V'_{\eta,n}$. The single model output is V.

Identified model parameters are given in Table 2. Some parameters are defined by the battery dimensions and chemistry. The remaining parameters are estimated based on data, as with the parameters in Table 1.

The measured and predicted discharge curves for a constant 2 A discharge are shown in Fig. 8. The model fits very well to the measured values. The accuracy towards the end of discharge is most sensitive to the Redlich-Kister parameters, the diffusion constant, and the volume of the surface layer. The

Table 2. Battery Model Parameters

Parameter	Value
$q^{\max}$	1.32×10^4 C
R	8.314 J/mol/K
T	292 K
F	96487 C/mol
n	1
D	7.0×10^6 mol s/C/m^3
τ_o	10 s
α	0.5
R_o	$0.085\ \Omega$
S_p	2×10^{-4} m^2
k_p	2×10^4 A/m^2
$v_{s,p}$	2×10^{-6} m^3
$v_{b,p}$	2×10^{-5} m^3
$\tau_{\eta,p}$	90 s
S_n	2×10^{-4} m^2
k_n	2×10^4 A/m^2
$v_{s,n}$	2×10^{-6} m^3
$v_{b,n}$	2×10^{-5} m^3
$\tau_{\eta,n}$	90 s

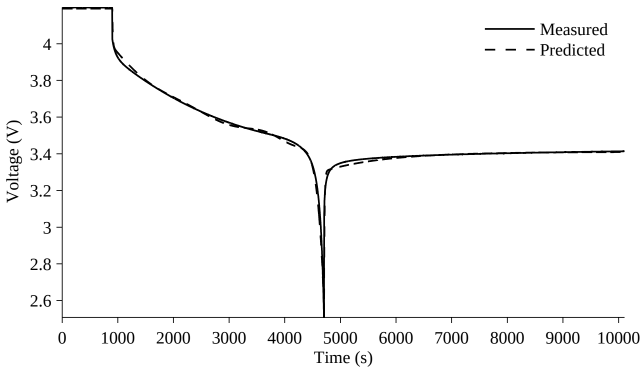


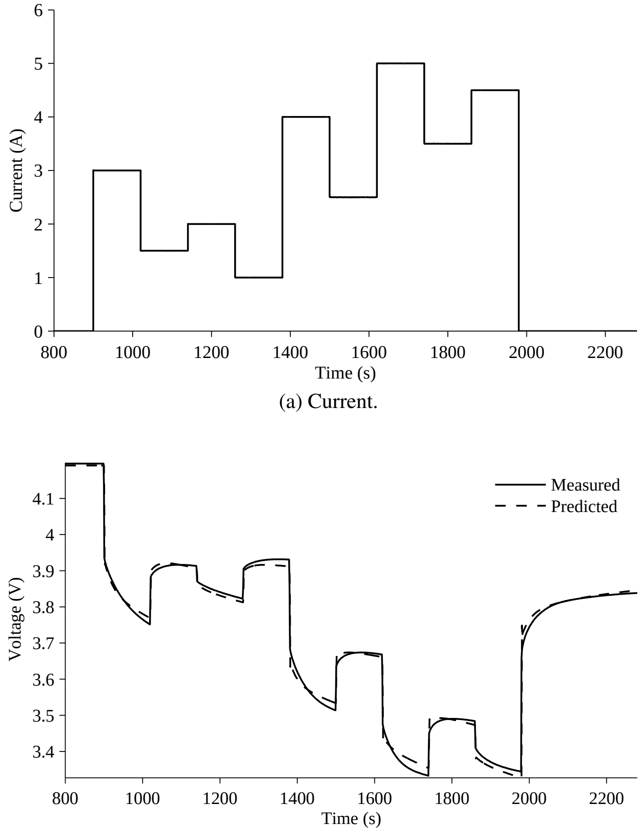
Figure 8. Measured and predicted 2 A discharge curves.

predicted end-of-discharge time can be finely tuned most easily by adjusting the time at which the amount of Li ions in the surface layer of the negative electrode hits zero and the surface layer of the postive electrode becomes full, because this is when the equilibrium voltage of these layers nears a combined 2.5 V. This is accomplished by adjusting the diffusion constant.

Model validation for a variable loading scenario is shown in Fig. 9. As shown in Fig. 9a, the load changes every 2 minutes, resulting in corresponding changes in voltage. Fig. 9b shows that the voltage predictions are fairly accurate in response to changes in load. Some errors are still present that may possibly be accounted for by including temperature effects.

4.7. Battery Aging

As described in Section 2, battery aging manifests in two major ways. The first is a loss of capacity due to parasitic and side reactions that result in a loss of active (mobile) Li ions. The second is an increase in internal resistance due to SEI layer growth and other factors. Capacity is typically measured with respect to a reference current and the associated EOD point as defined by a voltage cutoff. A decreased capacity will result in the voltage hitting the cutoff earlier. An increase in resistance will lower the voltage overall, so also result in hitting the cutoff earler. Therefore both these aging mechanisms result in a loss of measured capacity.



(a) Current.

(b) Measured and predicted variable loading discharge curves.

Figure 9. Model validation for variable loading.

Fig. 10 shows 2 A discharge curves with the battery at different ages. The arrows in the figure illustrate the direction the curves move as the battery ages. The EOD point moves earlier in time due to diminished capacity. The voltage drops down during discharge due to increased resistance. Steady-state voltage after discharge increases because the battery is not discharged as fully due to the increased resistance and the battery reaching EOD before then.

In the battery model, the total available charge in the battery is represented through $q^{\max}$. Therefore, the loss of active material can be represented in the model through a change in $q^{\max}$. Fig. 11a shows how the discharge curve changes as $q^{\max}$ is decreased by 1% with each new discharge. We can see a slight decrease in voltage, which is due to the Butler-Volmer term,

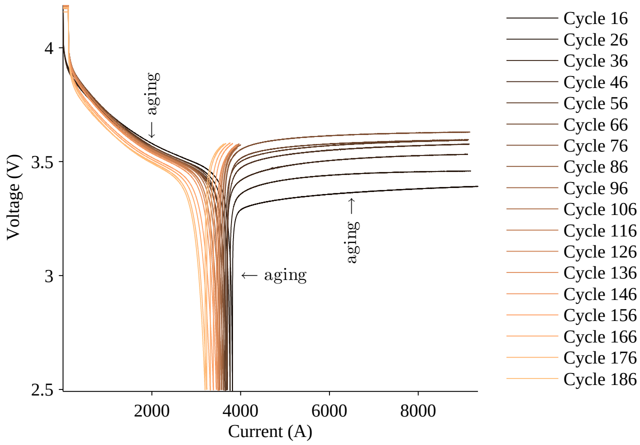


Figure 10. Measured 2 A discharge curves with battery aging.

which is dependent on mole fraction that is computed using $q^{\max}$. The EOD point moves earlier due to the discharge point being reached with less charge extracted. Steady-state voltage after discharge increases since the cutoff is reached with less relative charge extracted.

One way to validate this is by looking at the equilbrium potential plots as the battery ages, because the resistances are negligible. Fig. 12 shows the measured equilibrium potential of the battery ($V_{U,p} - V_{U,n}$) after 10 discharge cycles and after 30 discharge cycles. The curves are exactly the same, only the time scale changes; there is a 20 minute difference in EOD. Since equilibrium potential is a function only of mole fraction, and mole fraction is computed as $q/q^{\max}$, then decreasing $q^{\max}$ would change the time scale in this way.

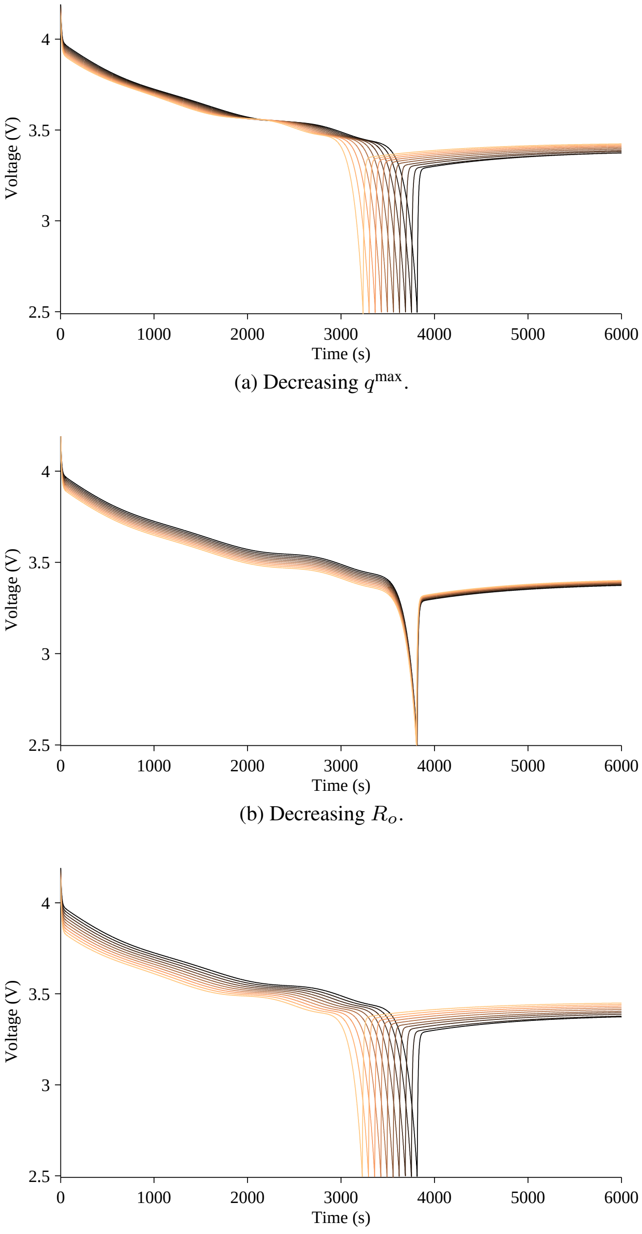
Increase in internal resistance can be captured through an increase in the R_o parameter. Fig. 11b shows how the discharge curve changes as R_o is increased by 5% with each new discharge. The voltage drops, but the change in EOD is not significant, because the dynamics near EOD are dominated mainly by the equilibrium potential contribution with some contribution from the Butler-Volmer dynamics. The change in charge extracted is thereby also not significant so changes in steady-state voltage after discharge are very small.

From Figs. 11a and 11b, it is clear that changes in both $q^{\max}$ and R_o are required to capture how the discharge curve changes with aging. Fig. 11c shows the combined effects, with $q^{\max}$ decreasing by 1% and R_o increasing by 5% with each new discharge. The qualitative changes observed in Fig. 10 are now reproduced.

5. Application to Prognostics

With an accurate model and known future inputs to a system, prognostics should in turn be very accurate. In this section we demonstrate battery prognostics with the new model developed in this paper. We use the architecture described in Section 3. As an estimation algorithm, we use the unscented Kalman filter (UKF) with the battery model; see (Julier & Uhlmann, 1997, 2004) for details on the filter and (Daigle, Saha, & Goebel, 2012; Daigle, Saxena, & Goebel, 2012) for its application to prognostics. The UKF operates on a set of determinstically selected samples, called *sigma points*, that are used to represent the joint state-parameter distribution $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k))$.

For the prediction algorithm, we perform a simple simulation



(a) Decreasing $q^{\max}$.

(b) Decreasing R_o.

(c) Decreasing $q^{\max}$ and R_o.

Figure 11. Simulated 2 A discharge curves with aging. Curves for later cycles are shown in lighter colors.

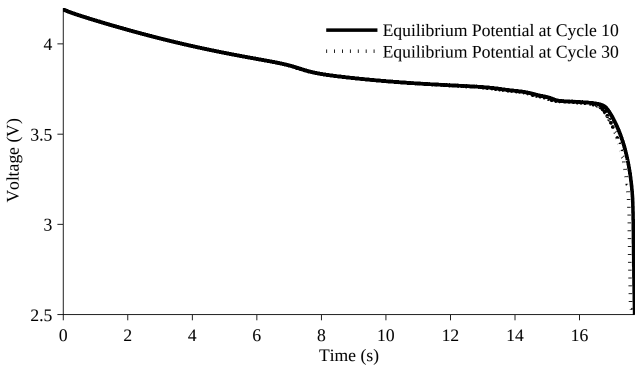


Figure 12. Shifting of equilibrium potentials due to aging.

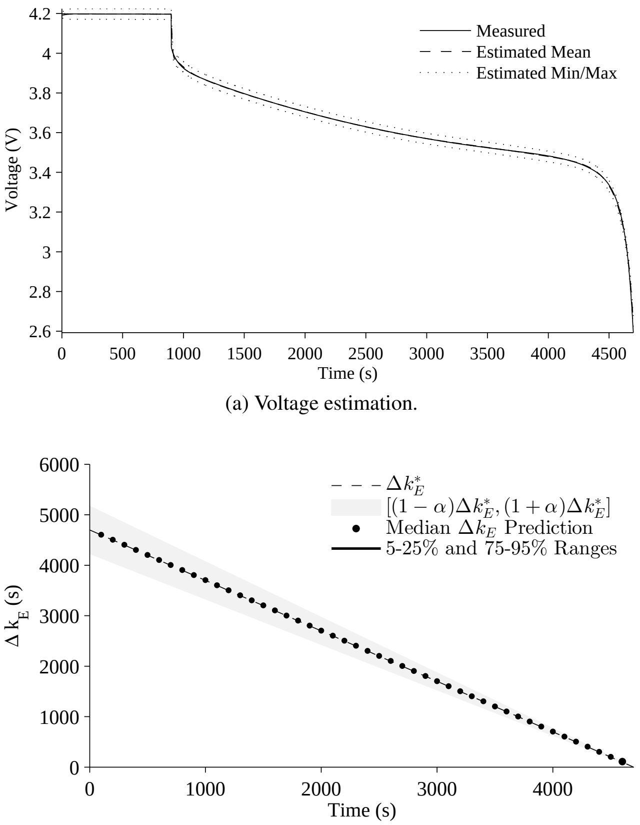
as described in (Daigle & Goebel, 2013). Each sigma point is simulated forward using the model until EOD is reached; from the corresponding EODs for each sigma point we can construct the EOD distribution. In this work, we assume that the future inputs (i_{app}) are known, so the only uncertainty present in the prediction is that related to the model. We use $V_{\text{EOD}} = 2.6\ V$ as the voltage cutoff defining EOD.

As performance metrics, we use percent root mean square error (PRMSE) for estimation accuracy, and for prediction we use relative accuracy (RA) and relative standard deviation (RSD). We plot results in α-λ plots, where α (e.g. 10%) defines an accuracy cone around the ground truth, and λ is a time point (Saxena, Celaya, Saha, Saha, & Goebel, 2010).

Fig. 13 shows prognosis results from a 2 A discharge, assuming the future inputs (i_{app}) are known. As shown in Fig. 13a, the UKF tracks very well the measured voltage, since the underlying model is very accurate. Prediction results are, in turn, also very accurate and with very little uncertainty (the only uncertainty is related to the state estimation uncertainty), as shown in Fig. 13b. Overall, RA averages over all prediction points (every 100 s) to 99.38%, and RSD to 0.75%. Since the spread is so small, the 5%–25% and 75%–95% ranges cannot be seen on the plot.

Fig. 14 shows prognosis results from a 3 A discharge. In the open loop, the model slightly underestimates EOD, as shown in Fig. 14b. Even though the UKF tracks well, once a prediction is made the model is in the open loop, and since the model predicts an earlier EOD for the given load there is a bias in the predictions. The UKF helps keep the state estimate accurate and so reduces the error compared to using the model itself.

Fig. 15 shows prognosis results from a variable loading discharge, with EOD being defined in this case as 3.35 V. In the open loop, the model slightly overestimates EOD, as shown in Fig. 15b. The UKF tracks well and corrects for some of the model errors (see Fig. 9b). The model is more accurate at predicting EOD when defined towards the end of a discharge rather than in the middle of a discharge, so if the voltage threshold is defined to be higher, the predictions will be less accurate. In this case RA averages to 88.41%, and RSD to 2.13%. RSD is larger because the UKF needs to account for the additional model uncertainty since the model is less accurate in variable-loading scenarios.



(a) Voltage estimation.

(b) EOD prediction

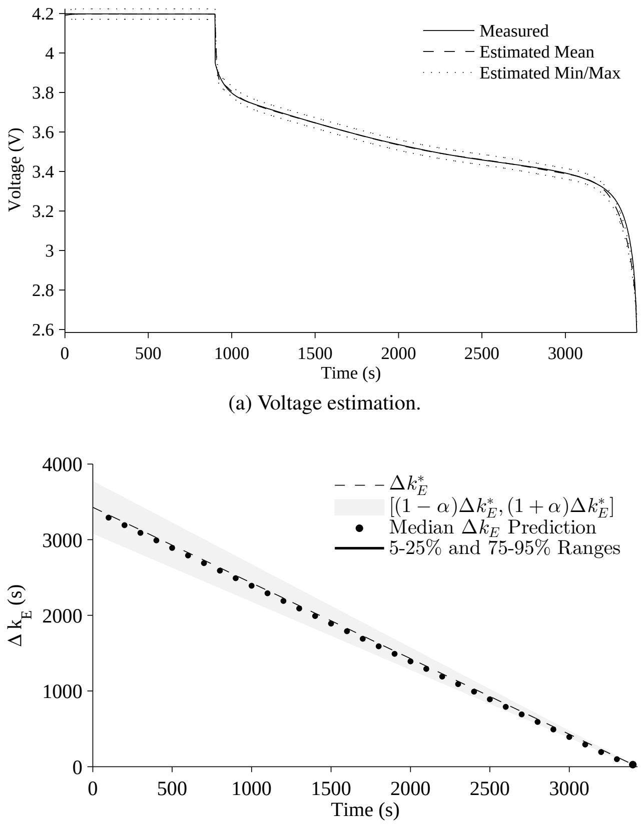
Figure 13. Prognostics results for 2 A discharge.

Additional results are shown in Table 3. PRMSE is very low as the UKF is able to track the voltage well in all cases. The predictions are also very accurate, with average RA above 90%, and precise, with RSD around 1% or lower. Although overall very good, the results are best around the 2 A discharge, suggesting that further model improvements (e.g., including temperature effects) may be necessary.

It is also worth mentioning that the model is not computationally intensive. The model is implemented with a fixed-step discrete-time simulation with a step size of 1 s. It takes on the order of 1×10^{-4} s to simulate 1 s of real time.

6. Conclusions

In this paper, we presented a new model for battery prognostics based on the underlying physics. Using equations based in electrochemistry, we developed a model that is both accu-



(a) Voltage estimation.

(b) EOD prediction

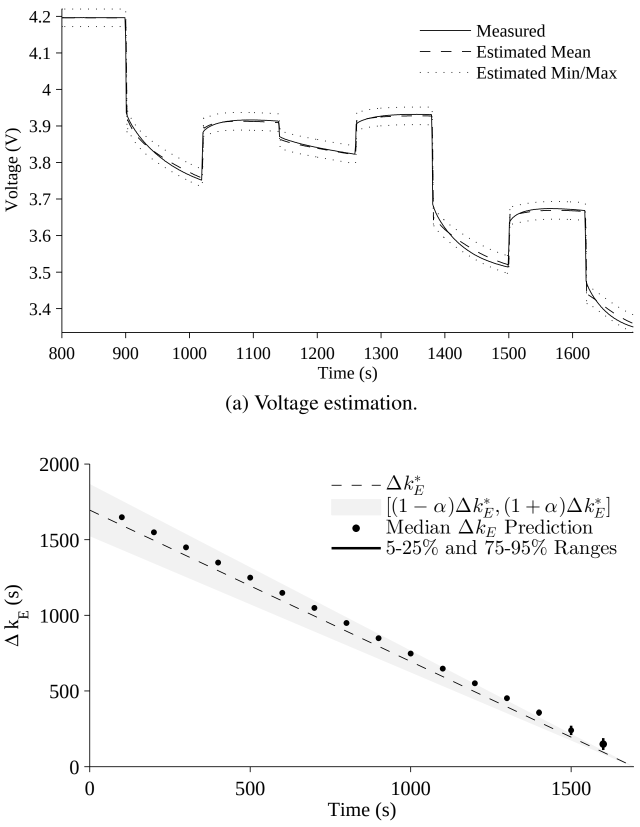
Figure 14. Prognostics results for 3 A discharge.

Table 3. Prognostics Performance

i_{app}	PRMSE	$\overline{\text{RA}}$	$\overline{\text{RMAD}}_{RUL}$
1.0	0.19	92.77	1.07
1.5	0.17	96.02	0.88
2.0	0.17	99.38	0.75
2.5	0.26	97.75	0.82
3.0	0.41	96.08	0.92

rate and efficient. As a result, prognostics results for EOD prediction are very accurate, with the uncertainty associated with the model remaining very small. Such a model has uses in other areas as well, such as control.

The model described here can also be applied to prognostics of battery packs. There are two approaches here: either each battery is modeled individually, or the batteries of the pack are lumped into an equivalent single-cell model. In the latter case, the prognostics framework as described here applies directly. In the former case, T_E must be redefined on a system-level (Daigle, Bregon, & Roychoudhury, 2012), such that the battery pack is said to be at EOD (or EOL) when any single battery in the pack is at EOD (or EOL).



(a) Voltage estimation.

(b) EOD prediction

Figure 15. Prognostics results for variable loading discharge.

The current model is limited in several ways that provide the basis for future work. For one, we have not described how the internal battery temperature changes over time. This is important because many of the potentials in the battery are functions of temperature. Further, diffusion rates should be impacted by temperature. Although we have described how aging can be represented within the model, we do not yet have descriptions of how these age-related parameters change over time, i.e., we require damage progression equations. Much work has already been done in this area but not with prognostics requirements and applications in mind. With such models, accurate end-of-life predictions can be made. Upon adding these missing elements, application of this modeling framework to different battery chemistries is also of interest.

ACKNOWLEDGMENT

This work was funded in part by the NASA System-wide Satefy Assurance Technologies (SSAT) project under the Aviation Safety (AvSafe) Program of the Aeronautics Research Mission Directorate (ARMD), by the NASA Automated Cryogenic Loading Operations (ACLO) project under the Office of the Chief Technologist (OCT) of Advanced Exploration Systems (AES), and by the Advanced Ground Systems Maintenance (AGSM) Project under the Ground Systems Development and Operations program. The authors also acknowledge Brian Bole, Georgia Institute of Technology, for obtaining experimental data.

References

Ceraolo, M. (2000, November). New dynamical models of lead-acid batteries. *IEEE Transactions on Power Systems*, *15*(4), 1184–1190.

Chen, M., & Rincon-Mora, G. (2006, June). Accurate electrical battery model capable of predicting runtime and I-V performance. *IEEE Transactions on Energy Conversion*, *21*(2), 504 - 511.

Daigle, M., Bregon, A., & Roychoudhury, I. (2012, September). A distributed approach to system-level prognostics. In *Annual conference of the prognostics and health management society* (p. 71-82).

Daigle, M., & Goebel, K. (2013, May). Model-based prognostics with concurrent damage progression processes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(4), 535-546.

Daigle, M., Saha, B., & Goebel, K. (2012, March). A comparison of filter-based approaches for model-based prognostics. In *Proceedings of the 2012 IEEE aerospace conference.*

Daigle, M., Saxena, A., & Goebel, K. (2012, September). An efficient deterministic approach to model-based prediction uncertainty estimation. In *Annual conference of the prognostics and health management society* (p. 326-335).

Doyle, M., Fuller, T. F., & Newman, J. (1993, June). Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *Journal of the Electrochemical Society*, *140*(6), 1526-1533.

Julier, S. J., & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of the 11th international symposium on aerospace/defense sensing, simulation and controls* (pp. 182–193).

Julier, S. J., & Uhlmann, J. K. (2004, mar). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401–422.

Karthikeyan, D. K., Sikha, G., & White, R. E. (2008). Thermodynamic model development for lithium intercalation electrodes. *Journal of Power Sources*, *185*(2), 1398–1407.

Lee, K. J., Smith, K., Pesaran, A., & Kim, G. H. (2013, November). Three dimensional thermal-, electrical-, and electrochemical-coupled model for cylindrical wound large format lithium-ion batteries. *Journal of Power Sources*(1), 20-32.

Newman, J., & Tiedemann, W. (1975). Porous-electrode theory with battery applications. *AIChE Journal*, *21*(1), 25-41.

Ning, G., & Popov, B. N. (2004). Cycle life modeling of lithium-ion batteries. *Journal of The Electrochemical Society*, *151*(10), A1584–A1591.

Rahn, C. D., & Wang, C.-Y. (2013). *Battery systems engineering*. Wiley.

Ramadesigan, V., Northrop, P. W., De, S., Santhanagopalan, S., Braatz, R. D., & Subramanian, V. R. (2012). Modeling and simulation of lithium-ion batteries from a systems engineering perspective. *Journal of The Electrochemical Society*, *159*(3), R31–R45.

Rong, P., & Pedram, M. (2006). An analytical model for predicting the remaining battery capacity of lithium-ion batteries. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, *14*(5), 441–451.

Ross, P. (2013). Boeing's battery blues. *IEEE Spectrum*, *50*(3), 11-12.

Saha, B., & Goebel, K. (2009, September). Modeling Li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the annual conference of the prognostics and health management society 2009.*

Sankararaman, S., Daigle, M., Saxena, A., & Goebel, K. (2013, March). Analytical algorithms to quantify the uncertainty in remaining useful life prediction. In *Proceedings of the 2013 IEEE aerospace conference.*

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*, *1*(1).

Biographies

Matthew Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems. Dr. Daigle is a member of the Prognostics and Health Management Society and the IEEE.

Chetan S. Kulkarni received the B.E. (Bachelor of Engineering) degree in Electronics and Electrical Engineering from University of Pune, India in 2002 and the M.S. and Ph.D. degrees in Electrical Engineering from Vanderbilt University, Nashville, TN, in 2009 and 2013, respectively. In 2002 he joined Honeywell Automation India Limited (HAIL) as a Project Engineer. From May 2006 to August 2007 he was a Research Fellow at the Indian Institute of Technology (IIT) Bombay with the Department of Electrical Engineering. From Aug 2007 to Dec 2012, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. Since Jan 2013 he has been a Research Engineer II with SGT Inc. at the Prognostics Center of Excellence, NASA Ames Research Center. His current research interests include physics-based modeling, model-based diagnosis and prognosis focused towards electrical and electronic devices and systems. Dr. Kulkarni is a member of the Prognostics and Health Management (PHM) Society, AIAA and the IEEE.

Advanced Methods for Determining Prediction Uncertainty in Model-Based Prognostics with Application to Planetary Rovers

Matthew Daigle[1] and Shankar Sankararaman[2]

[1] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
matthew.j.daigle@nasa.gov

[2] *SGT, Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA*
shankar.sankararaman@nasa.gov

Abstract

Prognostics is centered on predicting the time of and time until adverse events in components, subsystems, and systems. It typically involves both a state estimation phase, in which the current health state of a system is identified, and a prediction phase, in which the state is projected forward in time. Since prognostics is mainly a prediction problem, prognostic approaches cannot avoid uncertainty, which arises due to several sources. Prognostics algorithms must both characterize this uncertainty and incorporate it into the predictions so that informed decisions can be made about the system. In this paper, we describe three methods to solve these problems, including Monte Carlo-, unscented transform-, and first-order reliability-based methods. Using a planetary rover as a case study, we demonstrate and compare the different methods in simulation for battery end-of-discharge prediction.

1. Introduction

Prognostics focuses on predicting the time of and time until adverse events in components, subsystems, and systems. Model-based methods consist of an estimation phase, in which the current health state of a system is identified, followed by a prediction phase, in which the state is simulated until the adverse event. Because prognostics is mainly a prediction problem, uncertainty, which manifests due to many sources, cannot be avoided. This uncertainty will determine how the system evolves in time, i.e., the system evolution is a random process. To approach this problem in a systematic way, there are two problems to solve: (*i*) characterizing the sources of uncertainty, and (*ii*) quantifying the combined effect of the different sources of uncertainty within the prediction. With proper estimation of the prediction uncertainty, predictions can then be used to make informed decisions about the system.

In many applications, the largest source of uncertainty is that associated with the future inputs, and so most work in uncertainty quantification for prognostics has focused in that area. In the context of fatigue damage prognosis, different types of methods (Ling et al., 2011) such as rainflow counting techniques, auto-regressive moving-average models, and Markov processes have been used for constructing future loading trajectories. In (Sankararaman, Ling, Shantz, & Mahadevan, 2011), the authors construct future input trajectories as sequential blocks of constant-amplitude loading, and such trajectories are sampled in the prediction algorithm. In (Saha, Quach, & Goebel, 2012) the authors characterize the future inputs by determining statistics of the battery loading for typical unmanned aerial vehicle maneuvers based on past flight data, and construct future input trajectories as constrained sequences of flight maneuvers. Constant loading is assumed in (Luo, Pattipati, Qiao, & Chigusa, 2008) for a vehicle suspension system, and predictions are made for a weighted set of three different loading values.

Once each source of uncertainty has been characterized, it must be accounted for within the prediction, and thereby their combined effect on the overall prediction must be quantified. In previous work (Daigle, Saxena, & Goebel, 2012), we investigated methods for accouting for future input uncertainty in the predictions and introduced the unscented transform (UT) method for efficiently estimating the future input uncertainty, however, methods for future input characterization were not discussed and only constant-amplitude loading was considered. In other work (Sankararaman, Daigle, Saxena, & Goebel, 2013; Sankararaman & Goebel, 2013), we investigated the use of analytical methods, namely, first-order reliability (FORM) based methods for propagating the future input uncertainty, however it also was limited to constant-amplitude loading, and the methods were applied only for offline analysis and not within online prognostic algorithms.

Matthew Daigle et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

In this paper, we adopt a model-based prognostics approach (Orchard & Vachtsevanos, 2009; Daigle & Goebel, 2013; Luo et al., 2008). We discuss the uncertainty representation and quantification problem, and develop a novel generalized framework using the notion of surrogate variables, allowing the representation of state uncertainty, parameter uncertainty, future input uncertainty, and process noise in a common framework that allows consideration of both constant- and variable-amplitude loading situations. We explore three methods for prediction with uncertainty quantification, namely, Monte Carlo sampling, UT sampling, and inverse FORM. Using a rover battery system as a case study, we describe two methods for future input uncertainty characterization, implement the prediction methods, and compare their performance for battery end-of-discharge prediction in simulated constant- and variable-amplitude loading scenarios.

The paper is organized as follows. In Section 2, we define the prognostics problem and develop the uncertainty representation framework. In Section 3, we describe methods for handling uncertainty within the prediction process. In Section 4, we introduce the rover case study and present several examples in simulation demonstrating the methods and comparing their performance. Section 5 concludes the paper.

2. Model-based Prognostics

We first formulate the prognostics problem, and develop the uncertainty representation framework. We then provide an architecture for model-based prognostics.

2.1. Problem Formulation

We assume the system model may be generally defined as

$$\mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{v}(k)), \quad (1)$$

$$\mathbf{y}(k) = \mathbf{h}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{n}(k)), \quad (2)$$

where k is the discrete time variable, $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\theta}(k) \in \mathbb{R}^{n_\theta}$ is the unknown parameter vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(k) \in \mathbb{R}^{n_v}$ is the process noise vector, $\mathbf{f}$ is the state equation, $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ is the output vector, $\mathbf{n}(k) \in \mathbb{R}^{n_n}$ is the measurement noise vector, and $\mathbf{h}$ is the output equation.[1] The unknown parameter vector $\boldsymbol{\theta}(k)$ is used to capture explicit model parameters whose values are unknown and time-varying stochastically.

In prognostics, we are interested in predicting the occurrence of some (desirable or undesirable) event E that is defined with respect to the states, parameters, and inputs of the system. We define the event as the earliest instant that some event threshold $T_E : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_u} \to \mathbb{B}$, where $\mathbb{B} \triangleq \{0, 1\}$ changes from the value 0 to 1. That is, the time of the event k_E at some time of prediction k_P is defined as

$$k_E(k_P) \triangleq \inf\{k \in \mathbb{N} \colon k \geq k_P \wedge T_E(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k)) = 1\}. \quad (3)$$

The time remaining until that event, Δk_E, is defined as

$$\Delta k_E(k_P) \triangleq k_E(k_P) - k_P. \quad (4)$$

In the context of systems health management, the event E corresponds to some undesirable event such as the failure of a system, some process variable out-of-range, or a similar type of situation. T_E is defined via a set of performance constraints that define what the acceptable states of the system are, based on $\mathbf{x}(k)$, $\boldsymbol{\theta}(k)$, and $\mathbf{u}(k)$ (Daigle & Goebel, 2013). In this case, k_E is then conventionally called end of life, while Δk_E is conventionally called remaining useful life.

The system evolution is a random process due to the process noise $\mathbf{v}(k)$ and nondeterministic inputs $\mathbf{u}(k)$. Since the system evolution is a random process, at time k_P, the system takes some trajectory out of many possible trajectories, therefore, k_E and Δk_E are random variables. So, the prognostics problem is to predict the *probability distribution* of k_E (Daigle, Saxena, & Goebel, 2012; Sankararaman et al., 2013).

Problem (Prognostics)**.** The prognostics problem is to, at prediction time k_P, compute $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$ and/or $p(\Delta k_E(k_P)|\mathbf{y}(k_0{:}k_P))$.

In practice, obtaining this distribution is difficult because the state at k_P, system model, process noise distribution, and future input distribution may not be known precisely.

2.2. Representing Uncertainty

In order to predict k_E, four sources of uncertainty must be dealt with: (*i*) the initial state at time k_P, $\mathbf{x}(k_p)$; (*ii*) the parameter values $\boldsymbol{\theta}(k)$ for all $k \geq k_P$, denoted as $\boldsymbol{\Theta}_{k_P}$ (the subscript k_P indicates the start time of the trajectory); (*iii*) the inputs $\mathbf{u}(k)$ for all $k \geq k_P$, denoted as $\mathbf{U}_{k_P}$; and (*iv*) the process noise $\mathbf{v}(k)$ for all $k \geq k_P$, denoted as $\mathbf{V}_{k_P}$. In order to make a prediction that accounts for this uncertainty, we require the probability distributions $p(\mathbf{x})$, $p(\boldsymbol{\Theta}_{k_P})$, $p(\mathbf{U}_{k_P})$, and $p(\mathbf{V}_{k_P})$.

For describing the probability distribution of a generic trajectory $\mathbf{A}_k$, we introduce a set of *surrogate* random variables $\boldsymbol{\lambda}_a = [\lambda_a^1 \lambda_a^2 \ldots]$. We describe a trajectory using $\boldsymbol{\lambda}_a$ and instead define $p(\boldsymbol{\lambda}_a)$, which in turn defines $p(\mathbf{A}_k)$. These surrogate variables can be used to describe trajectories in a variety of ways. For example, we can associate $\mathbf{A}_k(k)$ with λ_a^1, $\mathbf{A}_k(k+1)$ with λ_a^2, etc. Or, we can describe a trajectory with a parameterized function, where the parameters are the random variables, e.g., $\mathbf{A}_k(k) = \lambda_a^1 + \lambda_a^2 \sin k$. The use of the surrogate variables provides flexibility to the user in defining the distribution of trajectories. So, for the parameter, input,

[1] Here, we use bold typeface to denote vectors, and use n_a to denote the length of a vector $\mathbf{a}$.

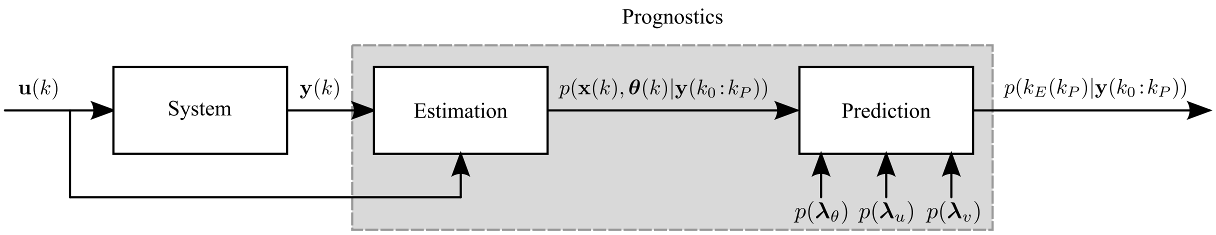

Prognostics
$\mathbf{u}(k)$
System
$\mathbf{y}(k)$
Estimation
$p(\mathbf{x}(k),\boldsymbol{\theta}(k)|\mathbf{y}(k_0:k_P))$
Prediction
$p(k_E(k_P)|\mathbf{y}(k_0:k_P))$
$p(\boldsymbol{\lambda}_\theta)$ $p(\boldsymbol{\lambda}_u)$ $p(\boldsymbol{\lambda}_v)$

Figure 1. Prognostics architecture.

and process noise trajectories we introduce the surrogate variables $\boldsymbol{\lambda}_\theta$, $\boldsymbol{\lambda}_u$, and $\boldsymbol{\lambda}_v$.

In the model-based prognostics paradigm, the probability distribution for the initial state at the time of prediction k_P, $p(\mathbf{x}(k_P), \boldsymbol{\theta}(k_P))$, is specified by an estimator, such as the Kalman filter, unscented Kalman filter (UKF) (Julier & Uhlmann, 1997), or particle filter (Arulampalam, Maskell, Gordon, & Clapp, 2002). The distribution may be represented by a set of statistical moments like a mean vector and covariance matrix (as in the Kalman filter), or a set of weighted samples (as in the UKF and particle filter). This defines only the parameter vector at k_P and not for future time steps. For the future values of $\boldsymbol{\theta}$, these are drawn from a distribution specified by $\boldsymbol{\lambda}_\theta$. In the simplest case, λ_θ^1 can define $\boldsymbol{\theta}(k_P)$, λ_θ^2 can define $\boldsymbol{\theta}(k_P + 1)$, etc, where the distribution of each λ_θ^i is the same and specified by $p(\boldsymbol{\theta}(k_p))$ as determined by the estimator. Alternatively, it can be assumed that $\boldsymbol{\theta}(k)$ is constant, in which case only one surrogate variable is required, with the distribution specified by the estimator.

For process noise, we define $p(\mathbf{V}_k)$ using $\boldsymbol{\lambda}_v$. It is often assumed in practice that, at each time instant, there is a single probability distribution from which the process noise values are drawn. The distribution is defined a priori based on the understanding of the system and its model. In the simplest case, there is a random variable for every time step, i.e, λ_v^1 defines $\mathbf{v}(k_P)$, λ_v^2 defines $\mathbf{v}(k_P + 1)$, etc. Since the number of random variables depends on the number of time steps, such a treatment potentially leads to dimensionality issues. Sankararaman and Goebel (Sankararaman & Goebel, 2013) have demonstrated that it is possible to approximate the effect of this process noise using an equivalent time-invariant process noise, i.e., a single random variable that defines a constant value of process noise for all k. In this case, $\boldsymbol{\lambda}_v$ would contain only that single random variable, whose probability distribution will be constructed by matching the likelihood of occurrence of the original time-varying process noise.

For the future input trajectories, the distribution depends on the particular system being prognosed and the environment it is operating within. As with the other trajectories, we describe $p(\mathbf{U}_{k_P})$ using $\boldsymbol{\lambda}_u$. Often, there is some knowledge about what the future input will be and only a few random variables are needed in $\boldsymbol{\lambda}_u$. For example, in a constant-loading scenario, the inputs can be defined with $\mathbf{u}(k) = \lambda_u^1$ for $k \geq k_P$. Any arbitrary function parameterized by a set of random variables may be used to define $\mathbf{u}(k)$, e.g., $\mathbf{u}(k) = \lambda_u^1 \cdot sin(k)$, or $\mathbf{u}(k) = \lambda_u^1 + \lambda_u^2 \cdot k$. The variables in $\boldsymbol{\lambda}_\theta$ may or may not be independent.

To predict k_E at time k_P, we require the initial state, $\mathbf{x}(k_P)$; the parameter trajectory up to k_E, $\boldsymbol{\Theta}_{k_P} = [\boldsymbol{\theta}(k_P), \ldots, \boldsymbol{\theta}(k_E)]$; the process noise trajectory up to k_E, $\mathbf{V}_{k_P} = [\mathbf{v}(k_P), \ldots, \mathbf{v}(k_E)]$; and the input trajectory up to k_E, $\mathbf{U}_{k_P} = [\mathbf{u}(k_P), \ldots, \mathbf{u}(k_E)]$. With this information, k_E can be computed simply by simulating the state forward in time, using the state equation $\mathbf{f}$, until T_E evaluates to 1, at which point the current time is k_E. Because we have only probability distributions, we need to propagate the uncertainty through this procedure in order to obtain the distribution for k_E (Sankararaman et al., 2013; Sankararaman & Goebel, 2013). Methods to do such uncertainty propagation will be described in Section 3.

2.3. Prognostics Architecture

We adopt a model-based prognostics architecture (Daigle & Goebel, 2013), in which there are two sequential problems, (*i*) the *estimation* problem, which requires determining a joint state-parameter estimate $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k_P))$ based on the history of observations up to time k, $\mathbf{y}(k_0{:}k_P)$, and (*ii*) the *prediction* problem, which determines at k_P, using $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k_P))$, $p(\boldsymbol{\lambda}_\theta)$, $p(\boldsymbol{\lambda}_u)$, and $p(\boldsymbol{\lambda}_v)$, a probability distribution $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$. The distribution for Δk_E can be trivially computed from $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$ by subtracting k_P from $k_E(k_P)$.

The prognostics architecture is shown in Fig. 1. In discrete time k, the system is provided with inputs $\mathbf{u}_k$ and provides measured outputs $\mathbf{y}_k$. The estimation module uses this information, along with the system model, to compute an estimate $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k))$. The prediction module uses the joint state-parameter distribution and the system model, along with the distributions for the surrogate variables, $p(\boldsymbol{\lambda}_\theta)$, $p(\boldsymbol{\lambda}_u)$, and $p(\boldsymbol{\lambda}_v)$, to compute the probability distribution $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$ at given prediction times k_P.

Algorithm 1 $k_E(k_P) \leftarrow \mathtt{P}(\mathbf{x}(k_P), \mathbf{\Theta}_{k_P}, \mathbf{U}_{k_P}, \mathbf{V}_{k_P})$

```
$k \leftarrow k_P$
$\mathbf{x}(k) \leftarrow \mathbf{x}(k_P)$
while $T_E(\mathbf{x}(k), \mathbf{\Theta}_{k_P}(k), \mathbf{U}_{k_P}(k)) = 0$ do
    $\mathbf{x}(k+1) \leftarrow \mathbf{f}(k, \mathbf{x}(k), \mathbf{\Theta}_{k_P}(k), \mathbf{U}_{k_P}(k), \mathbf{V}_{k_P}(k))$
    $k \leftarrow k + 1$
    $\mathbf{x}(k) \leftarrow \mathbf{x}(k+1)$
end while
$k_E(k_P) \leftarrow k$
```

3. Prediction

Prediction is initiated at a given time k_P using the current joint state-parameter estimate, $p(\mathbf{x}(k_P), \boldsymbol{\theta}(k_P)|\mathbf{y}(k_0{:}k_P))$. Approaches to determine this estimate are reviewed in (Daigle, Saha, & Goebel, 2012) and are not described here. The goal is to compute $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$ using the state-parameter estimates and assumptions about uncertainty regarding the future parameter, input, and process noise values.

Consider one realization of each of the uncertain quantities at prediction time k_P: the state $\mathbf{x}(k_P)$, the parameter trajectory $\mathbf{\Theta}_{k_P}$, the input trajectory $\mathbf{U}_{k_P}$, and the process noise trajectory $\mathbf{V}_{k_P}$. Then, the corresponding realization of k_E can be computed with the system model as shown in Algorithm 1. In Algorithm 1, the function $\mathtt{P}$ simulates the system model until the threshold T_E evaluates to 1.

This algorithm requires computing first realizations of the state-parameter distribution, the parameter trajectory, the input trajectory, and the process noise trajectory. As described in Section 2, the distribution for the state comes from an estimator, and the distributions for the parameter, input, and process noise trajectories are defined indirectly by a set of surrogate variables. At the higher level, we are interested in computing the distribution for k_E from the distributions for $p(\mathbf{x}(k_P), \boldsymbol{\theta}(k_P))$, $p(\boldsymbol{\lambda}_\theta)$, $p(\boldsymbol{\lambda}_u)$, and $p(\boldsymbol{\lambda}_v)$.

The function that takes these surrogate variables and computes a distribution for k_E, which we refer to as $\mathtt{G}$, is the real function we are interested in, i.e., $p(k_E) = \mathtt{G}(p(\mathbf{x}(k_P)), p(\boldsymbol{\theta}(k_P)), p(\boldsymbol{\lambda}_\theta), p(\boldsymbol{\lambda}_u), p(\boldsymbol{\lambda}_v))$. To compute $p(k_E)$, we must propagate the uncertainty through this function. That is, predicting $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$ is an uncertainty propagation problem. In the following subsections, we describe three different methods with which to solve this problem. They each compute realizations of the state, parameter trajectory, input trajectory, and process noise trajectory, and call the $\mathtt{P}$ function to obtain a realization of k_E. They differ in how they compute these realizations and how they construct $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$ from them, and, consequently, in their computational complexity.

3.1. Monte Carlo Sampling

To account for uncertainty in the prediction step, the simplest method is Monte Carlo sampling. Several realizations of the state, parameter trajectory, input trajectory, and process noise trajectory are sampled from their corresponding distributions. For each realization, k_E is computed. The resulting set of k_E values characterizes its distribution.

Algorithm 2 $\{k_E^{(i)}\}_{i=1}^N = \mathtt{MC}(p(\mathbf{x}(k_P), \boldsymbol{\theta}(k_P)|\mathbf{y}(k_0{:}k_P)), p(\boldsymbol{\lambda}_\theta), p(\boldsymbol{\lambda}_u), p(\boldsymbol{\lambda}_v), N)$

```
for $i = 1$ to $N$ do
    $(\mathbf{x}^{(i)}(k_P), \boldsymbol{\theta}^{(i)}(k_P)) \sim p(\mathbf{x}(k_P), \boldsymbol{\theta}(k_P)|\mathbf{y}(k_0{:}k_P))$
    $\boldsymbol{\lambda}_\theta^{(i)} \sim p(\boldsymbol{\lambda}_\theta)$
    $\mathbf{\Theta}_{k_P}^{(i)} \leftarrow \mathtt{construct\Theta}(\boldsymbol{\lambda}_\theta^{(i)}, \boldsymbol{\theta}^{(i)}(k_P))$
    $\boldsymbol{\lambda}_u^{(i)} \sim p(\boldsymbol{\lambda}_u)$
    $\mathbf{U}_{k_P}^{(i)} \leftarrow \mathtt{constructU}(\boldsymbol{\lambda}_u^{(i)})$
    $\boldsymbol{\lambda}_v^{(i)} \sim p(\boldsymbol{\lambda}_v)$
    $\mathbf{V}_{k_P}^{(i)} \leftarrow \mathtt{constructV}(\boldsymbol{\lambda}_v^{(i)})$
    $k_E^{(i)} \leftarrow \mathtt{P}(\mathbf{x}^{(i)}(k_P), \mathbf{\Theta}_{k_P}^{(i)}, \mathbf{U}_{k_P}^{(i)}, \mathbf{V}_{k_P}^{(i)})$
end for
```

Algorithm 2 shows the Monte Carlo prediction algorithm. The algorithm is given as input the joint state-parameter distribution, and the distributions of the $\boldsymbol{\lambda}_\theta$, $\boldsymbol{\lambda}_u$, and $\boldsymbol{\lambda}_v$ variables, along with the number of samples to take, N. For N times, the algorithm samples from the distributions, constructs the parameter, input, and process noise trajectories, and calls the $\mathtt{P}$ function to compute k_E. To construct the trajectories from the $\boldsymbol{\lambda}$ variables, the $\mathtt{construct\Theta}$ $\mathtt{constructU}$ and $\mathtt{constructV}$ functions must be provided by the user, as they depend on the chosen surrogate variables and how they are to be interpreted. Note that the $\mathtt{construct\Theta}$ function requires an additional input, $\boldsymbol{\theta}^{(i)}(k_P)$, which is a sample from the estimator-computed joint parameter estimate at time k_P.

In any prediction algorithm, computational complexity is driven by two factors: the number of evaluations of $\mathtt{P}$, and the length of time each sample takes to simulate to k_E (Daigle & Goebel, 2010). Assuming a fair comparison for the second factor, we can compare algorithms mainly by the first factor. In the case of Monte Carlo sampling, the number of samples N is arbitrary and determines the efficiency. In most cases, a very large value of N is required in order to reproduce accurately the important characteristics of the k_E distribution.

3.2. Unscented Transform Sampling

A more complex method that can improve the efficiency of prediction over the Monte Carlo method is to use the unscented transform (UT) to sample from the distributions (Daigle, Saxena, & Goebel, 2012). We present here an extended and generalized version of the method originally presented in (Daigle, Saxena, & Goebel, 2012) in order to accommodate the $\boldsymbol{\lambda}$ variable formulation.

The UT takes a random variable $\mathbf{a} \in \mathbb{R}^{n_a}$, with mean $\bar{\mathbf{a}}$ and covariance $\mathbf{P}_{aa}$, that is related to a second random variable

$\mathbf{b} \in \mathbb{R}^{n_b}$ by some function $\mathbf{b} = \mathbf{g}(\mathbf{a})$, and computes the mean $\bar{\mathbf{b}}$ and covariance $\mathbf{P}_{bb}$ with high accuracy using a minimal set of deterministically selected weighted samples, called *sigma points* (Julier & Uhlmann, 1997). The number of sigma points is linear in the dimension of the random variable, and so the statistics (mean and covariance) of the transformed random variable can be computed more efficiently than Monte Carlo sampling.

Here, $\boldsymbol{\mathcal{A}}^i$ denotes the ith sigma point from $\mathbf{a}$ and w^i denotes its weight. The sigma points are always chosen such that the mean and covariance match those of the original distribution, $\bar{\mathbf{a}}$ and $\mathbf{P}_{aa}$. Each sigma point is passed through $\mathbf{g}$ to obtain new sigma points $\boldsymbol{\mathcal{B}}$, i.e.,

$$\boldsymbol{\mathcal{B}}^i = \mathbf{g}(\boldsymbol{\mathcal{A}}^i)$$

with mean and covariance calculated as

$$\bar{\mathbf{b}} = \sum_i w^i \boldsymbol{\mathcal{B}}^i$$
$$\mathbf{P}_{bb} = \sum_i w^i (\boldsymbol{\mathcal{B}}^i - \bar{\mathbf{b}})(\boldsymbol{\mathcal{B}}^i - \bar{\mathbf{b}})^T.$$

In this paper, we use the symmetric unscented transform (SUT), in which $2n_a + 1$ sigma points are symmetrically selected about the mean according to (Julier & Uhlmann, 2004):

$$w^i = \begin{cases} \dfrac{\kappa}{(n_a+\kappa)}, & i = 0 \\ \dfrac{1}{2(n_a+\kappa)}, & i = 1, \ldots, 2n_a \end{cases}$$
$$\boldsymbol{\mathcal{A}}^i = \begin{cases} \bar{\mathbf{a}}, & i = 0 \\ \bar{\mathbf{a}} + \left(\sqrt{(n_a+\kappa)\mathbf{P}_{aa}}\right)^i, & i = 1, \ldots, n_a \\ \bar{\mathbf{a}} - \left(\sqrt{(n_a+\kappa)\mathbf{P}_{aa}}\right)^i, & i = n_a+1, \ldots, 2n_a, \end{cases}$$

where $\left(\sqrt{(n_a+\kappa)\mathbf{P}_{aa}}\right)^i$ refers to the ith column of the matrix square root of $(n_a + \kappa)\mathbf{P}_{aa}$, and κ is a free parameter that can be used to tune higher order moments of the distribution. When $\mathbf{a}$ is assumed to be Gaussian, (Julier & Uhlmann, 1997) recommends setting $\kappa = 3 - n_a$. Note that with the UT, weights may be negative, and are not to be directly interpreted as probabilities.

For prediction, the G function serves as $\mathbf{g}$ in the above formulation, where $\mathbf{a}$ corresponds to the joint distribution of the state and $\boldsymbol{\lambda}$ variables, and $\mathbf{b}$ corresponds to k_E. The prediction algorithm in this case is shown as Algorithm 3. The algorithm first uses the symmetric unscented transform to compute sigma points for the given probability distributions (treated together as a joint distribution), where each sigma point consists of a sample for the state-parameter vector and the $\boldsymbol{\lambda}$ variables. For each sigma point, the parameter, input, and process noise trajectories are constructed and the P function is called to compute the corresponding k_E. The mean and variance for k_E can be computed from its sigma points.

Algorithm 3 $\{k_E^{(i)}, w^{(i)}\}_{i=1}^N = \texttt{UT}(p(\mathbf{x}(k_P),\ \boldsymbol{\theta}(k_P)|\mathbf{y}(k_0{:}k_P)), p(\boldsymbol{\lambda}_\theta), p(\boldsymbol{\lambda}_u), p(\boldsymbol{\lambda}_v))$

```
$N \leftarrow 2(n_x + n_\theta + n_{\lambda_\theta} + n_{\lambda_u} + n_{\lambda_v}) + 1$
$\{\mathbf{x}^{(i)}(k_P), \boldsymbol{\theta}^{(i)}(k_P), \boldsymbol{\lambda}_\theta, \boldsymbol{\lambda}_u, \boldsymbol{\lambda}_v, w^{(i)}\}_{i=1}^N \leftarrow$
    $\texttt{SUT}((p(\mathbf{x}(k_P), \boldsymbol{\theta}(k_P)|\mathbf{y}(k_0{:}k_P)), p(\boldsymbol{\lambda}_\theta), p(\boldsymbol{\lambda}_u), p(\boldsymbol{\lambda}_v)))$
for $i = 1$ to $N$ do
    $\boldsymbol{\Theta}_{k_P}^{(i)} \leftarrow \texttt{construct}\Theta(\boldsymbol{\lambda}_\theta^{(i)}, \boldsymbol{\theta}^{(i)}(k_P))$
    $\mathbf{U}_{k_P}^{(i)} \leftarrow \texttt{constructU}(\boldsymbol{\lambda}_u^{(i)})$
    $\mathbf{V}_{k_P}^{(i)} \leftarrow \texttt{constructV}(\boldsymbol{\lambda}_v^{(i)})$
    $k_E^{(i)} \leftarrow \texttt{P}(\mathbf{x}^{(i)}(k_P), \boldsymbol{\Theta}_{k_P}^{(i)}, \mathbf{U}_{k_P}^{(i)}, \mathbf{V}_{k_P}^{(i)})$
end for
```

This prediction method will often require a smaller number of samples than with Monte Carlo sampling, since the number of sigma points grows only linearly with the problem dimension. This is partly due to the fact that the UT method provides only mean and covariance information, whereas additional higher-order moments can be computed with the Monte Carlo method. Extended versions of the UT are also available that compute higher-order statistical moments (Julier, 1998).

3.3. Inverse First-Order Reliability Method

The Monte Carlo and UT approaches are sampling-based techniques to predict the uncertainty in the k_E. Here we briefly explain an *optimization-based* method, the inverse first-order reliability method, for this purpose. The First-order Reliability Method (FORM) and the Inverse First-Order Reliability Method (inverse FORM) were originally developed by structural engineers to evaluate the probability of failure of a given structure (Haldar & Mahadevan, 2000). In an earlier publication (Sankararaman et al., 2013), we have extended these two approaches for uncertainty quantification in the context of remaining useful life estimation, i.e., propagate the uncertainty in present estimates of states and parameters, future loading, future process noise, and future parameter values through P (defined earlier in Algorithm 1) to calculate the uncertainty in k_E. In the present paper, we use the inverse FORM methodology to calculate the entire probability distribution of k_E in terms of the cumulative distribution function. Calculating the cumulative distribution function (CDF) is equivalent to calculating the probability density function $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$, since the density function can easily be obtained by differentiating the cumulative distribution function.

For a generic random variable Z, the cumulative distribution function is a mapping from a realization z of the random variable to the set $[0, 1]$, and is denoted by $F_Z(z)$. If $F_Z(z) = \eta$, then the probability that the random variable Z is less than a given value z is equal to η. In the context of prognostics, the goal is to compute the uncertainty in k_E. Typically, the

Algorithm 4 $k_E(k_P) \leftarrow \mathtt{P}_{\boldsymbol{\lambda}}(\omega)$

```
$[\mathbf{x}(k_P), \boldsymbol{\theta}(k_P), \boldsymbol{\lambda}_\theta, \boldsymbol{\lambda}_u, \boldsymbol{\lambda}_v] \leftarrow \omega$
$\boldsymbol{\Theta}_{k_P} \leftarrow \mathtt{construct\Theta}(\boldsymbol{\lambda}_\theta, \boldsymbol{\theta}(k_P))$
$\mathbf{U}_{k_P} \leftarrow \mathtt{constructU}(\boldsymbol{\lambda}_u)$
$\mathbf{V}_{k_P} \leftarrow \mathtt{constructV}(\boldsymbol{\lambda}_v)$
$k_E(k_P) \leftarrow \mathtt{P}(\mathbf{x}(k_P), \boldsymbol{\Theta}_{k_P}, \mathbf{U}_{k_P}, \mathbf{V}_{k_P})$
```

quantities $\mathbf{x}(k_P)$, $\boldsymbol{\theta}(k_P)$, $\boldsymbol{\lambda}_\theta$, $\boldsymbol{\lambda}_u$, and $\boldsymbol{\lambda}_v$ are vectors, and now, consider a new vector which is the concatenation of all these vectors as ω = [$\mathbf{x}(k_P)$, $\boldsymbol{\theta}(k_P)$, $\boldsymbol{\lambda}_\theta$, $\boldsymbol{\lambda}_u$, $\boldsymbol{\lambda}_v$]. Based on the probability distributions of $\mathbf{x}(k_P)$, $\boldsymbol{\theta}(k_P)$ $\boldsymbol{\lambda}_\theta$, $\boldsymbol{\lambda}_u$, $\boldsymbol{\lambda}_v$, the joint probability density of ω, denoted as $f_{\boldsymbol{\Omega}}(\omega)$, can be easily calculated. Note that ω is a realization of the random variable that is denoted by $\boldsymbol{\Omega}$.

In order to implement the inverse FORM, it is necessary to construct a function whose inputs are ω = [$\mathbf{x}(k_P)$, $\boldsymbol{\theta}(k_P)$, $\boldsymbol{\lambda}_\theta$, $\boldsymbol{\lambda}_u$, $\boldsymbol{\lambda}_v$] and the output is k_E. This function similar to $\mathtt{P}$, with one difference; while $\mathtt{P}$ takes realizations of entire trajectories, i.e., $\boldsymbol{\Theta}_{k_P}$,$\mathbf{U}_{k_P}$, and $\mathbf{V}_{k_P}$ as input arguments, the new function needs to consider realizations of the corresponding surrogate variables as input arguments. This new function, denoted by $\mathtt{P}_\lambda$, is indicated in Algorithm 4.

The inverse FORM approach is now explained using the function $k_E = \mathtt{P}_{\boldsymbol{\lambda}}(\omega)$. The reason for such vectorized representation using ω is not only to facilitate easy explanation of the FORM and inverse FORM algorithms but also demonstrate that these algorithms do not differentiate amongst state estimate values, parameter values, future loading trajectories, and process noise trajectories but treat them similarly.

Let the CDF of k_E be denoted as $F_{K_e}(k_E) = \eta$. Using $k_E = \mathtt{P}_{\boldsymbol{\lambda}}(\omega)$, the FORM approach can be used to calculate the value of η corresponding to a given value of k_E. Conversely, the inverse FORM approach can be used to calculate the value of k_E corresponding to a given value of η. By repeating the FORM procedure for multiple values of k_E, or by repeating the inverse FORM procedure for multiple values of η, the entire CDF $F_{K_e}(k_E)$ can be calculated. In a practical scenario, it is not reasonable to know what values of k_E need to be selected to implement the FORM procedure, since the goal is actually to compute the uncertainty in k_E. However, it is easier to select values of η (say, 0.1, 0.2, 0.3 and so on until 0.9) which span the entire probability range and implement the inverse FORM procedure for each of these η values. Therefore, we use the inverse FORM approach to quantify the uncertainty in k_E. The authors have explained the inverse FORM algorithm in detail in previous work (Sankararaman et al., 2013); in this section, the overall approach is briefly summarized and the algorithm is provided.

Both FORM and inverse FORM approaches linearize the curve represented by the equation $k_E = \mathtt{P}_{\boldsymbol{\lambda}}(\omega)$ and transform all the variables in $\boldsymbol{\Omega}$ to standard normal variables (Gaussian distribution with zero mean and unit variance) using well-defined, popular transformation functions. Thus, k_E can be expressed as a linear sum of Gaussian variables, and therefore the probability distribution of k_E can be computed easily. The key point in these algorithms is that the point of linearization is chosen to be the Most Probable Point (MPP), i.e., the point of maximum likelihood value. For example, in a Gaussian distribution, the MPP is at the mean. Since each uncertain quantity in $\boldsymbol{\Omega}$ may have its own distribution, the MPP is computed in the standard normal space, where the origin has the highest likelihood of occurrence. However, the origin may not satisfy the equation $k_E = \mathtt{P}_{\boldsymbol{\lambda}}(\omega)$, and the point of linerization needs to lie on the curve represented by the equation $k_E = \mathtt{P}_{\boldsymbol{\lambda}}(\omega)$. Therefore, the problem reduces to estimating the minimum distance (measured from the origin, in the standard normal space) point on the curve represented by the equation $k_E = \mathtt{P}_{\boldsymbol{\lambda}}(\omega)$. This is posed as a constrained minimization problem, and solved using a well-known gradient-based optimization technique, as described in Algorithm 5. Once the minimum distance (denoted by β) has been evaluated, then it can be proved that $F_{K_E}(k_E) = \Phi(-\beta)$, where $\Phi(.)$ represents the standard normal distribution function. Algorithm 5 explains the numerical implementation of the inverse FORM approach.

Algorithm 5 $\{k_E^{(i)}, \eta^{(i)}\}_{i=1}^N \leftarrow \mathtt{InverseFORM}(f_{\boldsymbol{\Omega}}(\omega), \mathtt{P}_{\boldsymbol{\lambda}})$

```
$N \leftarrow$ Number of $\eta$ values to consider.
$M \leftarrow$ Number of elements in $\omega$
{Example: $N = 9$, $\eta^{(i)} = 0.1 \times i$, $i = 1$ to $N$. }
for $i = 1$ to $N$ (For every $\eta$ value) do
    $\beta^{(i)} \leftarrow -\Phi^{-1}(\eta^{(i)})$
    $\omega_0 \leftarrow$ Select initial guess for optimization
    Convergence $= 0$
    $j = 0$ {Iteration number}
    while Convergence $\leftarrow 0$ do
        $\phi_j \leftarrow T(\omega_j)$ {Transformation to Std. Normal Space}
        $\phi_j \leftarrow [\phi_{jk}; k = 1$ to $M]$
        $\alpha_j \leftarrow [\alpha_{jk}; k = 1$ to $M]$ where $\alpha_{jk} = \frac{\partial \mathtt{P}_{\boldsymbol{\lambda}}}{\partial \phi_{jk}}$
        $\phi_{j+1} \leftarrow -\frac{\alpha_j}{|\alpha_j|} \times \beta^{(i)}$
        $\omega_{j+1} \leftarrow T^{-1}(\phi_{j+1})$ {Transformation to Original Space}
        if $\omega_{j+1} \approx \omega_j$ then
            Convergence $\leftarrow 1$
        end if
        $j \leftarrow j + 1$
    end while
    $k_E^{(i)} \leftarrow \mathtt{P}_{\boldsymbol{\lambda}}(\omega_j)$
end for
```

In the above algorithm, note that the user needs to specify functions T and T^{-1} for transforming original space to standard normal space and from standard normal space to original space respectively. There are several types of transformation available in the literature (Haldar & Mahadevan, 2000) and any valid transformation may be used. Further, note that the gradient α needs to be calculated in the standard normal space. This depends on (*i*) the gradient in the original space; and (*ii*) the chosen transformation T.

Using the algorithm, the values of k_E corresponding to the chosen values of η are first obtained, and then an interpolation technique can be used to obtain the entire CDF. Also, if the goal is to quickly obtain bounds on k_E, then we may consider two η values in either tail of the probability distribution (say, for example, η values of 0.1 and 0.9), and the corresponding probability bounds of k_E can be obtained.

4. Case Study

As a case study, we perform battery prognostics on a planetary rover and present simulation-based results. The rover and its simulation are described in detail in (Balaban et al., 2011). The rover battery system consists of two parallel sets of 12 batteries in series to provide around 48 V. We are interested in predicting end-of-discharge (EOD), which is defined as the point when the voltage of a single battery drops below 2.5 V.

We consider two different scenarios for the rover. In both scenarios, the rover is provided a sequence of waypoints that must be visited. The rover travels through the waypoints in order until the batteries discharge. In the first scenario, the desired forward speed of the rover is the same when moving to each waypoint. In the second scenario, the desired forward speed is different depending on which waypoint is being approached. Since the needed power draw from the batteries depends on speed, the first scenario resembles a constant-amplitude loading situation, and the second scenario resembles a variable-amplitide loading scenario.

The battery prognostics architecture works as follows. The rover provides voltage measurements on all batteries, and the current going into the batteries. Because there are two sets of batteries in parallel, each battery sees only half the measured current. The measured current and voltage are fed into the battery prognoser. The battery prognoser uses an unscented Kalman filter (UKF) to perform state estimation (see (Julier & Uhlmann, 1997, 2004; Daigle, Saha, & Goebel, 2012) for details on the UKF). The state estimate is then fed into the predictor, which makes EOD predictions every 100 seconds.

In the remainder of the section, we describe the details of the underlying battery model used by the rover simulation and the battery prognoser, and provide simulation-based experimental results for different scenarios and comparing the different methods presented in Section 3.

4.1. Battery Model

The battery model is extended from that presented in (Daigle, Saxena, & Goebel, 2012). The model is based on an electrical circuit equivalent as shown in Fig. 2, following similar modeling approaches to (Chen & Rincon-Mora, 2006; Ceraolo, 2000). The large capacitance C_b holds the charge q_b of the battery. The nonlinear C_b captures the open-circuit potential and concentration overpotential. The R_{sp}-C_{sp} pair captures the major nonlinear voltage drop due to surface overpotential, R_s captures the so-called Ohmic drop, and R_p models the parasitic resistance that accounts for self-discharge. This empirical battery model is sufficient to capture the major dynamics of the battery while ignoring temperature effects and additional minor processes. The governing equations for the battery model are presented in continuous time below. The implementation of the proposed methodology considers a discrete-time version with a discrete time-step of 1 s.

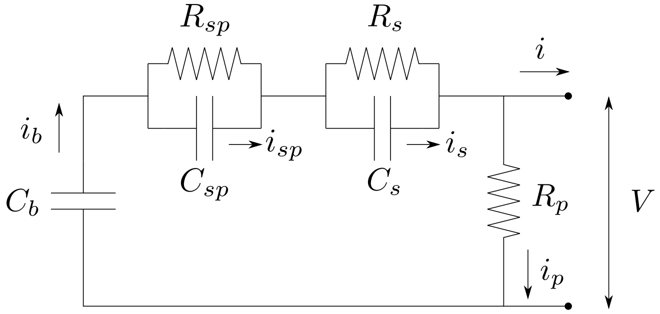


Figure 2. Battery equivalent circuit.

The state-of-charge, SOC, is computed as

$$SOC = 1 - \frac{q_{max} - q_b}{C_{max}}, \tag{5}$$

where q_b is the current charge in the battery (related to C_b), q_{max} is the maximum possible charge, and C_{max} is the maximum possible capacity. The resistance related to surface overpotential is a nonlinear function of SOC:

$$R_{sp} = R_{sp_0} + R_{sp_1} \exp\left(R_{sp_2}(1 - SOC)\right), \tag{6}$$

where R_{sp_0}, R_{sp_1}, and R_{sp_2} are empirical parameters. The resistance, and, hence, the voltage drop, increases exponentially as SOC decreases.

Voltage drops across the individual circuit elements are given by

$$V_b = \frac{q_b}{C_b}, \tag{7}$$

$$V_{sp} = \frac{q_{sp}}{C_{sp}}, \tag{8}$$

$$V_s = \frac{q_s}{C_s}, \tag{9}$$

$$V_p = V_b - V_{sp} - V_s, \tag{10}$$

where q_{sp} is the charge associated with the capacitance C_{sp}, and q_s is the charge associated with C_s. The voltage V_b is also the open-circuit voltage of the battery, which is a nonlinear function of SOC (Chen & Rincon-Mora, 2006). This is captured by expressing C_b as a third-order polynomial func-

Table 1. Battery Model Parameters

Parameter	Value
C_{b_0}	19.80 F
C_{b_1}	1745.00 F
C_{b_2}	-1.50 F
C_{b_3}	-200.20 F
R_s	0.0067 Ω
C_s	115.28 F
R_p	1×10^4 Ω
C_{sp}	316.69 F
R_{sp_0}	0.0272 Ω
R_{sp_1}	1.087×10^{-16} Ω
R_{sp_2}	34.64
q_{max}	3.11×10^4 C
C_{max}	30807 C

tion of SOC:

$$C_b = C_{b_0} + C_{b_1} SOC + C_{b_2} SOC^2 + C_{b_3} SOC^3 \qquad (11)$$

The terminal voltage of the battery is

$$V = V_b - V_{sp} - V_s. \qquad (12)$$

Currents associated with the individual circuit elements are given by

$$i_p = \frac{V_p}{R_p}, \qquad (13)$$

$$i_b = i_p + i, \qquad (14)$$

$$i_{sp} = i_b - \frac{V_{sp}}{R_{sp}}, \qquad (15)$$

$$i_s = i_b - \frac{V_s}{R_s}. \qquad (16)$$

The charges are then governed by

$$\dot{q}_b = -i_b, \qquad (17)$$

$$\dot{q}_{sp} = i_{sp}, \qquad (18)$$

$$\dot{q}_s = i_s. \qquad (19)$$

In the case of the battery, the event E we are interested in predicting is EOD. T_E is specified as $V < V_{\text{EOD}}$, where $V_{\text{EOD}} = 2.5$ V.

The parameter values of the battery model are given in Table 1. All voltages are measured in Volts, resistances are measured in Ohms, charges are measured in Coulombs, and capacitances are measured in Coulombs per Volt (or Farads). Note that C_{b_0}, C_{b_1}, C_{b_2}, and C_{b_3} are simply fitting parameters in Eq. 11 and do not have physical meaning.

For the battery model, $\mathbf{x} = [q_b \; q_{sp} \; q_s]$, $\boldsymbol{\theta} = \varnothing$ (i.e., all parameters are assumed constant and no parameters will be estimated online), and $\mathbf{y} = [V]$. We consider power P to be the input to the battery, so $i = P/V$, i.e, $\mathbf{u} = [P]$. Here, we choose power as the input, rather than current as in previous battery prognostics works, because it is simpler to describe battery load in terms of power. For the same power demands from the rover onto the battery, current will increase as battery voltage decreases; it is necessary to capture this current-voltage relationship in order to use current as input. Therefore, it is much easier to predict future power usage than to predict future current draw, and hence, power is used as input.

4.2. Future Input Characterization

In the experiments presented in this section, we will consider only uncertainty in the state and in the future inputs (methods for dealing with process noise are described in (Daigle, Saxena, & Goebel, 2012; Sankararaman & Goebel, 2013)). Therefore we need to define $p(\boldsymbol{\lambda}_u)$ and the `constructU` function. We explore three methods that differ in complexity and the amount of system knowledge used.

The future input trajectory to a battery model depends on how the rover will be used. When moving from one waypoint to the next, the rover must turn towards the next waypoint while maintaining its foward speed (that is how the locomotion controller is designed to work). For the same forward speed, turning actually requires more power than going straight, because the rover must also move against the ground torques produced while turning in addition to the opposing forces produced when moving forwards. Further, because of the turning, the actual distance traveled between two waypoints is greater than the straight-line distance between them because the rover actually takes a curved path.

To correctly account for all these details, a system-level approach is required (Daigle, Bregon, & Roychoudhury, 2012). In this case, the whole rover and its locomotion controller would be considered as the system under prognosis. Thus, the whole rover would be simulated moving through the different waypoints, and this would define very precisely (depending on model fidelity) the power drawn from the batteries as a function of time. However, such an approach is more computationally expensive than considering only a single battery model.

A simpler approach is to assume that, in the current operation of the system, the future inputs to the system will look like the past inputs. That is, we can assume that the future power requirements will be constant with the mean and variance defined by the past power requirements over some time window. If the window size is large enough, then the differences in power that arise may be represented well enough in the statistics of the past behavior. Although simple, such an approach may work well in some circumstances.

As a middle ground, we can incorporate some system-level knowledge into predicting the future power requirements without resorting to a system-level simulation. We can do this by computing the mean and variance of the power draw and distance traveled between waypoint pairs for each forward speed setting. Then, knowing the current rover location and the remaining waypoints, given a realization of each distance and power variable for the remaining waypoints and the desired forward speed when heading to each waypoint, we can compute the future power as a function of time based on the expected path the rover is going to take. This approach uses system knowledge, i.e., knowledge beyond the battery model, in order to compute useful predictions of the future inputs to the battery, and therefore makes useful EOD predictions for the battery, but without resorting to a system-level simulation.

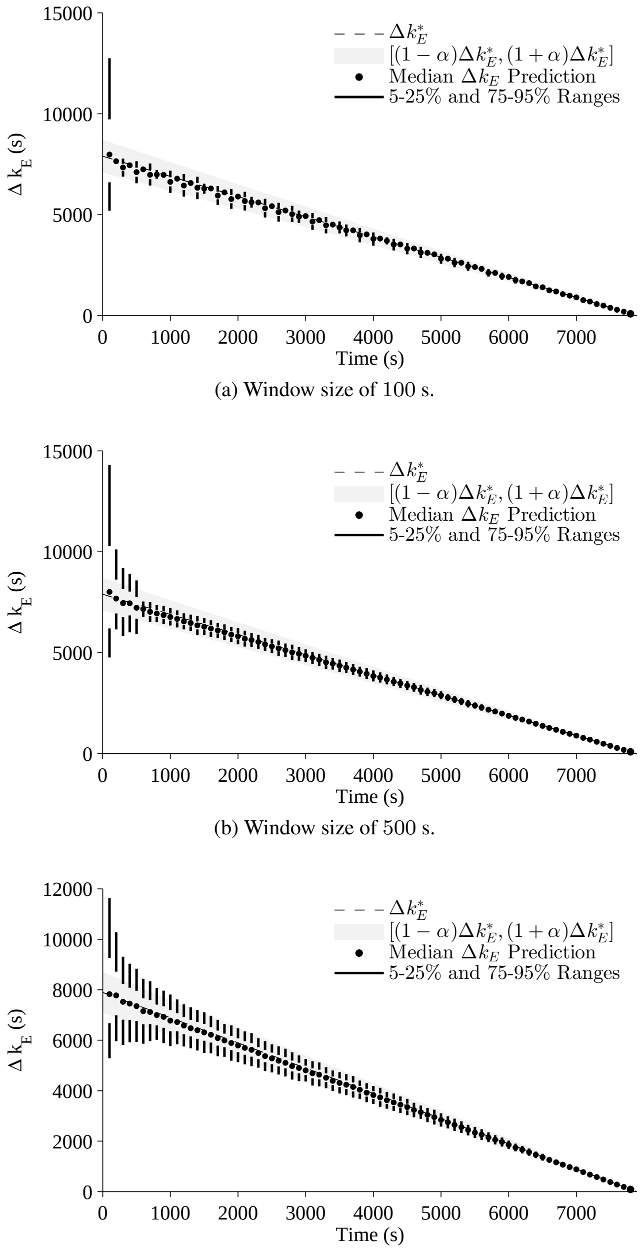
4.3. Results

In the simulation experiments considered in this section, all parameters are considered known exactly and no process noise is added. The two potential sources of uncertainty are related to the state estimate obtained by the UKF and the future input assumptions. Predictions are made every 100 s until EOD, and the accuracy and precision metrics are averaged over all these predictions. We use the relative accuracy (RA) metric as defined in (Saxena, Celaya, Saha, Saha, & Goebel, 2010) as a measure of accuracy and relative standard deviation (RSD) as a measure of spread. In the following plots, the * superscript indicates the ground truth values.

4.3.1. Constant-Loading Scenario

We consider first the scenario where the rover must move between equidistant waypoints at the same forward speed, resembling a constant-loading situation. Let us first assume that the future inputs (the battery power) are known exactly. There are 3 states in the battery model, so 7 sigma points are used by the UKF, and these are directly simulated forward to compute the k_E distribution using the sigma point weights. In this case, since uncertainty is limited only to that in the state estimate, predictions are both very accurate and precise, with RA averaging to 99.65% and RSD to 0.64%.

Now assume that the past power requirements are statistically representative of the future power requirements. Here, we consider Monte Carlo sampling with 3500 samples. We consider window sizes of 100 s, 500 s, and unlimited size. The results are shown in Fig. 3. In all three cases, the uncertainty starts initially very large, because the window is not large enough to accurately capture the statistics of the power usage. With a small window size (Fig. 3a), the statistics of the power usage averaged over the window fluctuate. The variance is larger when both turns and forward movements appear in the same window, and smaller when only forward movements are in the window. With a larger window size the variance will average to a larger value that accounts for both turns and forward movements, as seen in Figs. 3b and 3c. Since the past power usage turns out to be a good indicator of future power usage in this scenario, the results are fairly accurate, with RAs of 98.14%, 98.47%, and 97.53% for 100 s, 500 s, and unlimited window sizes, respectively. Corresponding RSDs are 3.83%, 5.57%, and 7.61%. The spread increases as window size increases since more variation is accounted for in larger windows.



(a) Window size of 100 s.

(b) Window size of 500 s.

(c) Unlimited window size.

Figure 3. Δk_E predictions using Monte Carlo sampling.

Using a window size of 500 s, the results using the UT are shown in Fig. 4. Here, the results are comparable to using Monte Carlo sampling with the same window size, with an average RA of 98.69% and RSD of 5.24%. The UT method, however, needs only 9 total samples, with there being only 3 states and one surrogate input variable to consider. This

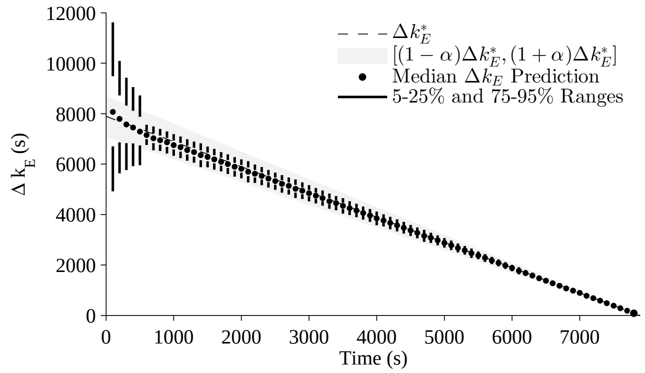


Figure 4. Δk_E predictions with a window size of 500 seconds using UT sampling.

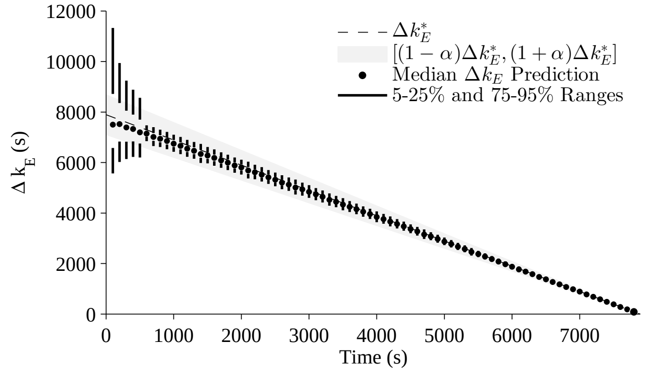


Figure 5. Δk_E predictions with a window size of 500 seconds using inverse FORM.

results in a substantial decrease in computational cost compared to the Monte Carlo approach. Fig. 5 shows the results using inverse FORM. The results are similar to using both Monte Carlo and UT sampling, and RA averages to 98.46% and RSD to 3.43%.

Using knowledge about the future waypoints to be visited, we can improve over using a window of past data to determine future inputs to the system. Fig. 6 shows the improved future input characterization method using Monte Carlo sampling with 3500 samples. The plots look the same for UT sampling and inverse FORM. The accuracy is comparable to the previous approach, with an average RA of 98.29% for Monte Carlo sampling, 98.32% for UT sampling, and 98.30% for inverse FORM. RSDs, however, are lower now since the future inputs are known with more precision than could be derived from a window of past samples. RSD averages to 1.61% for both Monte Carlo and UT sampling and 4.67% for inverse FORM.

4.3.2. Variable-Loading Scenario

We now consider the second scenario that uses the same waypoints as the previous scenario, but the rover is commanded to go different speeds depending on which waypoint is being headed towards, resembling a variable-loading situation. Assuming the future inputs are known exactly, the average RA is 99.50% and the average RSD is 0.70%. The only uncertainty is in the state estimate.

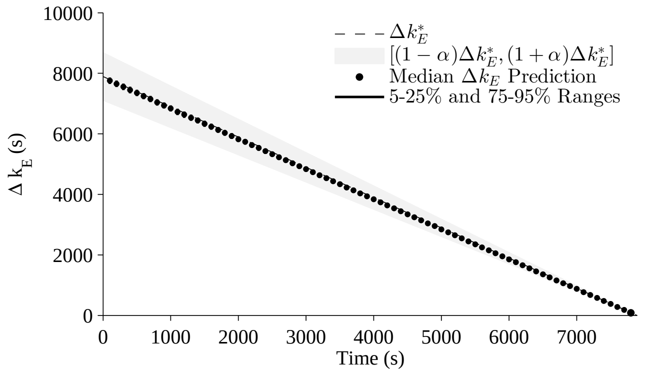
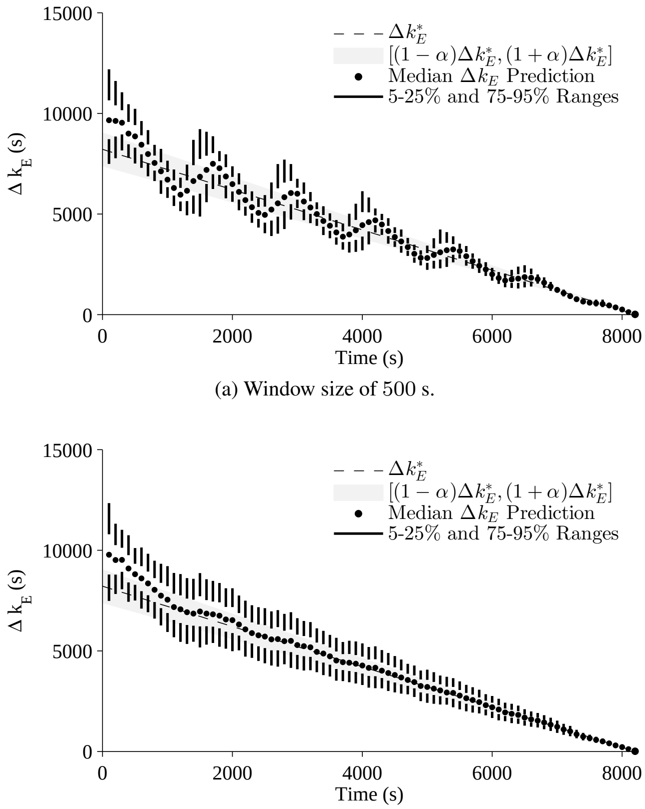


Figure 6. Δk_E predictions with improved future input characterization using Monte Carlo sampling.

Because the speed of the rover will change with each new waypoint, it is no longer correct to assume that past power requirements are representative of future power requirements. Fig. 7a shows the results when we incorrectly make this assumption for Monte Carlo sampling with 3500 samples, using a window size of 500 s. Clearly, the predictions are very inaccurate. RA averages to 90.53% and RSD to 13.74%. Using UT sampling we find similar results, with RA averaging to 90.12% and RSD to 13.54%. Using inverse FORM, RA averages to 90.59% and RSD to 9.24%. When the average speed of the rover in the future is greater than what is assumed based on the window of past samples, then Δk_E is overestimated. When the average speed is less than what is assumed, Δk_E is underestimated. Because the average speed over the window changes based on the previous waypoints within that window, the predictions fluctuate above and below ground truth. If the window size is increased, such that it accounts for all possible speed settings, then accuracy can be improved because the assumed average speed based on past samples will match the future average speed, however, spread will also increase since multiple speeds are considered in the window. Predictions with an unlimited window size are shown in Fig. 7b. The predictions initially fluctuate as the window begins to fill up, but by 2000 s the predictions have smoothed out. The spread is clearly larger than with the smaller window size, but predictions are more accurate once the window contains all potential future speeds. In this case, RA improves to 96.62% but RSD increases to 17.47%.

Predictions can be improved by using system knowledge to help characterize the future inputs. In this case, the future power as a function of time is computed based on the rover's current location, the remaining waypoints, and the desired



(a) Window size of 500 s.

(b) Unlimited window size.

Figure 7. Δk_E predictions using Monte Carlo sampling.

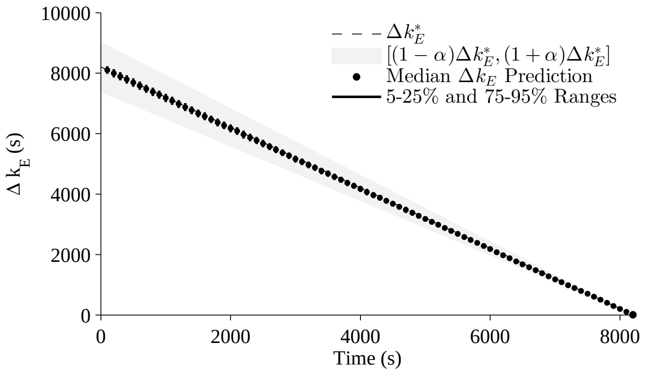


Figure 8. Δk_E predictions with improved future input characterization using Monte Carlo sampling.

speeds when heading to each waypoint, and measured statistics on average power between waypoints and average distance traveled (to account for turns). Fig. 8 shows the results using Monte Carlo sampling. The plots for UT sampling and inverse FORM look the same. All three methods are now clearly very accurate and precise. RA averages to 98.85% and RSD to 1.95% for Monte Carlo sampling, 98.82% and 1.91% for UT sampling, and 97.77% and 2.72% for inverse FORM. Unlike when using a window, here knowledge of the future waypoints and desired speeds allows accurate predictions to be made from the start of the scenario, and with very little spread. That is, the future inputs are well-known and so predictions are very close to the optimal.

In the case above, for UT sampling, there are 3 states to consider and at most 100 surrogate input variables. For the input variables, there are two variables associated with each remaining waypoint, one for the power that will be consumed heading towards the waypoint and one for the distance to travel to a waypoint from the previous waypoint (due to turning while moving forward, the distance is more than the linear distance between the waypoints). Since there are 50 waypoints, there are 100 random variables needed. This yields $2(103) + 1 = 207$ samples, which is relatively small compared to what Monte Carlo sampling would require to achieve the same performance. As the rover visits waypoints, the number is reduced, so 207 samples is only the maximum. When only a single waypoint is left, only $2(5) + 1 = 11$ sigma points are needed.

4.4. Discussion

The two scenarios demonstrate the importance of the future input characterization problem. Even though the rover is a complex system, in the first scenario, the simple assumption that the future inputs will look like the past inputs was sufficient for accurate and precise predictions. The additional power required by turns was captured using the statistics of a window of past samples, so that they did not have to be explicitly accounted for and assuming constant future inputs was sufficient. Using system-level information about the waypoints the rover would visit improved significantly on the uncertainty associated with the future inputs but did not significantly impact accuracy.

In the second scenario, the assumption that the future inputs look like the past inputs did not provide as accurate or precise results as with the first scenario. Performance could have been potentially worse if the rover did not cycle through different speed settings and instead, for example, always increased the speed for the next waypoint. In this scenario, truly accurate predictions could be made only when system-level knowledge was utilized to predict the battery inputs. This approach still made some simplifying assumptions allowing a component-level prognostics approach to still be used.

Given a particular method for future input characterization, Monte Carlo sampling, UT sampling, and inverse FORM all had comparable accuracy and precision. With Monte Carlo sampling, a large number of samples were used and with smaller numbers of samples, performance decreases, so the number of samples required depends on the prognostics performance requirements. In this sense Monte Carlo sampling has an advantage because its computational complexity can be tuned. In addition it is the relatively simplest approach to implement of the three methods. A disadvantage is that it is a stochastic algorithm, which can be problematic for verification and certification procedures (Daigle, Saxena, & Goebel, 2012). UT sampling, in contrast, is a determinstic algorithm, and it selects only the minimal number of samples and this number grows only linearly with the number of random variables. A disavdvantage though is that it computes only mean and variance of the predictions, which may not be enough information in some cases. Inverse FORM, on the other hand, is not only deterministic but also allows control of both computational complexity and accuracy by selecting desired CDF values and computing the corresponding percentile values of k_E. The probability distribution of k_E can be reconstructed from that information and any desired statistical moments may be calculated. For each inverse CDF calculation, three to four iterations are usually required for optimization convergence. If the number of random variables is denoted by n (length of vector ω), each iteration of Inverse-FORM requires $n+1$ sample evaluations of $\mathtt{G}$, where one evaluation is required for computing k_E and n evaluations for computing the gradient vector of k_E. Therefore, if it is desired to repeat inverse FORM for k different CDF values, then $k \times 4 \times (n+1)$ evaluations of $\mathtt{P}$ are required. Thus, the computational complexity linearly increases with the number of random variables and results in increased information regarding uncertainty.

5. Conclusions

In this paper, we provided a general formulation of the prognostics problem and its uncertainty. Given descriptions of the sources of uncertainty, i.e., state uncertainty, parameter uncertainty, future input uncertainty, and process noise, we provided an algorithmic framework for incorporating this uncertainty into the predictions. With the novel concept of surrogate variables, we presented three methods for propagating the uncertainty: Monte Carlo sampling, unscented transform sampling, and the inverse first-order reliability method. Using battery prognostics on a planetary rover as a case study, we proposed two future input characterization methods and compared the performance of the different prediction algorithms for these methods for different scenarios in simulation. All approaches had similar performance, yet each offer different advantages and disadvantages that suggest when one would be preferred over another.

In future work, we will further investigate these ideas on other systems, and further develop the uncertainty quantification framework. While the proposed methods are promising for estimating uncertainty in prognostics, their applicability to multi-modal probability distributions, particular in the context of remaining useful life estimation, needs to be investigated. Further, we will also focus on model uncertainty quantification and develop methods for estimating model errors and model parameter uncertainty separately, instead of simply using lumped process noise terms.

Acknowledgment

This work was funded in part by the NASA System-wide Satefy Assurance Technologies (SSAT) project under the Aviation Safety (AvSafe) Program of the Aeronautics Research Mission Directorate (ARMD), and by the NASA Automated Cryogenic Loading Operations (ACLO) project under the Office of the Chief Technologist (OCT) of Advanced Exploration Systems (AES).

References

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.

Balaban, E., Narasimhan, S., Daigle, M., Celaya, J., Roychoudhury, I., Saha, B., … Goebel, K. (2011, September). A mobile robot testbed for prognostics-enabled autonomous decision making. In *Annual conference of the prognostics and health management society* (p. 15-30). Montreal, Canada.

Ceraolo, M. (2000, November). New dynamical models of lead-acid batteries. *IEEE Transactions on Power Systems*, *15*(4), 1184–1190.

Chen, M., & Rincon-Mora, G. A. (2006, June). Accurate electrical battery model capable of predicting runtime and I-V performance. *IEEE Transactions on Energy Conversion*, *21*(2), 504 - 511.

Daigle, M., Bregon, A., & Roychoudhury, I. (2012, September). A distributed approach to system-level prognostics. In *Annual conference of the prognostics and health management society* (p. 71-82).

Daigle, M., & Goebel, K. (2010, October). Improving computational efficiency of prediction in model-based prognostics using the unscented transform. In *Proc. of the annual conference of the prognostics and health management society 2010.*

Daigle, M., & Goebel, K. (2013, May). Model-based prognostics with concurrent damage progression processes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(4), 535-546.

Daigle, M., Saha, B., & Goebel, K. (2012, March). A comparison of filter-based approaches for model-based prognostics. In *Proceedings of the 2012 IEEE Aerospace Conference.*

Daigle, M., Saxena, A., & Goebel, K. (2012, September). An efficient deterministic approach to model-based prediction uncertainty estimation. In *Annual conference of the prognostics and health management society* (p. 326-335).

Haldar, A., & Mahadevan, S. (2000). *Probability, reliabil-*

ity, and statistical methods in engineering design. John Wiley & Sons, Inc.

Julier, S. J. (1998, April). A skewed approach to filtering. In *Proc. aerosense: 12th int. symp. aerospace/defense sensing, simulation and controls* (Vol. 3373, p. 54-65).

Julier, S. J., & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of the 11th international symposium on aerospace/defense sensing, simulation and controls* (pp. 182–193).

Julier, S. J., & Uhlmann, J. K. (2004, Mar). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401–422.

Ling, Y., Shantz, C., Mahadevan, S., & Sankararaman, S. (2011). Stochastic prediction of fatigue loading using real-time monitoring data. *International Journal of Fatigue*, *33*(7), 868–879.

Luo, J., Pattipati, K. R., Qiao, L., & Chigusa, S. (2008, September). Model-based prognostic techniques applied to a suspension system. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *38*(5), 1156 -1168.

Orchard, M., & Vachtsevanos, G. (2009, June). A particle filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*(3-4), 221-246.

Saha, B., Quach, C. C., & Goebel, K. (2012, March). Optimizing battery life for electric UAVs using a Bayesian framework. In *Proceedings of the 2012 IEEE Aerospace Conference.*

Sankararaman, S., Daigle, M., Saxena, A., & Goebel, K. (2013, March). Analytical algorithms to quantify the uncertainty in remaining useful life prediction. In *Proceedings of the 2013 IEEE Aerospace Conference.*

Sankararaman, S., & Goebel, K. (2013, April). Uncertainty quantification in remaining useful life of aerospace components using state space models and inverse form. In *Proceedings of the 15th Non-Deterministic Approaches Conference.*

Sankararaman, S., Ling, Y., Shantz, C., & Mahadevan, S. (2011). Uncertainty quantification in fatigue crack growth prognosis. *International Journal of Prognostics and Health Management*, *2*(1).

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*, *1*(1).

BIOGRAPHIES

Matthew Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. During the summers of 2006 and 2007, he was an intern with Mission Critical Technologies, Inc., at NASA Ames Research Center. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems. Dr. Daigle is a member of the Prognostics and Health Management Society and the IEEE.

Shankar Sankararaman received his B.S. degree in Civil Engineering from the Indian Institute of Technology, Madras in India in 2007 and later, obtained his Ph.D. in Civil Engineering from Vanderbilt University, Nashville, Tennessee, U.S.A. in 2012. His research focuses on the various aspects of uncertainty quantification, integration, and management in different types of aerospace, mechanical, and civil engineering systems. His research interests include probabilistic methods, risk and reliability analysis, Bayesian networks, system health monitoring, diagnosis and prognosis, decision-making under uncertainty, treatment of epistemic uncertainty, and multidisciplinary analysis. He is a member of the Non-Deterministic Approaches (NDA) technical committee at the American Institute of Aeronautics, the Probabilistic Methods Technical Committee (PMC) at the American Society of Civil Engineers (ASCE), and the Prognostics and Health Management (PHM) Society. Currently, Shankar is a researcher at NASA Ames Research Center, Moffett Field, CA, where he develops algorithms for uncertainty assessment and management in the context of system health monitoring, prognostics, and decision-making.

Improving the Diagnostic Performance for Dynamic Systems through the use of Conflict-Driven Model Decomposition

Anibal Bregon[1], Alexander Feldman[2], Belarmino Pulido[3], Gregory Provan[4], and Carlos Alonso-González[5]

[1,3,5] *Depto. de Informática, Universidad de Valladolid, Spain*
{anibal,belar,calonso}@infor.uva.es

[2,4] *Dept. of Computer Science, University College Cork, Ireland*
alex@llama.gs;g.provan@cs.ucc.ie

Abstract

This work studies potential ways of integration of two techniques for fault detection, isolation, and identification in dynamic systems: Lydia-NG suite of diagnosis algorithms and Consistency-based Diagnosis with Possible Conflicts. By integrating both techniques, Lydia-NG will benefit from a more efficient fault detection and isolation task, and Possible Conflicts will benefit from the identification capabilities of Lydia-NG. In this paper, we define a common framework that integrates both techniques, and then we apply the proposed integrated approach to a three-tank system, and draw some conclusions about potential ways of integration.

1. Introduction

The need for safety and reliability in engineering systems provides the motivation for developing Integrated Systems Health Management (ISHM) methodologies that include efficient fault diagnosis mechanisms. In this work we focus on model-based approaches to on-line fault diagnosis of dynamic systems. Online methods for model-based diagnosis require the use of quick and robust fault detection methods to establish discrepancies between observed and expected system behavior. Discrepancies due to faults trigger fault isolation processes that are responsible for determining the root cause of the fault. However, accurate and timely online fault diagnosis of complex dynamic systems is difficult and can be computationally expensive (Pouliezos & Stavrakakis, 1994; Isermann, 2006; Gertler, 1998).

In this work we have used the Lydia-NG modeling language and the Lydia-NG suite of algorithms (Feldman, Provan, & Gemund, 2010). The main idea of Lydia-NG is to perform multiple simulations for various hypothesized health states of the plant. The output of these multiple simulations is then processed and combined into single diagnostic output. Lydia-NG implements several strategies for the generation of fault candidates and a number of algorithms for active testing. These algorithms are based on AI search and include best-first, and bottom-up greedy search. Lydia-NG has been successfully used for complex applications like space satellites (Feldman, Castro, Gemund, & Provan, 2013). However, when applied to online fault diagnosis of large dynamic systems, running all the hypothesized health states becomes a quite difficult and time consuming task.

Several approaches have been proposed in recent years to deal with the complexity issue. System decomposition methods, have been proposed to reduce the complexity in the fault diagnosis task (Bregon, Biswas, & Pulido, 2012) by generating smaller simulation submodels which can run in parallel and provide independent diagnosis decisions. The Possible Conflict, PC, approach (Pulido & Alonso-González, 2004), is an off-line dependency compilation technique from the DX community, which decomposes the global system model into minimal submodels, and performs on-line behavior estimation using simulation (Bregon, Alonso, Biswas, Pulido, & Moya, 2012), state observers (Daigle, Bregon, & Roychoudhury, 2012), dynamic Bayesian networks (Alonso-González, Moya, & Biswas, 2011), or state-based neural networks (Pulido, Zamarreño, Merino, & Bregon, 2012). If a discrepancy is found, a set of fault candidates is generated by a minimal hitting-set algorithm of the triggered PCs. However, additional techniques must be used to refine the set of

Anibal Bregon et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

fault candidates.

The goal of this work consists of integrating PCs within the Lydia-NG diagnosis framework. First, PCs will decompose the global simulation model into a set of smaller simulation submodels. Then, PCs will be used for efficient online fault detection and fault localization, providing a subset of fault candidates from the minimal hitting-set of the deviated PC residuals. The subset of fault candidates is then used as input to Lydia-NG, where simulations are run only for each one of the fault candidates, and its result is processed and combined to provide the diagnosis output.

We have tested the proposed diagnosis framework by using a three-tank system. Theoretical study on the integration proposal shows that the complexity of LYDIA-NG is highly reduced by the inclusion of PCs in the framework. The experimental results present a particular diagnosis scenario where the proposed integration framework is used.

The rest of the paper is organized as follows. Section 2 presents the basic definitions and running example used in this work. Section 3.1 briefly introduces Lydia-NG, and section 3.2 describes basic ideas of system decomposition using PCs. Section 4 presents our proposal to integrate PCs within the Lydia-NG diagnosis framework. Section 5 describes the experimental results obtained for a three-tank system. Section 6 presents related work. And, finally, section 7 presents the discussion and conclusions.

2. Concepts and Definitions

In this section we present our basic definitions and a running example that we use to illustrate the significant concepts of this paper.

Since both LYDIA-NG and PCs are model-based diagnosis approaches, we provide a set of definitions about models and faults that will enable us to further explain both techniques using the same framework.

2.1. Definitions

For the purpose of this work we will focus our description on continuous systems diagnosis, with only one nominal state, and whose behavior can be described as a set Σ of Ordinary Differential Equations (ODEs). The model of our system will be the basic system description to perform diagnosis:

Definition 1 (Model)**.** The system model can be defined as $M(\Sigma, U, Y, X, \Theta)$, where: Σ is a set of ODEs, defined over a collection of known and unknown variables: U is a set of inputs, Y a set of outputs, X a set of state and intermediate i.e., unknown) variables, and Θ is the set of Model Parameters.

Definition 2 (System Description)**.** SD is made up of (M, H, σ, Π), where,

- H is the health-vector defined by means of $(h_1, \ldots, h_k)$, health variables, that allow us to characterize the set of states in the system, i.e. each $h_i \in H$ is a potential mode for the system. either nominal or faulty.
- σ is a mapping function: $\sigma(M, H_c) \rightarrow M_{Hc}(\Sigma_{Hc}, U_{Hc}, Y_{Hc}, X_{Hc}, \Theta_{Hc})$, that given the system description, M, and the current health status, Hc, provides the model for behavior estimation for the current mode (or current system description M_{Hc}): $\Sigma_{Hc} \subseteq \Sigma$, $U_{Hc} \subseteq U$, $Y_{Hc} \subseteq Y$, $X_{Hc} \subseteq X$, and $\Theta_{Hc} \subseteq \Theta$).
- Π is a mapping function $\Pi(\theta_{cc}) \rightarrow \{Hc \mid Hc \subseteq H\}$ that, given a set of parameters, provides the set of health statuses that relate to the set of model parameters: $\theta_{cc} \subseteq \Theta_{cc}$.

For consistency-based diagnosis using PCs, we only use $\sigma(M, Hn)$ with Hn corresponding to a nominal mode. Since we are dealing with a continuous system working in one nominal mode, we can compute offline the set of PCs for $M_{Hn}(\Sigma_{Hn}, U_{Hn}, Y_{Hn}, X_{Hn}, \Theta_{Hn})$, as will be described later.

An implicit assumption in our modeling approach is that we can use the same set of equations for both the nominal behavior estimation and the faulty behavior estimation, just changing the value of θ_i in equation $c_i \in \Theta$.

Consistency Based Diagnosis (CBD) performs fault detection and fault isolation using only models of correct behavior in a *two stage process.* First, we identify if there exists a discrepancy between the observed behavior (obtained via sensor outputs y_i) and the expected behavior (estimated by the Possible Conflict, $\hat{y}_{i_{pc}}$). We define a discrepancy in terms of a residual.

Definition 3 (Residual)**.** A residual is a real-valued measure $R(y_i, \hat{y}_{i_{pc}})$ of the difference between real and simulated system output at time t.

Corresponding to each residual is a conflict (Reiter, 1987). Hence, fault detection consists of computing every conflict.

The second step is fault isolation. *Fault isolation* consists of computing the minimal hitting sets of the conflicts. Conflicts are important since the *set of minimal diagnoses* of a system is given by the minimal hitting set of the set of minimal conflicts (Reiter, 1987). Intuitively, a conflict is a set of components that cannot be-have properly simultaneously, given the system description and current observations of abnormal behavior. In other words, given a system description and some current observations, they entail that at least one component of the conflict must be faulty, in the sense that it departs from the behavior described by its model of correct behavior.

There is no general framework for CBD of dynamic systems. Nevertheless, most existing approaches rely upon an iterative process of behavior estimation, conflict detection, and fault candidate generation (i.e. fault isolation). In this work we use the Possible Conflicts (PCs) approach to avoid the on-line computation of conflicts and speed up overall fault isolation. PCs are designed to compute off-line those subsystems capable of becoming conflicts online.

The output of the consistency-based diagnosis using PCs is a set of fault candidates C defined in the lattice provided by Θ^*.

2.2. Running Example

In this paper, we use the three-tank system shown in figure 1 as the running example. The three tanks are denoted as T_1, T_2, and T_3. They all have the same area $A_1 = A_2 = A_3 = 3$ [m^2]. The experiments are performed assuming the gravity $g = 10$ and the liquid with density $\rho = 1$.

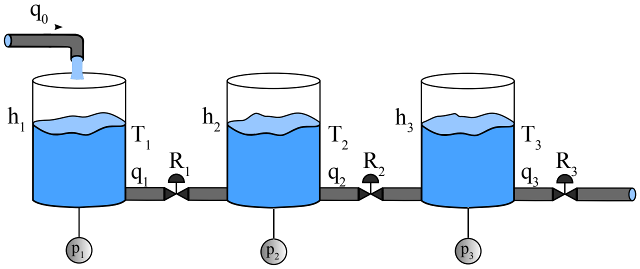


Figure 1. Diagram of the three-tank system.

Tank T_1 gets filled from a pipe q_0 with a constant flow of 1.5 [m^3/s]. It drains into T_2 via a pipe q_1. The liquid level is denoted as h_1. There is a pressure sensor p_1 connected to T_1 that measures the pressure in Pascals [Pa]. Starting from the Newton's (and Bernoulli's) equations and manipulating them (the actual derivation is trivial and irrelevant in this paper) we derive the following Ordinary Differential Equation (ODE) that gives the level of the liquid in T_1:

$$\frac{dh_1}{dt} = \frac{q_0 - k_1\sqrt{h_1 - h_2}}{A_1} \tag{1}$$

In eq. 1, the coefficient k_1 is used to model the area of the drainage hole and its friction factor. We emphasize the use of k_1 because, later, we will be "diagnosing" our system in term of changes in k_1. Consider a physical valve R_1 between T_1 and T_2 that constraints the flow between the two tanks. We can say that the valve changes proportionally the cross-sectional drainage area of q_1 and hence k_1. The diagnostic task will be to compute the true value of k_1, given p_1, and from k_1 we can compute the actual position of the valve R_1. The water levels of T_2 and T_3, denoted as h_2 and h_3 respectively, are given by:

$$\frac{dh_2}{dt} = \frac{k_1\sqrt{h_1 - h_2} - k_2\sqrt{h_2 - h_3}}{A_2}, \tag{2}$$

$$\frac{dh_3}{dt} = \frac{k_2\sqrt{h_2 - h_3} - k_3\sqrt{h_3}}{A_3}. \tag{3}$$

Values k_1, k_2, and k_3, are constant values with no physical meaning, and we have set them with a value of 0.75. Finally, we turn the water level into pressure:

$$p_i = \frac{g\, h_i\, A_i}{A_i} = g\, h_i \tag{4}$$

where i is the tank index ($i \in \{1, 2, 3\}$). Hence, there are three equations relating the pressure in the tank with the level in the tank. To observe the behavior of the system we have an observational model, that allows us to know or read each value p_i. We use p_i^* to distinguish the measured variable from the model output p_i:

$$p_i^* = p_i \tag{5}$$

It is assumed that the initial water level in the three tanks is zero. Additionally, we make explicit the relation between the state variables, h_i in our example, and their derivatives, dh_i:

$$h_i = \int dh_i \cdot dt \tag{6}$$

These equations allow us to select an integral or differential approach for behavior simulation, depending on the selected causality. These equations make no influence in the diagnosis results, because they will have no θ_i, and consequently no health status.

3. Algorithms

This section presents the fundamental ideas of the Lydia-NG diagnosis framework and the structural model decomposition approach with PCs.

3.1. Lydia-NG

We next show an algorithmic framework for computing diagnoses. The basic idea of the Lydia-NG diagnostic library (shown in Fig. 2) is to perform multiple simulations for various hypothesized health states of the plant. The output of these multiple simulations is then processed and combined into single diagnostic output.

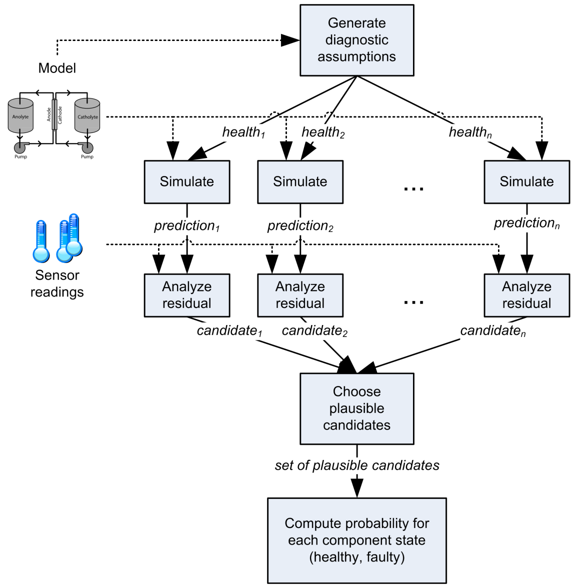


Figure 2. Overview of the Lydia-NG diagnostic method

The Lydia-NG diagnostic library consists of the following building blocks:

Generator of Diagnostic Assumptions: A diagnostic assumption is a set of hypothetical assignments for the health or fault state of each component in the system. The "all nominal" diagnostic assumption assigns healthy status to each component. Lydia-NG allows one nominal and one or more faulty states per component.

Simulation Engine: Given a diagnostic assumption, Lydia-NG can construct a simulation model of the system. This simulation model consists of equations. By solving this system of equations Lydia-NG computes values for one or more *observable* variables. The values of these observable variables is also referred to as a *prediction.*

Residual Analysis Engine: A prediction is compared to the sensor data by a residual analysis engine. This engine combines the individual discrepancies in each sensor data/predicted variable pair to produce a single real value that indicates how close is the prediction of the simulation engine to the sensor data obtained from the plant. A simulation that results in all predicted values coincide with the measured ones will result in the residual being close to zero. The data structure containing predictions, their corresponding sensor data and the computed residual is called a *diagnostic candidate* or simply *candidate.*

Candidate Selection Algorithm: Not all candidates generated by the residual analysis engine are used for computing the final system health. The candidate selection algorithm discards each candidate whose residual is larger than the residual of the "all nominal" candidate.

System State Estimation Algorithm: Lydia-NG uses the set of candidates that is computed by the candidate selection algorithm to compute an estimate for the health of each component. This is done by the system state estimation algorithm.

Algorithm 1 shows the top-level diagnostic process. The inputs to Alg. 1 are a model and a scenario, and the result is a diagnosis.

Algorithm 1 supports a large variety of simulation methods that may or may not use time as an independent variable. The only requirement toward the simulation engine is to predict a number of variables whose types can be mapped to Lydia-NG and to be relatively fast.

Algorithm 1 Diagnosis framework

```
function DIAGNOSE(SCN) returns a diagnosis
    inputs: SCN, diagnostic scenario
    local variables: h, FDI vector, health assignment
                     p, real vector, prediction
                     Ω, a set of diagnostic candidates
                     DIAG, diagnosis, result
    while $\mathbf{h} \leftarrow$ NEXTHEALTHASSIGNMENT() do
        $\mathbf{p} \leftarrow$ SIMULATE($M, \gamma, \mathbf{h}$)
        $r \leftarrow$ COMPUTERESIDUAL($\mathbf{p}, \alpha$)
        $\Omega \leftarrow \Omega \cup \langle \mathbf{h}, r \rangle$
    end while
    DIAG ← COMBINECANDIDATES($\Omega$)
    return DIAG
end function
```

The basic idea of Alg. 1 is to simulate for various health assignments and to compare the predictions with the observed sensor data (i.e., telemetry). There are several important aspects of this algorithms that ultimately affect the diagnostic accuracy as measured by various performance metrics.

The first algorithmic property that determines many of the diagnostic performances is the order in which health-assignments are generated. In Alg. 1 this is implemented in the NextHealthAssignment function. The latter subroutine also determines when to stop the search and should be properly parametrized depending on the model and the user requirements. In the standard LYDIA-NG diagnostic library we provide the following diagnostic search policies:

Breadth-First Search (BFS): This policy first generates the nominal health assignment, then single-faults, double-faults, etc.

Depth-First Search (DFS): This search policy starts with the nominal health assignment, then adds a single-fault, continues with a double fault including the first, and so on, until all components are failed. After the all-faulty assignment is generated, the algorithm backtracks one step and generates a sibling assignment and continues traversing down and backtracking in the same manner until no more backtracking is possible.

Backwards Greedy Stochastic Search (BGSS): In this mode, the search start from the all-faulty assignment. A random health variable is then flipped and the flip is retained iff the flip leads to a decrease in the residual. The order of health variables is arbitrary. As the whole search process is stochastic, it needs to be run multiple iterations in order to achieve the desired completeness. A formal description of this method for Boolean circuit models can be found in (Feldman et al., 2010).

Each simulation produces what we call a *candidate*: a set of predicted values for a given health-assignment. The second important property of Alg. 1 is the comparison and ordering of the diagnostic candidates. This is done by mapping the predicted and observed variables into a single real-number, called a *residual*.

Residual generation functions in LYDIA-NG bear resemblance to loss functions in decision theory. For example, residuals may be squared or absolute residuals (Feldman et al., 2013). A disadvantage of the squared residuals function R_{sq} is that it adds a lot weight to outliers. In decision theory, the absolute loss function that corresponds to the R_{abs} function is discontinuous. The latter, however, is not a problem for the algorithms described in this paper and we prefer R_{abs} over R_{sq}.

3.2. Consistency-based diagnosis with PCs

3.2.1. Model decomposition with PCs

The Possible Conflicts (PCs) approach (Pulido & Alonso-González, 2004) is a model decomposition method that finds (off-line) every subset of equations capable of generating conflicts. PCs provide the structural and causal model of a subsystem with minimal redundancy. The set of equations in a PC can be used to simulate the correct behavior of the subsystem. Hence, PCs can be used in CBD of dynamic systems (PULI01, 2001). For the sake of self-containment, we summarize here the proposal for PCs computation given in (Pulido & Alonso-González, 2004).

To compute PCs, we need the structural model of the system under study, which can be obtained from the set of equations in the system description, once we select a given working mode, tailored for our new problem formulation, instead of the original process which was suitable for system descriptions provided as hypergraphs (Pulido & Alonso-González, 2004).

We will illustrate the process using the three-tank system in Fig. 1, and the set of equations in its model as described in Section 2.2.

We need an abstraction of our model description $SD = (M, H, \sigma, \Pi)$. Let's assume we compute the set of PC for a given nominal mode characterized by H_n. Using $\sigma(M, H_n)$, we obtain $M_{H_n} = (\Sigma_{H_n}, U_{H_n}, Y_{H_n}, X_{H_n}, \Theta_{H_n})$. For the structural model, we only need the information about the measured and unknown variables in each model equation. Thus each equation in $\Sigma_{H_{n_i}}$ will provide one structural constraint $(c_i,\ S_i,\ X_i)$, where S_i accounts for the measured variables from U_{H_n}, Y_{H_n} in c_i, and X_i accounts for the unknown (state or intermediate variables in c_i).

For the three tank system the structural model is made up of the following constraints:

Constraint	*Sensors*	*Unknowns*
c_1	$\{q_0\}$	$\{d_{h1}, h_1, h_2\}$
c_2	$\{\}$	$\{d_{h2}, h_1, h_2, h_3\}$
c_3	$\{\}$	$\{d_{h3}, h_2, h_3\}$
c_4	$\{\}$	$\{p_1, h_1\}$
c_5	$\{\}$	$\{p_2, h_2\}$
c_6	$\{\}$	$\{p_3, h_3\}$
c_7	$\{p_1^*\}$	$\{p_1\}$
c_8	$\{p_2^*\}$	$\{p_2\}$
c_9	$\{p_3^*\}$	$\{p_3\}$
c_{10}	$\{\}$	$\{h_1, d_{h1}\}$
c_{11}	$\{\}$	$\{h_2, d_{h2}\}$
c_{12}	$\{\}$	$\{h_3, d_{h3}\}$

where constraints c_1 to c_3 are related to equations (1) to (3); constraints c_4 to c_6 are related to the equation (4) for each one of the tanks; constraints c_7 to c_9 make explicit the diagnosis observational model, relating the output variable p_i and its associated sensor p_i^*; and constraints c_{10} to c_{12} make explicit the dynamic in the system: relation between the derivative of the state variables and the state variable itself.

The first step in PC computation is to look for the complete set of minimally redundant subsets of equations. The redundancy is related to the set of unknown variables in the equations.

Definition 4 (Minimal Evaluation Chain (MEC))**.** A subset of equations that represents a potential source of discrepancy if the set of equations could be actually solved[1].

A MEC represents a strictly overdetermined[2] set of equations that can potentially be solved using local propagation (elimination method): each MEC will have n constraints and $n-1$ unknowns. MECs are computed to guarantee that there is a complete matching in its associated bipartite graph (made up of unknowns as nodes and equations as edges). This is a necessary condition to obtain a causal assignment and potentially provide a computational model.

A summary of the algorithms used to compute MECs in a system can be found in Appendix A. The set of MECs in the system in Fig. 1 is:

- $mec_1 = \{c_7, c_4, c_{10}, c_1, c_5, c_8\}$
- $mec_2 = \{c_8, c_5, c_{11}, c_2, c_4, c_6, c_7, c_9\}$
- $mec_3 = \{c_9, c_6, c_{12}, c_3, c_5, c_8\}$

We need to know the different ways an equation can be solved, because we can deal with non-linear models. These ways are usually called the set of possible causal assignments for the variables in an equation. Using this set of causal assignments, we can define the set of possible causal assignments for every unknown variable in the bipartite graph. We assume that the set of possible causal assignments is known for the system model, and we build the complete set of valid causal assignments for the set of MECs, using exhaustive search (Pulido & Alonso-González, 2004). We call each valid causal assignment Minimal Evaluation Model (MEM), because each MEM represent the precise order to solve or to evaluate the overdetermined set of equations in a MEC. Given the equations and the evaluation order provided by a MEM, they can be used to build simulation models[3].

Definition 5 (Minimal Evaluation Model (MEM))**.** A MEM is a MEC with a valid global causal assignment for every unknown in the MEC.

For the three tank system, we assume that the causality is given by the expression in equations (1) to (6), except for the observational model (in this case we allow solving constraints ec_7 to ec_9 in both directions because we need to convert some system measurements Y in MEM inputs). The set of MEMs for the three-tank system and their discrepancy nodes are shown in Table 1.

3.2.2. Fault detection and isolation using PCs

In the MEM there is a special node called *discrepancy* node (representing the only variable that is estimated by two different ways). Therefore, that node is the potential source of a discrepancy using only the values of measured variables as inputs, and the past value of state-variables.

In CBD (Reiter, 1987; Kleer & Williams, 1987) a conflict arises given a discrepancy between observed and predicted values for a variable. Under fault conditions, conflicts are observed when the model described by a MEM is evaluated with available observations and produce a discrepancy, because the model equations and the input/measured values are inconsistent (Reiter, 1987; Kleer & Williams, 1987). This notion of possible discrepancy generation leads to the definition of *Possible* Conflict:

Definition 6 (Possible Conflict)**.** The set of constraints in a MEC that give rise to at least one MEM.

Every MEC in the three-tank system has one MEM. Then, there are three PCs in the system, one for each MEC.

Each MEM is the computational model for a PC, and each equation in a MEM contains zero or more parameters that can be the source of potential faults (θ_{cc} in our model description). The set of parameters related to each PC is also shown in the fourth column in Table 1. Given a non-zero residual, we then isolate the fault parameters involved in the pc structural model: Θ_{pc}. This information is the basis for the integration of Consistency-based diagnosis of dynamic systems with Possible Conflicts and Lydia-NG.

4. On-line Fault diagnosis with Lydia-NG and PCs

In CBD, diagnosis must discriminate among 2^N behavioral mode assignments when just correct, $ok(\cdot)$, and incorrect modes, $\neg ok(\cdot)$, are present for N components. When B behavioral models are allowed, diagnosis must discriminate among B^N mode assignments. This is the problem faced by any model-based diagnosis proposal which attempts fault identification (DRES96, 1996), and it is also present in Lydia-NG. In this section, we present an integration proposal, where the system model is partitioned using PCs. As explained in Section 2, the output of the consistency-based diagnosis using PCs is a set of

[1] A MEC is equivalent to the structure of an Analytical Redundancy Relation, ARR, or a Minimally Structurally Overdetermined set, MSO, in works by Staroswiecki and co-workers and Nyberg and co-workers, respectively.

[2] A redundant set of equations would be an Evaluation Chain. Since we are interested only on minimal conflicts, we just focus on the set of MECs that are by definition minimally overdetermined.

[3] A MEM is equivalent to the analytical expression of an ARR, an R-Conflict in the work by Cordier and the HIMALAYA group, and also is the set of formulas used to compute a conflict in GDE.

Table 1. MEMs for the three-tank system and their discrepancy nodes.

MEM	Associated MEC	Discrepancy node	Fault Parameters
$\{c_7, c_4, c_{10}, c_1, c_5, c_8\}$	mec_1	p_1^*	k_1, A_1
$\{c_8, c_5, c_{11}, c_2, c_4, c_6, c_7, c_9\}$	mec_2	p_2^*	k_1, k_2, A_2
$\{c_9, c_6, c_{12}, c_3, c_5, c_8\}$	mec_3	p_3^*	k_2, k_3, A_3

fault candidates C defined in the lattice provided by Θ^*. Then, this set of diagnosis candidates is used as input to Lydia-NG, thus reducing the number of health state simulations that needs to be considered by Lydia-NG.

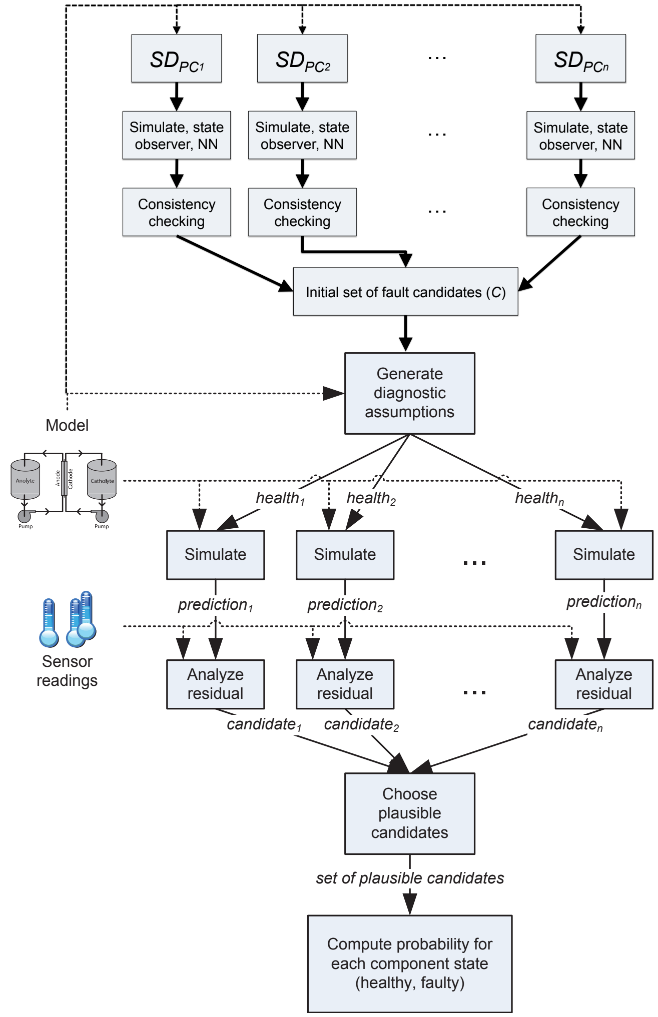


Figure 3. Lydia-NG and PCs integration framework.

Fig. 3 shows the basic idea of our integration proposal. The simulation model for each PC uses some of the system measurements as input, and provides an estimation for exactly one variable (the potential discrepancy). Then, an executable model, SD_{pc_i} for each pc_i, is built. This executable model can be a simulation model, a state observer, or even a neural network (Bregon, Biswas, & Pulido, 2012; Pulido et al., 2012). Summarizing, the integration of Lydia-NG and CBD with PCs is possible given the set of candidates, C: each candidate C_i is a subset of Θ_{pc}. Then invoking $\Pi(C_i)$, Lydia-NG can obtain the set of health statuses, Hc related to C_i, and use them as input for its search. Given the current implementation of Lydia-NG, we can obtain the system description (system model) imposed by $Hc : \sigma(M, Hc)$, which is enough to characterize the current model and perform simulation of the Hc health status.

Algorithm 2 shows the algorithm for our integrated diagnosis framework. Y_{pc_i} denotes the set of input observations available for the executable model of a PC, SD_{pc_i}; and $\hat{Y}_{pc_i}$ represents the set of predictions obtained from SD_{pc_i}. The function ObtainObservations obtains from the diagnostic scenario the observations which have to be used as input for each PC. Function EstimateBehavior provides an estimation of a measured variable by using the executable model of each PC (either a simulation model, a state observer model, or a neural network).

For the detection part, to determine significant deviations from the PC residuals (PC residuals are computed by using an absolute residual function). We use the Z-test for robust fault detection using a set of sliding windows as detailed in (Daigle et al., 2010). A small window, N_2, is used to estimate the current mean of the residual signal, μ_r. The variance of the nominal residual signal is computed using a large window N_1 preceding N_2, by a buffer N_{delay}, which ensures that N_1 does not contain any samples after fault occurrence. The variance and the confidence level determined by the user are then used to dynamically compute the detection thresholds ϵ_r^- and ϵ_r^+. Other approaches can be used to determine significant deviations from the residuals, such as the Dynamic Time Warping distance, DTW (Keogh & Ratanamahatana, 2005).

Once the initial set of fault candidates has been isolated, the Lydia-NG part of the algorithm is run (as shown in algorithm 2). The algorithm takes the set of isolated fault candidates as input, and the NextHealthAssignment function only considers the health assignments related to the fault candidates. In this version of the integrated framework, the global system model is used as the simulation model, instead of the PC submodels, thus providing a more direct way to integrate both approaches. In future versions of the

Algorithm 2 Integrated PCs and Lydia-NG diagnosis approach.

```
function PCs-Lydia-diagnosis(SCN) returns a diagnosis
    inputs: SCN, diagnostic scenario
    local variables: $Y_{pc_i}$, set of input observations
                     $\hat{Y}_{pc_i}$, estimation from the PC
                     $\Theta_{pc_i}$, fault parameters involved in the PC
                     $\mathbf{h}$, FDI vector, health assignment
                     $\mathbf{p}$, real vector, prediction
                     $\Omega$, a set of diagnostic candidates
                     DIAG, diagnosis, result
    repeat
        $Y_{pc_i} \leftarrow$ ObtainObservations(SCN)
        $\hat{Y}_{pc_i} \leftarrow$ EstimateBehavior($SD_{pc_i}, Y_{pc_i}$)
        $r_{pc_i} \leftarrow$ ComputeResidual($\hat{Y}_{pc_i}, Y_{pc_i}$)
        if $r_{pc_i} < \epsilon_r^-$ or $r_{pc_i} > \epsilon_r^+$ then
            $\Theta_{pc_i}$ = confirm $pc_i$ as a real conflict
            $C \leftarrow$ MHS($C, \Theta_{pc_i}$)
        end if
    until Every $pc_i$ is activated or time elapsed or a unique fault candidate has been isolated
    while $\mathbf{h} \leftarrow$ NextHealthAssignment($\Pi, C$) do
        $\mathbf{p} \leftarrow$ Simulate($M, \gamma, \mathbf{h}$)
        $r \leftarrow$ ComputeResidual($\mathbf{p}, \alpha$)
        $\Omega \leftarrow \Omega \cup \langle \mathbf{h}, r \rangle$
    end while
    DIAG $\leftarrow$ CombineCandidates($\Omega$)
    return DIAG
end function
```

framework, the PC submodels will also be used as the simulation model in Lydia-NG, thus providing faster simulation results.

5. Results

In this section we show some diagnosis results for our integrated framework. We present an on-line fault diagnosis scenario for a particular fault in the three-tank system and discuss the results obtained with and without our integrated framework.

5.1. Fault Diagnosis Scenarios

This section describes the scenario of nominal conditions for the system, and the fault scenario of partial blockage of valve R_1 at time 100 s.

Nominal Scenario: Figure 4 shows a simulation experiment for the three-tank model. It is intuitive from the tanks equations that the pressure in Tank 1, p_1, is larger than the pressure in Tank 2, p_2, which is larger than the pressure in Tank 3, p_3. This is confirmed by the plot in fig. 4. For this nominal scenario, none of the three PCs found for the system is triggered. The advantage of including PCs within the Lydia-NG framework is evident for this case. Since none of the PCs is triggered, Lydia-NG is not run, thus avoiding the time-consuming simulations for the different health states when no actual fault has occurred in the system. **Fault Scenario:** Now consider we introduce a 40% blockage fault in valve R_1 occurring at time 100 s. Figure 5 shows the plots of the three-tank system simulation for this fault, and Figure 6 shows the plots for the PCs for such a fault. The left column in Figure 6 shows the measured and estimated pressure for each one of the PCs, while the right column shows the residual signal computed for each PC.

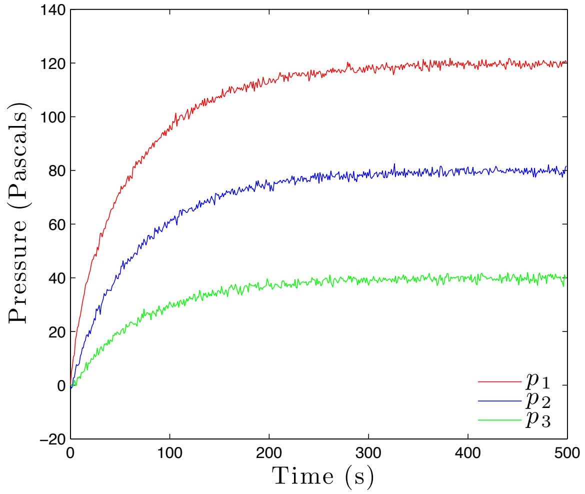


Figure 4. Simulation results for the three-tank model.

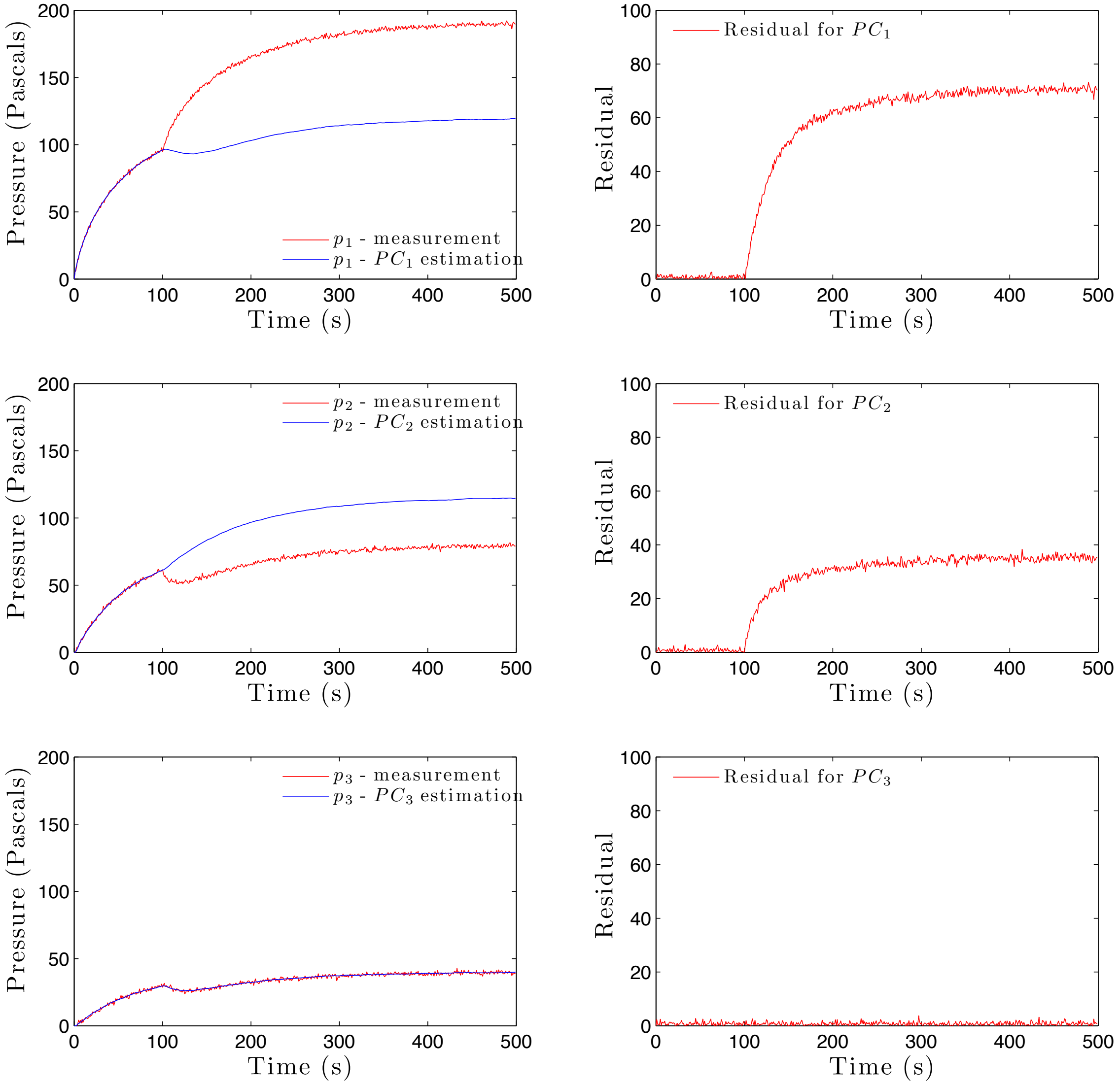


Figure 6. Simulation results for a 40% blockage fault at time 100 s in valve R_1.

5.2. On-line fault diagnosis

This section briefly describes how LYDIA-NG runs with and without the use of PCs. The first phase is residual analysis, where LYDIA-NG runs a set of simulations such that a residual is computed for each simulation: see Figure 2. Because LYDIA-NG uses real-value health variables, the space of potential diagnostic assumptions, and the corresponding set of simulations, is enormous, and infinite in the worst case. The heuristics used for generation of diagnostic assumptions are critical to the success and efficiency of LYDIA-NG.

LYDIA-NG ranks the residual outputs, discarding those candidates whose residual value is larger than the residual of the "all nominal" candidate. The remaining candidates are assigned probabilities of occurrence, using a method described in (Feldman et al., 2013). The fault isolation process assigns probabilities of failure to system components, and these are reported as ranked diagnoses.

In the following we compare the results for running LYDIA-NG with and without PCs. Without PCs, LYDIA-NG uses the global system model described earlier; with PCs (i.e., using Algorithm 2), the generation of diagnostic assumptions is governed by the PC-based algorithm.

For the diagnosis scenario with a 40% blockage fault in valve R_1 occurring at time 100 s, our results are as follows.

Non-PC-based Approach: LYDIA-NG computes residuals based on the difference between the pressures

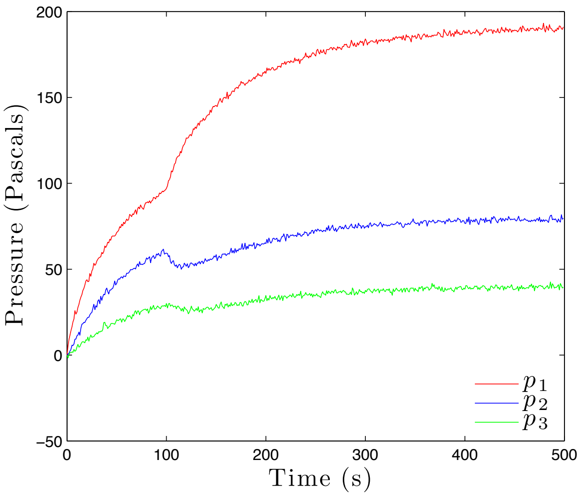


Figure 5. Simulation results for a 40% blockage fault at time 100 s in valve R_1.

shown in Figures 4 and 5. The non-zero residual at time 104 s creates a set of simulations in which LYDIA-NG analyzes several valve %-blockage cases for R_1, R_2 and R_3. LYDIA-NG estimates the valve positions by "guessing" the *true* valve positions and computes the health probability by subtracting the commanded valve position from the estimated one. LYDIA-NG is able to isolate the most-likely fault as (R_1, 40%).

PC-based Approach: When computing diagnoses for this fault, at time 104 s, an increase in the residual of PC_2 is detected, and consequently k_1, k_2, and A_2 are selected as the initial set of fault candidates. At the next time step, at time 105 s, PC_1 is triggered, thus selecting k_1 and A_1 as possible fault candidates. A minimal hitting set algorithm is run, determining that the only single fault candidate in the system is k_1. At this point, the fault identification for k_1 is triggered by using LYDIA-NG.

Running this diagnosis scenario with a (trivial) input of R_1 (as derived from the candidate k_1), as opposed to R_1, R_2 and R_3, results in an 80× speedup of LYDIA-NG as compared to the non-PC approach. This is a result of reducing the diagnosis assumption space.

6. Related work

LYDIA-NG belongs to a class of MBD methods that use continuous-valued models and sensor data, and use entropy based methods for test selection to disambiguate diagnoses. It is a generalization of LYDIA, which used discrete-value models.

In terms of diagnostics solvers, LYDIA-NG is related to the HyDE (Hybrid Diagnosis Engine) solver (Narasimhan & Brownston, 2007). The HyDE-S variant accepts as input interval-valued hybrid models and continuous-valued sensor data. Another solver, FACT (Daigle et al., 2010), can also use continuous-valued models and sensor data, but requires that the model be represented as a hybrid bond graph. Given an anomaly, FACT first uses an observer-based approach (adopted from the FDI community) with statistical techniques for robust fault detection. Fault isolation is performed using qualitative inference, i.e., by matching qualitative deviations caused by fault transients to those predicted by the model.

Recent works have demonstrated the similarities between model-based diagnosis approaches from the DX and the FDI communities (Cordier et al., 2004). In such framework, it has been demonstrated the equivalence of several structural model decomposition techniques such as PCs, minimal ARRs and Minimally Structurally Overdetermined sets (Armengol et al., 2009). As a consequence, the proposal in this work can be easily extended to other structural methods.

Using CBD we need to generate the set of candidates C and wait for every *pc* to be confirmed. An FDI approach would use exoneration using the structural information in the set of PCs. In CBD we wait for additional observations in order to reject modes that are not consistent with available information. Combining our results with LYDIA-NG provides an additional boost for candidate discrimination by including fault models through health statuses.

The approach can be further refined using qualitative information (for instance residual qualitative signatures), or add a quantitative parameter estimation (Bregon, Biswas, & Pulido, 2012). Additionally, different methodologies can be coupled to estimate correct and/or faulty behavior (Alonso-González et al., 2011).

This work is clearly a first step to combine both techniques for hybrid non-linear systems. PCs has been extended to work with hybrid systems (Bregon, Alonso, et al., 2012), and can greatly benefit from LYDIA-NG state estimation capabilities. The approach then would be similar to coupling different techniques for continuous/discrete state estimation as in (Hofbaur & Williams, 2004; Bayoudh, Travé-Massuyès, & Olivé, 2008).

7. Conclusions

This work has presented an integrated framework for on line fault detection, isolation and identification of dynamic systems.

Two different approaches have been integrated: The LYDIA-NG suite of diagnosis algorithms and the Possible Conflicts framework for on-line consistency based diagnosis. Lydia-NG is a simulation based diagnosis system that filters out diagnosis candidates discarding those of them that generates residuals larger than the *all-nominal* assumption, i.e., fault free and nominal system configuration. Although the system incorporates important facilities, such as diagnostic test generation based on entropy measure, its main drawback is the lack of focus for the initial set of candidates, which may be large, and the cost of simulating the complete system for every considered candidate. On the contrary, the set of Possible Conflicts identifies minimal computational subsystems that decompose the complete system and that can be simulated independently. PCs are based on Reiter's theory of diagnosis from first principles and are able to generate fault isolation candidates from model of correct behavior without hypothesizing an initial set of candidates. Hence, using consistency-based diagnosis with PCs candidate generation is rather efficient, although additional techniques are required to further refine fault candidates for fault isolation and identification. They also lack some of the facilities incorporated in Lydia-NG like generation of diagnostic tests.

In this paper we have combined both approaches, complementing each other, looking to preserve the best of each approximation. This integration can be tackled in different ways. We have opted for a simple integration approach that still is able to improve any of them. PCs are used to generate the initial set of isolation candidates *a la Reiter*. These candidates are later refined by Lydia-NG, which simulates the complete system in the modes –potential health statuses– defined by the candidates. In this way, we exploit PCs ability to generate isolation candidates with Lydia-NG ability to reject fault candidates that do not comply with current observations and diagnosis assumptions.

Our three tank system running example shows the potential of this approach. First, when the system is fault free, no PC becomes a real conflict and no candidate is generated. This avoids running Lydia-NG for fault detection, which is performed by the PCs approach, thus potentially providing a significant saving on computing time, depending on the size of the complete system and on the number and overlapping degree of the PCs. Second, when a fault is detected, PCs may generate a low number of fault candidates, depending on the number of PCs and its overlapping degree but also on the real faulty parameter, thus providing an automatic focus for Lydia-NG fault candidate search.

Acknowledgment

A. Bregon, B. Pulido, and C. Alonso's funding for this work was provided by the Spanish MCI TIN2009-11326 grant.

References

Alonso-González, C., Moya, N., & Biswas, G. (2011). Dynamic Bayesian network factors from possible conflicts for continuous system diagnosis. In *Proc. of the 14th Int. Conf. on Advances in AI* (pp. 223–232). Berlin: Springer-Verlag.

Armengol, J., Bregon, A., Escobet, T., Gelso, E., Krysander, M., Nyberg, M., et al. (2009). Minimal Structurally Overdetermined sets for residual generation: A comparison of alternative approaches. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS09* (p. 1480-1485). Barcelona, Spain.

Bayoudh, M., Travé-Massuyès, L., & Olivé, X. (2008). Coupling Continuous and Discrete Event System Techniques for Hybrid System Diagnosability Analysis. In *Proc. of the 18th European Conf. on Artificial Intelligence, ECAI08* (pp. 219–223). Amsterdam, The Netherlands.

Bregon, A., Alonso, C., Biswas, G., Pulido, B., & Moya, N. (2012). Fault Diagnosis in Hybrid Systems using Possible Conficts. In *Proc. of Safeprocess'12.* Mexico City, Mexico.

Bregon, A., Biswas, G., & Pulido, B. (2012). A Decomposition Method for Nonlinear Parameter Estimation in TRANSCEND. *IEEE Trans. Syst. Man. Cy. Part A*, *42*(3), 751-763.

Cordier, M., Dague, P., Lévy, F., Montmain, J., Staroswiecki, M., & Travé-Massuyès, L. (2004). Conflicts versus Analytical Redundancy Relations: a comparative analysis of the Model-based Diagnosis approach from the Artificial Intelligence and Automatic Control perspectives. *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics*, *34*(5), 2163-2177.

Daigle, M., Bregon, A., & Roychoudhury, I. (2012, aug). Qualitative Event-based Diagnosis with Possible Conflicts Applied to Spacecraft Power Distribution Systems. In *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* (p. 265-270).

Daigle, M., Roychoudhury, I., Biswas, G., Koutsoukos, X., Patterson-Hine, A., , et al. (2010, September). A Comprehensive Diagnosis Methodology for Complex Hybrid Systems: A Case Study on Spacecraft Power Distribution Systems. *IEEE Transac-*

tions of Systems, Man, and Cybernetics, Part A, *4*(5), 917–931.

Daigle, M., Roychoudhury, I., Biswas, G., Koutsoukos, X., Patterson-Hine, A., & Poll, S. (2010). A comprehensive diagnosis methodology for complex hybrid systems: A case study on spacecraft power distribution systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *40*(5), 917–931.

Dressler, O. (1996). On-line diagnosis and monitoring of dynamic systems based on qualitative models and dependency-recording diagnosis engines. In *Proceedings of the Twelfth European Conference on Artificial Intelligence, ECAI-96* (p. 461-465).

Feldman, A., Castro, H. V. de, Gemund, A. van, & Provan, G. (2013). Model-Based Diagnostic decision-support system for satellites. In *Aerospace Conference, 2013 IEEE* (pp. 1–14).

Feldman, A., Provan, G., & Gemund, A. van. (2010). Approximate Model-Based Diagnosis Using Greedy Stochastic Search. *Journal of Artificial Intelligence Research*, *38*, 371–413.

Gertler, J. (1998). *Fault Detection and Diagnosis in Engineering Systems.* Marcel Dekker, Inc.

Hofbaur, M., & Williams, B. (2004, Oct.). Hybrid estimation of complex systems. *IEEE T. Syst. Man. Cy. Part B*, *34*(5), 2178 -2191.

Isermann, R. (2006). *Fault-Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance.* Springer.

Keogh, E., & Ratanamahatana, C. (2005). Exact indexing of dynamic time warping. *Knowledge and Information Systems*, *7*(3), 358-386.

Kleer, J. de, & Williams, B. C. (1987). Diagnosing multiple faults. *Artificial Intelligente*, *32*, 97-130.

Narasimhan, S., & Brownston, L. (2007). HyDE–A General Framework for Stochastic and Hybrid Model-based Diagnosis. In *Proc. 18th International Workshop on Principles of Diagnosis (DXâĂŹ07), Nashville, USA* (pp. 162–169).

Pouliezos, A., & Stavrakakis, G. (1994). *Real Time Fault Monitoring of Industrial Processes.* Kluwer Academic Publishers.

Pulido, B., & Alonso, C. (2001). Dealing with cyclical configurations in MORDRED. In *IX Conferencia Nacional de la Asociacion Española de Inteligencia Artificial, (CAEPIA-01)* (p. 983-992). Gijon, Spain.

Pulido, B., Alonso, C., & Acebes, F. (2001). Lessons learned from diagnosing dynamic systems using possible conflicts and quantitative models. In *Engineering of Intelligent Systems. XIV Conf. IEA/AIE-2001* (Vol. 2070, p. 135-144). Budapest, Hungary.

Pulido, B., & Alonso-González, C. (2004, Octubre). Possible Conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics*, *34*(5), 2192-2206.

Pulido, B., Bregon, A., & Alonso-González, C. (2010). Analyzing the influence of differential constraints in Possible Conflicts and ARR Computation. In P. Meseguer, L. Mandow, & R. Gasca (Eds.), *Current Topics in Artificial Intelligence* (Vol. 5988, p. 11-21). Springer Berlin.

Pulido, B., Zamarreño, J., Merino, A., & Bregon, A. (2012, July). Using structural decomposition methods to design gray-box models for fault diagnosis of complex systems: a beet sugar factory case study. In A. Bregon & A. Saxena (Eds.), *Procs. of the First European Conference of the Prognostics and Health Management Society* (p. 225-238). Dresden, Germany.

Reiter, R. (1987). A Theory of Diagnosis from First Principles. *Artificial Intelligence*, *32*, 57-95.

A. Possible Conflicts Computation algorithms

Each PC is a set of overdetermined set of equations, derived from the model, that can be solved. The algorithm to compute PCs finds recursively every combination of model equations that are strictly overdetermined using depth first search (i.e. we find any possible overdetermined system of equations for every equation). The algorithm proceeds finding one new equation potentially solving one of the remaining unknowns.

Algorithm 3 Step 1: Find every strictly overdetermined sub-system, PC, in M_H for each equation c in Σ_H

```
function FIND_EVERY_PC($M_H$) returns $SPCs$
    for all equation $c$ in $\Sigma_H$ do
        find_pc ($M_H \setminus$ {c}, {c}, $c_{unknowns}$, $SPCs$)
    end for
end function
```

A similar algorithm should analyze the set of equations and unknown variables in each *pc*, and find out if its associated set of equations have a globally consistent causal assignment, i.e. we can obtain a solution for the model and generate a simulation model, for instance. This is the basic algorithm that pays no attention to potential cyclical configurations – algebraic loops or loops containing differential equations–. These checks can be done later, and depend on the kind of equations in the model, the simulation language used, and the presence of appropriate equation solvers for loops (Pulido & Alonso, 2001; Pulido, Bregon, & Alonso-González, 2010).

Algorithm 4 Step 1.2: Find possible conflict, pc, from available equations in the model, RM: find a new equation for each remaining unknown variable, $unknowns$, in the pc

```
function FIND_PC($RM$, $pc$, $unknowns$, $SPCs$)
    if $unknowns == \{\}$ then
        if $pc$ is minimal w.r.t. $SPCs$ then
            remove every superset of $pc$ in $SPCs$
            insert $pc$ in $SPCs$
        end if
    else
        for all equation $c' \in RM$ do
            for all $y \in (c'_{unknowns} \cap unknowns)$ do
                find_pc($RM \setminus \{c'\}$, $pc \cup \{c'\}$, $unknowns \cup \{c'_{unknowns}\} \setminus \{y\}$, $SPCs$)
            end for
        end for
    end if
end function
```

Biographies

Anibal Bregon received his B.Sc., M.Sc., and Ph.D. degrees in Computer Science from the University of Valladolid, Spain, in 2005, 2007, and 2010, respectively. From September 2005 to June 2010, he was Graduate Research Assistant with the Intelligent Systems Group at the University of Valladolid, Spain. He has been visiting researcher at the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA; the Dept. of Electrical Engineering, Linkoping University, Linkoping, Sweden; and the Diagnostics and Prognostics Group, NASA Ames Research Center, Mountain View, CA, USA. Since September 2010, he has been Assistant Professor and Research Scientist at the Department of Computer Science from the University of Valladolid. Dr. Bregon is a member of the Prognostics and Health Management Society and the IEEE. His current research interests include model-based reasoning for diagnosis, prognostics, health-management, and distributed diagnosis and prognostics of complex physical systems.

Alexander Feldman is a postdoc at University College Cork and a visiting researcher at Ecole Polytechnique Federale de Lausanne (EPFL), Delft University of Technology and PARC (former Xerox PARC). He has obtained his Ph.D. (cum laude) in computer science/artificial intelligence and M.Sc. (cum laude) in parallel and distributed systems from the Delft University of Technology. He has published in leading conference proceedings and international journals covering topics in artificial intelligence, model-based diagnosis, and engineering. In cooperation with NASA Ames Research Center and PARC, Alexander Feldman has co-organized the International Diagnostic Competitions (DXC). Alexander Feldman's interest cover wide spectrum, including topics such as model-based diagnosis, automated problem solving, software and hardware design, design of diagnostic space applications, digital signal processing, and localization.

Belarmino Pulido received his degree, MsC degree, and PhD degree in Computer Science from the University of Valladolid, Valladolid, Spain, in 1992, 1995, and 2001 respectively. In 1994 he joined the Departamento de Informática in the University of Valladolid, where he is Associate Professor since 2002. His main research interests are Model-based reasoning and Knowledge-based reasoning, and their application to Supervision and Diagnosis. He has worked in different national and European funded projects related to Supervision and Diagnosis of complex industrial systems. He is member of different professional associations (IEEE since 2000, ACM since 2003, CAEPIA -Spanish section of ECCAI- since 1997). Moreover, he is the coordinator of the Spanish Network on Supervision and Diagnosis of Complex Systems since 2005.

Gregory Provan is a Professor at the Computer Science Department at University College Cork (UCC), in Cork, Ireland. He received his BSE from Princeton University, USA, his MSc from Stanford University, USA, and his DPhil from University of Oxford, UK. He is currently Director of the Complex Systems Lab at UCC. Prior to joining UCC he was on faculty at the University of Pennsylvania, USA, and spent over 10 years in industrial research. His interests are in complex systems (design, analysis and control), diagnostics, algorithm design, and bioinformatics. His current research focuses on modelling and analysis of sustainable energy systems, model-based diagnostics of a variety of complex systems, and design of methods for efficiently embedding code in such systems. Prof. Provan is the coauthor of over 100 refereed articles, and has been Principal Investigator in a variety of grants and research contracts.

Carlos Alonso-González got his degree in Sciences in 1985, and his PhD in Physics in 1990, from the University of Valladolid, Valladolid, Spain. Currently he is Associate Professor with the Department of Computer Science from the University of Valladolid. He has being working in different national funded projects related to Supervision and Diagnosis of continuous industrial environments. His main research interests are Knowledge-based systems for Supervision and Diagnosis of dynamic systems, model-based diagnosis, Knowledge Engineering and Machine Learning.

Efficient Generation of Minimal Dynamic Bayesian Networks for Hybrid Systems Fault Diagnosis using Hybrid Possible Conflicts

Belarmino Pulido, Noemi Moya, Carlos J. Alonso-González, Anibal Bregon

Departamento de Informática, Universidad de Valladolid, Valladolid, 47011, Spain
{belar,noemi,calonso,anibal}@infor.uva.es

ABSTRACT

Hybrid systems diagnosis requires different sets of equations for each operation mode in order to estimate the continuous system behaviour. In this work we rely upon Hybrid Possible Conflicts (HPCs), which are an extension of Possible Conflicts (PCs) for hybrid systems, that introduce the information about potential system modes as control specifications that activate/deactivate different sets of equations. We also introduce the concept of Hybrid Minimal Evaluation Models (H-MEMs) to represent the set of globally consistent causal assignments in an HPC for any potential mode.

H-MEMs can be explored for a specific operation mode, and its computational model automatically generated. In this work, the selected computational models are minimal Dynamic Bayesian Networks (DBNs). Since DBNs can be directly generated from PCs, and can be used for fault detection and isolation, we propose to efficiently generate Minimal DBNs models on-line using the H-MEM structure. By introducing fault parameters in the DBN model, we can also perform fault identification, providing an unifying framework for fault diagnosis, under single fault assumption. We test the approach in a simulation four-tank system.

1. INTRODUCTION

Dynamic systems with hybrid behaviour are present in almost every field in our society. Fault diagnosis for these systems is of capital importance to prevent malfunctions or breakdowns, and to increase the security and the quality of the final products. However, it is difficult to provide accurate and timely online fault diagnosis. because their behaviour is made up of continuous behaviour commanded by discrete events.

For the last 15 years two research communities: the Control Theory, known as the FDI[1] approach, and Artificial Intelligence, known as the DX approach, have worked on hybrid systems modelling and diagnosis (Cocquempot, El Mezyani, & Staroswiecki, 2004; Hofbaur & Williams, 2004; Narasimhan & Biswas, 2007; Narasimhan & Brownston, 2007). Current research activities are focused on two main issues: full or approximate estimation of the set of possible states, and tracking of nominal and faulty system behaviour (Rienmüller, Bayoudh, Hofbaur, & Travé-Massuyès, 2009). To tackle the first issue, different kinds of automata have been used to model the complete set of modes, and transitions between them, which introduces the need to enumerate all the set of modes (states) and transitions, and to track the entire set of consistent modes. Both issues do not scale well for complex systems. To avoid the pre-enumeration of modes, we have followed the proposal by Narasimhan et al. (Narasimhan & Biswas, 2007), which uses Hybrid Bond-Graphs (HBGs) to model the whole system, and depending on the value of the switching junctions, used to model the hybrid behaviour, it is able to generate on-line new models for tracking the new system mode. That work has been recently extended to efficiently generate simulation models using model block diagrams based on HBGs properties (Roychoudhury, Daigle, Biswas, & Koutsoukos, 2011), and to efficiently generate state observers (Podgursky, B., & Koutsoukos, 2010).

Bregon et al. (Bregon, Alonso, Biswas, Pulido, & Moya, 2012) introduced Hybrid Possible Conflicts (HPCs) as an extension of Possible Conflicts (Pulido & Alonso-González, 2004) using HBGs (Narasimhan & Biswas, 2007), and Block Diagrams (Roychoudhury et al., 2011). HPCs can track hybrid systems behaviour, efficiently changing on-line for each mode the PC simulation model, and performing diagnosis without pre-enumerating the set of modes in the system. But HPCs do not provided a unified diagnosis framework using one technique.

In this work we propose to use minimal Dynamic Bayesian Networks, DBNs, derived from HPCs as a unique modelling framework for hybrid systems fault detection, isolation, and identification, together with new algorithms to automatically generate on-line the DBN computational model, and then using that model to track the system. To automatically generate

Belarmino Pulido et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] Acronym for Fault Detection and Isolation.

on-line the computational model we introduce the concept of H-MEM, that provides the potential set of equations that can appear in any HPC. The approach has been tested on a four-tank laboratory plant with satisfactory results.

The rest of the paper is organized as follows. Section 2 presents the case study used along the paper and introduces HBG modelling technique. Section 3 summarizes the HPCs background. Section 4 describes our new diagnosis framework with DBNs computed from HBGs. Section 5 presents some results obtained applying our proposal to the case study. Section 6 describes some related work and Section 7 draws some conclusions and future work.

2. Case Study

The hybrid four-tank system in Figure 1 will be used along the paper. It has one input flow to tanks 1 and 3, that can be sent to both tanks or only to one of them. Tanks 2 and 4 are connected to tanks 1 and 3, respectively, through a pipe placed at a distance *h* above their bases. Sources of discrete behaviour are: commanded valves at the input of tanks 1 and 3, and the pipes connecting tanks 1 and 2, and between tanks 3 and 4, at a given height $h_i > 0$. There are four measurements in the system: pressure sensors at the bottom of every tank.

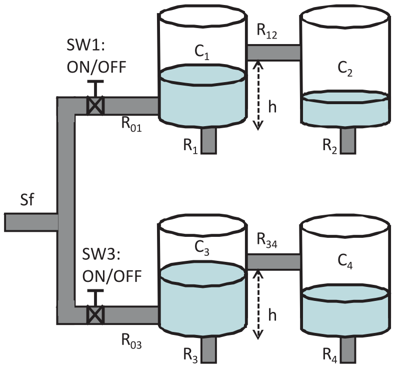


Figure 1. Schematics of the four-tank system

We used HBGs, an extension of Bond-Graphs (BGs) (Karnopp, Margolis, & Rosenberg, 2006), to model our system. BGs is a domain-independent energy-based topological modelling language for physical systems. BGs rely upon primitive elements: storage (capacitances, C, and inductances, I), dissipative (resistors, R) and energy transformation (transformers, TF, and gyrators, GY) elements. There are also effort and flow sources (Se and Sf) to define interactions between the system and the environment. Elements in a BG are connected by 0 or 1 junctions (representing ideal parallel or series connections between components). Each bond has two associated variables (effort and flow). The rate of energy is defined as $effort \times flow$ for each bond. The SCAP algorithm (Karnopp et al., 2006) is used to assign causality automatically to the BG.

Related to the primitive elements, sources and junctions there is a set of well-established equations relating flow and effort variables. The exact expression of each equation depends on the assigned causality. For instance, for a resistance, R, element with effort and flow variables, e_1, f_1, the corresponding equation would be $e_1 = R \times f_1$ or $f_1 = R \times e_1$. Energy storage elements, such as a capacitor C, provide the following equation $e_2 = \frac{1}{C} \int f_2 dt$, for variables effort and flow, e_2, f_2, in integral causality. Finally, 0-junctions and 1-junctions model ideal common effort or common flow connections, where efforts (equivalently flows) are all equal ($e_1 = e_2 = e_3$), while sum of flows (correspondingly efforts) must equal zero ($f_1 - f_2 + f_3 = 0$). Additionally, there are effort and flow detectors, De and D_f respectively, that provide measurements of system magnitudes.

To model hybrid systems we need to use some kind of connections which allow changes in their state. HBGs (Narasimhan & Biswas, 2007) extend BGs by including those connections. If a switching junction is set to ON, it behaves as a regular junction. When it changes to OFF, all bonds incident on the junction are deactivated forcing 0 flow (or effort) for 1 (or 0) junctions. A finite state machine control specification (CSPEC) implements those junctions. Transitions between the CSPEC states can be triggered by endogenous or exogenous variables, called guards. CSPECs capture controlled and autonomous changes as described in (Roychoudhury et al., 2011).

Figure 2 shows the HBG model of the four tanks system, where there are four measurements for diagnosis: the level of the four tanks by means of pressure sensors, p_i related to capacitances C_i, that are represented as effort detectors $De : p_i,\ i = 1..4$ in the BG.

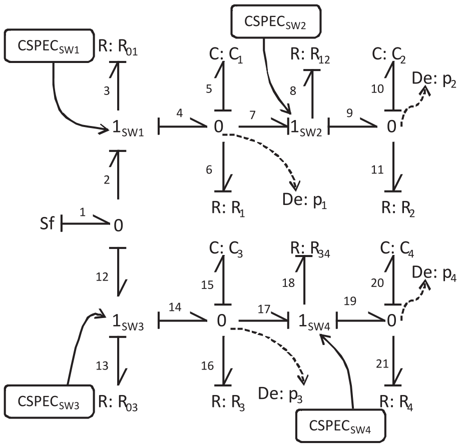


Figure 2. Bond graph model of the plant.

Regarding the hybrid behaviour, it has four switching junctions: SW_1, SW_2, SW_3 and SW_4. SW_1 and SW_3 are controlled *ON/OFF* transitions, while SW_2 and SW_4 are autonomous transitions related to water level of tanks 1 and 3 surpassing the height h and overflowing to tanks 2, and 4 respectively. Both kind of transitions are represented using a finite state machine. Figure 3 shows: a) the automaton associated with switching SW_1 and b) the automaton representing the autonomous transition in SW_2 (since the system is symmetric, automata for SW_3 and SW_4 will be equivalent).

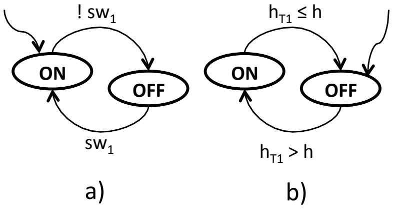


Figure 3. a) Automaton associated with the *ON/OFF* switching junction SW_1; b) Automaton representing the autonomous transition in SW_2.

3. Hybrid PCs and minimal DBNs background

3.1. Hybrid Possible Conflicts (HPCs)

The Possible Conflict, PC, approach is a dependency-compilation technique from the DX community (Pulido & Alonso-González, 2004), that have been successfully used for system model decomposition in consistency-based diagnosis of continuous systems. PCs define minimal structurally overdetermined subsets of equations with sufficient analytical redundancy to generate fault hypotheses from observed measurement deviations. In the original approach, only structural and causal information from the system model is used. PCs are computed using a hypergraph abstracting the structural model of the system. Recently, we have proposed an extension that allows to compute PCs directly from bond graph models (Bregon, Biswas, & Pulido, 2012).

The PC approach has been recently extended to cope with hybrid system dynamics, using Hybrid Bond-Graphs (Roychoudhury et al., 2011; Narasimhan & Biswas, 2007) as the modelling approach. The extension is called Hybrid Possible Conflicts (Bregon, Alonso, et al., 2012). Main advantage of HBG modelling technique is that pre-enumeration of the modes in the system is not necessary. However, its main concern when applied to fault diagnosis of hybrid systems (Narasimhan & Biswas, 2007) is related to the task of causality reassignment for the entire bond graph model, because during this causality reassignment process, the diagnosis system needs to stop tracking the behaviour of the system, making it sensitive to miss faults that occur during (or immediately after) such reassignment process. However, recent proposals for fast causality reassignment in HBGs can be used to speed up this process for efficient on-line simulation (Roychoudhury et al., 2011). Typically, changes in causality do not propagate within the model, or only a small part of the model causality needs to be reassigned. Moreover, when causality needs to be reassigned, changes will be typically local to the hybrid junction. HPCs incorporate the proposal by Roychoudhury et al. (Roychoudhury et al., 2011) to generate new causality assignments for HPCs once a mode change is observed. Currently, our main assumption is that we are able to track the current system mode.

For the case study we have found four HPCs. Each one of them estimates one of the measured variables (p_1, p_2, p_3, or p_4). Figure 4 shows the HBG fragments of these four HPCs.

In this example, we first computed HPCs assuming that all switching junctions are set to *ON*, but when any of these junctions is switched to *OFF*, causality in the system needs to be reassigned. Even though causality may change, the HPC generation process does not need to be restarted again (Bregon, Alonso, et al., 2012).

There are two basic possibilities for the existing HPCs depending on whether the change in the switching junction induces a change in causality or not. First, the change in the switching junction induces a change in causality which affects the HPC. A new causality will be assigned to the HPC and it will be updated. If there is not a valid causal assignment, the HPC will disappear. Second, as a result of the change in the switching junction there is no change in causality. In this case, either a PC can remain the same, or a part of the PC can be affected by the switch and disappear or the whole PC can disappear (the discrepancy node disappears).

HBGs main advantage is that the complete set of modes do not need to be known or enumerated in advance. However, many times there is no such HBG model available. In this work we propose to compute HPCs for a generic set of ODEs, given that some of them are only valid under given system configurations, thus it is needed to extend original algorithms to compute PCS by introducing the information about discrete dynamics. To show our approach we have used as system model the set of equations that can be derived from an HBG model, as explained above in order to ease the comparison of this approach with the previous one. But in general, any set of ODEs can be our system model because for computing our HPC models we work mainly at the structural and causal level.

Figure 5 represent one possible MEM for PC_2 in Figure 4. The MEM provides a computational model that can be implemented as a simulation or state-observer model (Pulido, Bregon, & Alonso-González, 2010). In this work we propose to implement our PCs as a set of Dynamic Bayesian Networks, providing a framework capable to perform not only fault detection and isolation but fault identification using the same computational model.

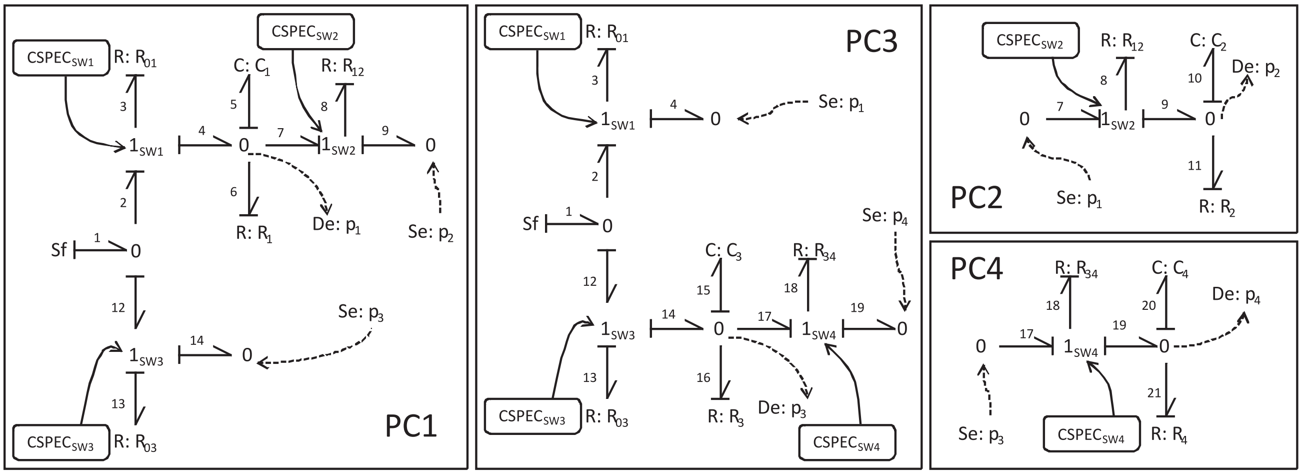


Figure 4. Hybrid Bond graphs of the four HPCs found for the four-tank system.

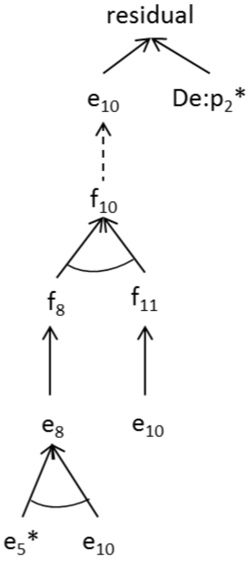


Figure 5. MEM for PC_2 subsystem in Figure 4. The effort and flow variables in the graph correspond to pressures and flows in PC_2 in Figure 4.

3.2. Minimal Dynamic Bayesian Networks(DBNs)

Dynamic Bayesian Networks (DBNs) are a probabilistic temporal model representation of a dynamic system. Basically, a DBN is a two slices Bayes Network (BN). Assuming that the system is time invariant and a First Order Markov process, two static and identical BN connected by inter slice arcs are enough to model the system (Murphy, 2002). Inter slice arcs model system dynamics. Intra slice arcs model instantaneous (algebraic) relations.

The system variables (X, Z, U, Y) represented in a DBN are the inputs (U), the state variables (X), the observed or measured variables (Y) and, in some cases, other hidden variables (Z) . Once we have the nodes, we need to define the arcs and the parameters in the model, the state transition model (graphically represented by the inter slice arcs) and the observational model (represented by intra slice arcs).

Figure 6 represent the DBN for MEM2 in Figure 5. Blue arrows represent the inter slice arcs modelling system dynamics. Orange arrows represent the intra slice or instantaneous relations among system variables. Alonso-Gonzalez et al (Alonso-Gonzalez, Moya, & Biswas, 2011) provided the method to automatically transform a MEM from a PC to a DBN model. Following the method we obtain the DBN model in Figure 6.

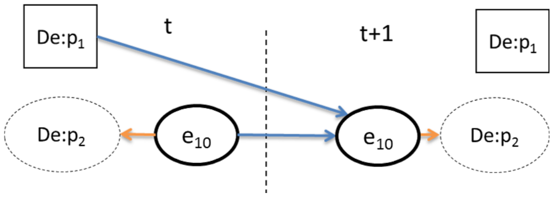


Figure 6. DBN for the PC_2 subsystem in Figure 5. Input node is measured pressure $e5$ = *De:p1*, state variable is the pressure in tank 2, $e10$, and the output node is the measured pressure in tank 2 *De:p2*.

Exact inference in DBNs is not computationally tractable. Hence, Monte Carlo simulation methods are used for approximate inference, particularly Particle Filter algorithm (Koller & Lerner, 2001). The unknown continuous stochastic distribution of the state is approximated by a discrete distribution obtained by weighted samples. After propagation of the state, the weights are updated with current observations. In this work, we assume a Gaussian distribution.

DBNs can be used along all the stages in the diagnosis process. They provide a unified framework for fault diagnosis. DBNs can be generated from a PC derived from a BG model (Alonso-Gonzalez et al., 2011) and have been successfully applied for fault diagnosis of continuous systems (Roychoudhury, Biswas, & Koutsoukos, 2008; Alonso-Gonzalez et al., 2011) .

In this work, we propose to integrate DBNs to monitor the continuous behaviour of the system, and to use HPCs to generate different DBNs for each new mode. We propose to build a different DBN for each mode, instead of using a hybrid DBN able to track the complete set of modes related with the HPC.

4. Efficiently computing Hybrid PCs

In (Pulido & Alonso-González, 2004) PCs were computed for a unique mode. The computation was made in two steps: first, we obtained the set of minimally overdetermined sets of equations, which are called Minimal Evaluation Chains, MECs –equivalent to minimal ARRs (Analytical Redundancy Relations) or MSO (Minimal Structurally Overdetermined) sets of equations (Armengol et al., 2009)–. Second, introducing causal information in the model[2], for each MEC we obtained the complete set of globally consistent causal assignments, each one called Minimal Evaluation Model or MEM. Each MEM provides the computational model required to build a simulation or a state-observer model (Pulido et al., 2010).

In previous works we have demonstrated that the structural and causal models can be automatically obtained from Bond-Graph models, deriving a Temporal Causal Graph, TCG, that represent a consistent causal assignment for the system in one mode. And we can compute the set of Possible Conflicts from the TCG (Bregon, Pulido, Biswas, & Koutsoukos, 2009). But in this section we propose to extend the approach to any causal model (we always can start from a system model made up of a set of ODEs, and then to abstract the structural and causal information in them to generate a causal model where only the presence of measured and unknown variables in an equation is relevant). In (Bregon, Alonso, et al., 2012) we proposed how to obtain HPCs from HBGs, using HSCAP (Roychoudhury et al., 2011) to avoid computing a new causal assignment in the HBG whenever there is a mode change.

In this work we propose to compute the HPCs directly from a set of labelled equations, obtained as an abstraction of the set of ODEs which is our model. In order to efficiently generate computational models as minimal DBN factors, we need to extend the algorithms computing HPCs in two ways. First, we need to compute the set of HPCs, i.e. PCs with labelled equations related to discrete dynamics. Second, we need to automatically and efficiently build the DBN behavioural models from the computational model provided by a MEM.

4.1. Inclusion of constraints to represent discrete dynamics

To fulfill the first requirement we first introduce information about discrete dynamics in the model, and later on we modify the original algorithms to compute PCs. In this work we assume that each equation in the system model is valid in a set of configurations, and these configurations can be characterized as constraints: one constraint is a well-formed formula, WFF, in propositional logic. The propositions in the WFF will represent the control specifications related to the switching junction automata (as shown in Figure 3) because they will have only boolean values related to the ON/OFF state of the switching junction.

These modifications can be summarized as follows:

- first, we add the information about constraints in the equations as WFF. Each switching junction introduces an atomic proposition, whose values *true* or *false* will be function of the switching junction control specifications being *ON* or *OFF*.
- Second, the automaton representing the switching junction is explicitly modelled as a set of constraints in adjacent equations, forcing different causal assignments.

For 1-junctions (and 0-junctions have a corresponding dual version):

- when the switch is *ON*: flows must be equal $f_i = f_j = f_k$, and effort variables must sum up to 0: $\sum e_l = 0$
- when the switch is *OFF*: there is no effort related to the junction, and each flow is transformed into a zero flow source: $Sf : f_i = 0$, $Sf : f_j = 0$, and $Sf : f_k = 0$.

As an example, the behaviour of switching junction SW_1 in Figure 2 will provide the set of equations in Table **??**, and their corresponding evaluation forms[3].

Table 1. Equations with no causality, their associated evaluation form (with some possible causality assignments), and constraints for the equations in the four tank system model.

Equation	Evaluation form	Constraint
$(ec_3 : e_2, e_3, e_4)$	$ec_{3_1} : e_3 := e_2 - e_4$	sw_1
$(ec_2 : f_2, f_3, f_4)$	$ec_{2_1} : f_2 := f_3; f_2 := f_4; f_3 := f_4$	sw_1
$(ec_4 : f_3, e_3)$	$ec_{4_1} : f_3 := e_3/R01$	sw_1
$(ec_4 : f_3, e_3)$	$ec_{4_2} : f_3 := 0$	$\neg sw_1$

It must be noticed that there is no valid evaluation form for $(ec_3 : \;\; e_2 \;\, e_3 \;\, e_4)$ when $\neg sw_1$ is $true$. Then, even if ec_3

[2] We made difference between static and dynamic relations, called differential equations. Different causal assignments for differential equations provide integral or derivative approaches for behaviour estimation.

[3] HBGs provide a systematic way to derive the ODEs. But we can start the process from any kind of modelling language.

and ec_4 share the unknown variable e_3, they can only be used together when sw_1 is evaluated to $true$. Those equations with no contraints represent equations that are valid in any working mode.

Using this new system model, we only need to introduce slight modifications in the algorithms developed to compute PCs as described in (Pulido & Alonso-González, 2004); we just need to include the information about the constraints modelling discrete behaviour while building the set of MEMs for a given HPC.

The algorithm for computing MECs is the same. Only those MECs with a valid switching configuration will provide a valid MEM, i.e., those MECs will be PCs. We consider a switching configuration as valid if its associated WWF is satisfiable, that is, there is at least one configuration where it is consistent.

Algorithms for computing PCs can be easily extended to introduce constraints related to mode changes: when building each MEM, in each step we try to justify or remove one unknown variable using known values. Now, we impose the additional requirement that the set of constraints in the MEM and the constraint in the new equation, if any, provide a satisfiable formula (i.e., it contains no contradiction). For any expression or equation where there is more than one constraint, we must explore in parallel every potential solution. Hence, for a given MEC we can obtain a collection of MEMs, and each MEM will be valid in a limited set of operation modes, determined by satisfiable WWFs.

For instance, we can use ec_{3_1} to estimate the value of e_3, then use e_3 and equation ec_{4_1} to estimate the value of f_3 under the constraint sw_1, i.e. when switching junction SW_1 is set to *ON*. However, when switching junction SW_1 is set to *OFF*, $\neg sw_1$ is $true$ and we can not use both equations ec_{3_1} and ec_{4_1}. We can only use ec_{4_2} that fixes the value of f_3 to zero.

Hence, each extended MEM represent now a global consistent causal assignment for the equations in a HPC, together with a WFF in propositional logic made up of the conjunction of every constraint. For instance, both $sw_1 \wedge sw_2 \wedge sw_3$ and $sw_1 \wedge \neg sw_2 \wedge sw_3$ are satisfiable formula, but $sw_1 \wedge \neg sw_1$ is not. We term *label* of a MEM to its WFF.

The complete set of MEMs plus their associated constraints represent all the consistent causal assignments for the equations in a HPC, i.e. they represent the evaluation form for all the possible behavioural models in a HPC. Each MEM in a HPC will have one discrepancy node[4], but will have different sets of equations depending of the current mode on the system. For any given MEM, the hyperarcs represent the equations used to compute the head of the hyperarc using the variables in the tail of the hyperarc. Leaf nodes in the hypergraph are either measured variables or previously estimated unknown variables, i.e. potential cyclical configurations. In the hypergraph differential constraints are represented as dashed hyperarcs, and they do not introduce loops.

We call that complete collection of extended MEMs for any HPC, Hybrid MEM (H-MEM). In Figure 7 we show the H-MEM for HPC_1, in our case study. Labels in the right hand side of the hyperarcs as $\{sw_j\}$ or $\{\neg sw_j\}$ represent the constraints related to the original switching junction CSPECs. Remainder labels represent either the name of the equation, ec_k, or the faulty parameter related with the equation: either R_x, or C_x for faults in resistance or capacitance elements in the original BG model. This H-MEM represents the most complex configuration in our case study, since the models for HPC_1 in different modes require different causality assignments. For a given mode, we will use only those paths from the leaf nodes to the discrepancy node (to compute a residual) whose constraints are consistent.

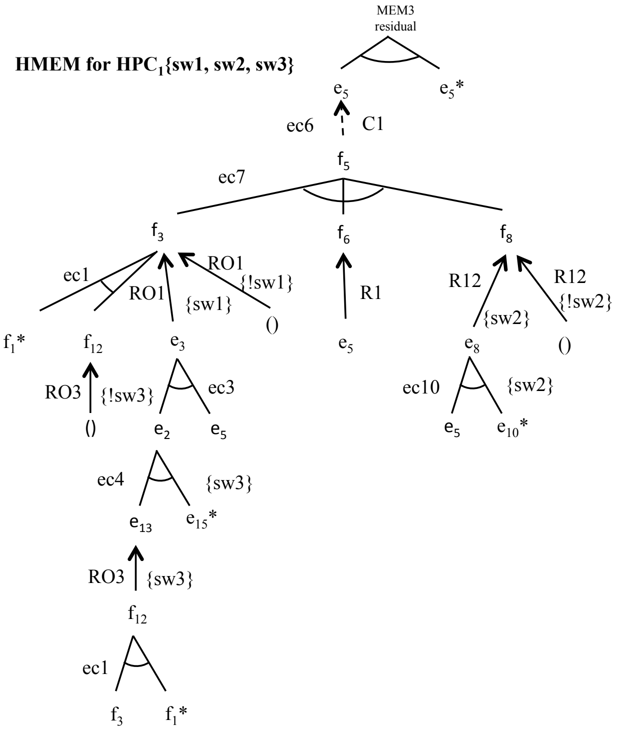


Figure 7. Hybrid Minimal Evaluation Model for HPC_1. We use dashed arcs for differential equations, and solid arcs for instantaneous equations. Observed variables are marked with an asterisk. Different arcs entering in a node represent different paths.

We do not compute the complete H-MEM. We only need to know the set of constraints contained in the HPC, because each mode is defined by a WFF involving just the atomic propositions for switching-junctions involved in the HPC. Once we know the For instance, the H-MEM for HPC_1 in

[4] The only variable estimated and measured, which can be the origin of a real conflict. The discrepancy node is equivalent to a residual in FDI terminology.

Figure 7[5] has consistent MEMs in the following configurations, among others:

$$sw_1 \wedge sw_2 \wedge sw_3$$
$$\neg sw_1 \wedge sw_2 \wedge sw_3$$
$$sw_1 \wedge \neg sw_2 \wedge sw_3$$

It must be noticed that in engineering systems, not every possible configuration is feasible for security reasons. For instance, in our system is not possible to have both switches, sw_1 and sw_3 both *off* at the same time. Hence, those models will never be generated.

In fact, we do not need to know all these configurations. Once a new mode is suspected, if one H-MEM contains constraints related to that mode, we build its actual MEM, just using the expressions of the equations that are valid under the current mode, depending on the actual values of the switches. We depth-first traverse, dft, the hypergraph from the discrepancy node to the leaf nodes:

```
FUNCTION  mem = create-MEM(hmen: H-MEM, m: mode)
begin
  dft(hmem.discrepancy_node, hmem, m, mem);
 return mem;
end FUNCTION

FUNCTION dft (c: node, hmen: H-MEM, m: mode, mem: MEM)
begin
insert(c) in mem; // c is current node
for y = each node in hmem | (c,y) is an edge in hmem
 AND (c,y).label is compatible with m
do
  if y was not visited yet then
    insert(c, y) in mem;
    insert(c,y).label in mem.label;
    dft(y, hmem, m, mem);
  end
end
end FUNCTION
```

The model described by that MEM can be implemented in many different ways. In this work we opted for DBNs implemented as Particle Filters.

4.2. From H-MEMs to minimal DBNs

In (Alonso-Gonzalez et al., 2011) it was described how to derive the transition and the observational model of a DBN-PC factor from a MEM. The transition model estimates the next state(s) value(s) in the DBN from the inputs, and the current state(s) value(s); the observational model computes the value of the system output (only one output in a MEM because it is minimal) given the state(s). In that work the model was built manually. In this section we explain how to efficiently and automatically derive the model for the DBN-PC factor from the MEM; that model will be later implemented as a particle filter in Matlab(©MathWorks).

While building a MEM it is trivial to identify the discrepancy node, input and output variables (measurements), and state variables: first, we are using integral causality to build the DBN-PC factor, then it is straightforward to determine the state variables that will be the nodes computed by the set of differential constraints, i.e. those modeling dynamics. In our case study, we have only one state variable for each HPC, and they correspond to the capacitor elements in the original Hybrid Bond-Graph: e_5, e_{10}, e_{15}, and e_{20}. Second, in the HBG framework, only measured outputs can be the origin of a discrepancy. Hence, there will be only one output variable in each H-MEM, and that will be the discrepancy node.

Once we identify those elements, we implement the proposal by Alonso-González et al. (Alonso-Gonzalez et al., 2011). We assume that the analytical expression for each equation in the MEM is known. Then, we just need to find the transition model and the observational model for the DBN factor.

Obtaining the observational model is simple: we depth-first search in the MEM from the discrepancy node to state variables and inputs, just using instantaneous constraints. That sequence of equations in reverse order is the analytical expression of the observational model.

```
FUNCTION get_observational_model (mem: MEM,
om: observational_model)
begin
   return dfs(MEM.discrepancy_node, mem, om);
end FUNCTION

FUNCTION om = dfs (n: node, mem: MEM,
om: observational_model)
begin
for y = each node in mem | (n,y) is an edge in mem
do
  add (n,y) at the beginning of om;
  if (n,y) is a differential edge then
    mark n as mem.observed-state-variable;
  else
    dfs (y, mem, om);
  end
end
end FUNCTION
```

The transition model can be obtained searching depth first the MEM from the state variables to state variables and inputs, following the requirements in (Alonso-Gonzalez et al., 2011). The analytical model is obtained from the transcription of that sequence of equations in reverse order.

```
FUNCTION get_transition_model (mem: MEM,
  tm: transition_model, om: observational_model)
begin
  for st = each state-variable
   in mem.observed-state-variable
  do
    dfs2 (st, mem, tm, om);
  end
end FUNCTION

function dfs2 (st: node, mem: MEM,
tm: transition_model, om: observational_model)
begin
 for y= each node | (n, y) is an edge in mem
  AND (n,y) is not an edge in om
```

[5] To make the H-MEM more readable, we have collapsed every effort or flow variable of the same junction with a unique index. For instance e4=e5=e6=e7 or f2=f3=f4

```
  do
    if (st,y) has not yet been visited then
      mark (st,y) as visited;
      insert (st,y) at the beginning of tm;
      tm= dfs2 (y, mem, tm, om);
    end
  end
end FUNCTION
```

Different proposals have been made to improve the algorithms computing the set of PCs. One of them is able to find any cyclical configuration (Pulido & Alonso, 2001). Those configurations can introduce algebraic loops in the model. Those loops containing differential constraints are no longer loops; in fact, they represent the integration step in the simulation model. Those loops containing just algebraic loops can be solved if there is a direct path from observed variables to an unknown variable in the loop. Otherwise, we need to create a subset of equations that need to be solved using a numeric solver (Pulido et al., 2010). In this case, there could be efficiency problems for the DBN-PC factor. All these analysis must be done before we build either the observational or the transition model.

Regarding the efficiency of this proposal, there will be always a trade-off in terms of space and computation time. Depending on the system under study, different heuristics can be applied to customize the algorithms performance. For small systems with a limited number of modes, most of the computation can be done off-line and cached to speed up the on-line code generation. In the H-MEM there will be a number of equations that will always appear in the system because they are causality independent. These could be also pre-compiled, since they will always provide the same analytical expression in any MEM in any mode.

5. Results

The four-tank hybrid system in Figure 1 has been used to show the applicability of our proposal.

Simulated data has been generated with 5% level of noise, during 1000 s with a sample period of 0.1. We run several experiments with different mode configurations and different faults, varying the size and time of fault occurrence. Results for all these situations were equivalent to the example presented next.

5.1. Tracking and fault isolation results

For the four tank system we computed the set of four HPCs and their corresponding H-MEMs, using the new algorithms. Since our models were provided by the HBG of the whole system when every switching-junction was set to ON, we obtained the same results.

Figure 8 shows the results obtained for one of the experiments run. First row (Figure 8) compares the three measurements and its estimation by the DBN-PC, while second row shows the residual obtained for each DBN-PC. DBN from HPC4 has not been included in the figure as this PC is always deactivated during the experiment. The results of the experiments have 10000 time stamps. The graphs built with those signals were difficult to read due to its size. Some time intervals during stationary state have been ommited to avoid that problem. Because of that, the time stamps that are mentioned below will not match with the time stamps in Figure 8, but the comments about the real time stamps are correct.

Initially, water tanks are empty, and start to fill in at constant rate. Hence, the initial configuration of the system is SW_1 and SW_3 set to *ON*, and SW_2 and SW_4 set to *OFF*. Tanks 1 and 3 start to fill in, and approximately at instant 500 sampling periods both tanks reach stationary state. At this time, the level in tank 1, h_{T1}, and the level in tank 3, h_{T3}, are lower than the height of the connecting pipes, h, and consequently, there is no flow through the connecting pipes.

At instant 2000 sampling steps, controlled junction SW_3 is set to *OFF*, so the system mode changes. Simultaneously, $HPC1$ and $HPC3$, which contain constraints related to SW_3, must change their models to accommodate the new operation mode. It is necessary to reassign causality in our H-MEMs. Once the new computational expression for the HPCs have been generated, the corresponding DBNs are built. As shown in Figure 8, DBN-PC1 and DBN-PC3 are able to correctly estimate the level of tank 1 and 3, respectivley, immediately after the mode change. Regarding $HPC2$ and $HPC4$, since both HPCs do not contain constraints related to the switching junction SW_3, none of them is affected by the mode change so their DBNs do not need to be generated.

SW_3 has been set to *OFF*, so the level of tank 3 decreases until it becomes zero, while the level of tank 1 increases. At instant 2100 sampling periods, the level of tank 1 reaches the height of the connecting pipe between tanks 1 and 2. At this point, the equations related to autonomous transition in SW_2 are set to *ON* and water begins to fill in tank 2. HPC_1 and HPC_2 are affected by this mode change. In both cases, the models of their H-MEMs are updated and the DBNs are generated quickly. Both of them are able to correctly estimate the measurements for the new mode.

At instant 7000 sampling periods a 20$\%$ leak in tank 1 occurs. As a consequence, the level of tank 1 decreases, while the estimation of $HPC1$ does not. Hence, residual of $HPC1$, which is the only one containing C_1 as a fault candidate, activates, triggering the fault isolation procedure. Regarding $HPC2$, since the level of tank 1 decreases due to the fault, at instant 7050 sampling periods the autonomous junction SW_2 transitions again to *OFF* mode, and $HPC2$ changes mode again. The H-MEM for $HPC2$ is updated immediately and DBN-PC2 is built; it is able to correctly estimate the level of tank 2 for the new mode.

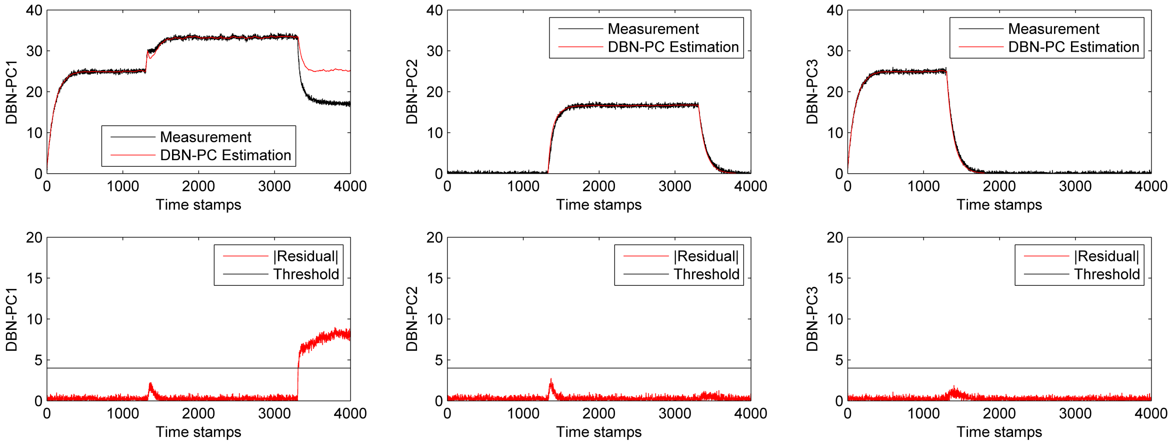


Figure 8. Experiment for the four-tank system.

For the case study, the average time for updating the H-MEM to generate the current MEM, and to generate the analytical model for the DBN is less than 1 ms. The algorithms were built in Java and they were run in a Intel Core i3 processor with RAM of 4GB.

5.2. Fault identification results

A 20% leak in tank 1, which is related to parameter C_1 in the models, was introduced at instant 7000 sampling periods and 9 sampling periods later (0.9 seconds) DBN for HPC1, DBN-PC1, detects a fault. According to the Fault Signature Matrix, FSM, in Table 2, the set of fault candidates is $\{C_1, R_1, R_01, R_{03}, R_{12}\}$. The FSM describes the relation between the set of faulty parameters in the model and the set of HPCs.

Table 2. Fault Signature Matrix for the four tank system. The parameters in row are directly obtained from the BG model, and their corresponding constituent equations.

	$HPC1$	$HPC2$	$HPC3$	$HPC4$
C_1	1			
C_2		1		
C_3			1	
C_4				1
R_{01}	1		1	
R_{03}	1		1	
R_1	1			
R_2		1		
R_3			1	
R_4				1
R_{12}	1	1		
R_{34}			1	1

DBN-PC1 can be extended with a node for the faulty parameter which needs to be identified as explained in (Alonso-Gonzalez et al., 2011). In this scenario, five DBNs were built, one for each fault candidate. Figure 9 shows the results obtained using the DBN-PC1 to estimate C_1. The DBN is able to track the system behavior and to obtain an estimation for the parameter quickly converging to a 19.3% fault in C_1. DBN-PCs to estimate the remaining faults were not able to converge. Hence, the candidates were discarded.

6. Related Work

In our approach we do not need to enumerate the complete set of modes, as required by other works using parameterized ARRs (Cocquempot et al., 2004; Bayoudh, Travé-Massuyès, & Olivé, 2009), or using pure discrete models. We just need to provide the constraints for the equations. Later on, for each HPC, its model will be generated for a specific working mode. However, in the general case, we would need to compute the set of HPCs at least for the configuration where every switch is on. But we don't need to instantiate them. We just need to check what HPC has a valid causal assignment for the current operation mode. In that set we build a superset of parameterized ARRs. Our main assumption to do so is that every structural model is a subset of the structural model where every switch is ON.

Using DBNs derived from HPCs for fault detection, isolation and identification avoids using several techniques as in (Narasimhan & Biswas, 2007; Rienmüller et al., 2009). But the mode must be observed, thus requiring an hybrid state estimation (Hofbaur & Williams, 2004; Koutsoukos, Kurien, & Zhao, 2003).

Using minimal DBN-PCs as a unified model, we do not provide general solutions for hybrid systems diagnosis such as HyDe (Narasimhan & Brownston, 2007). But we combine continuous estimation with discrete changes, and we do not restrict our solution to pure discrete systems as in HyDe or probabilistic approaches. Moreover, interleaving different DBN continuous models we avoid the usage of hybrid DBNs.

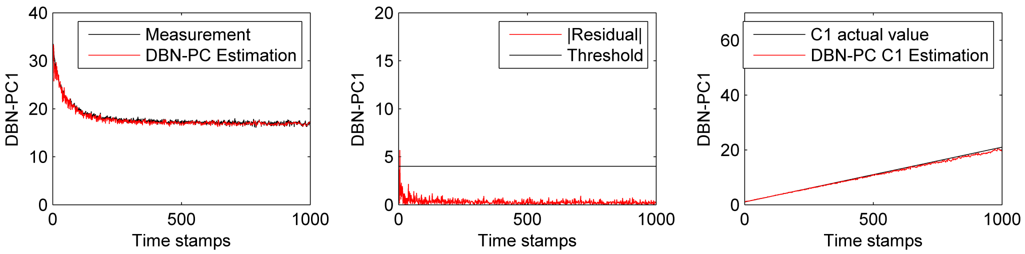


Figure 9. Identifiying the fault in parameter C1.

Our work is closely related to efficient generation of simulation (Roychoudhury et al., 2011) and state observer models (Podgursky et al., 2010) in TRANSCEND (Narasimhan & Biswas, 2007), because we assume that changes in model causality will be mostly local, but we do not rely upon the Hybrid SCAP algorithm to generate a valid HBG model for the entire system. Instead, we need to know every feasible causal assignment in the system description, and perform on-line local search in the H-MEMs. Finally, (Bregon, Alonso, et al., 2012) proposed to obtain the set of HPCs from block diagrams derived from HBGs (Roychoudhury et al., 2011). We improve that proposal by directly generating DBN computational models instead of simulation models, thus improving fault detection capabilities, and the process can be performed for any structural and causal model conform with our definitions in section 4.

7. Conclusions

This work proposes an efficient and unified solution for hybrid systems fault detection, isolation and identification, assuming that it is possible to identify the current system state.

Efficiency is obtained by avoiding the explicit consideration of every possible mode configuration. HPCs, avoid computing PCs from scratch for every new configuration. Finally, a new algorithm is proposed for efficient on-line computation of minimal DBN-PCs.

Implementing HPCs as minimal DBNs provides a unified solution, because DBNs naturally allow fault detection, fault isolation and fault identification of continuous systems. Using HPCs we transform a hybrid diagnosis problem in a sequence of continuous diagnosis problems, avoiding the use of hybrid DBNs. An additional effect of using HPCs to generate DBNs is that we must not simulate the complete system DBN model, thus improving on-line computational efficiency.

As further work, we plan to integrate this proposal in a common framework including both discrete and parametric faults (Moya, Bregon, Alonso-González, Pulido, & Biswas, 2012; Moya, Bregon, Alonso-González, & Pulido, 2013). Besides, we want to incrementally generate the MEM in new modes from the MEM in the previous one; and to test the approach in a more demanding scenario with larger set of modes, and faster dynamics. Finally, we will couple this framework with a reliable hybrid state estimator.

Acknowledgments

This work has been partially supported by the Spanish MCI TIN2009-11326 grant.

References

Alonso-Gonzalez, C., Moya, N., & Biswas, G. (2011). Dynamic Bayesian network factors from possible conflicts for continuous system diagnosis. In *Proc. of the 14th Intl. Conf. on Advances in AI* (pp. 223–232). Berlin: Springer-Verlag. Available from `http://dl.acm.org/citation.cfm?id=2075561.2075588`

Armengol, J., Bregón, A., Escobet, T., Gelso, E., Krysander, M., Nyberg, M., et al. (2009). Minimal Structurally Overdetermined sets for residual generation: A comparison of alternative approaches. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS09* (pp. 1480–1485).

Bayoudh, M., Travé-Massuyès, L., & Olivé, X. (2009). Diagnosis of a Class of Non Linear Hybrid Systems by On-line Instantiation of Parameterized Analytical Redundancy Relations. In *Proc. of the XX Intl. Workshop on Principles of Diagnosis, DX'09* (p. 283-289). Stockholm, Sweden.

Bregon, A., Alonso, C., Biswas, G., Pulido, B., & Moya, N. (2012). Fault Diagnosis in Hybrid Systems using Possible Conficts. In *Proc. of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Safeprocess'12.* Mexico City, Mexico.

Bregon, A., Biswas, G., & Pulido, B. (2012). A Decomposition Method for Nonlinear Parameter Estimation in TRANSCEND. *IEEE Trans. Syst. Man. Cy. Part A*, *42*(3), 751-763.

Bregon, A., Pulido, B., Biswas, G., & Koutsoukos, X. (2009). Generating Possible Conflicts from Bond Graphs Using Temporal Causal Graphs. In *Proceedings of the 23rd European Conference on Modelling and Simulation, ECMS'09* (p. 675-682). Madrid, Spain.

Cocquempot, V., El Mezyani, T., & Staroswiecki, M. (2004, Jul.). Fault detection and isolation for hybrid systems using structured parity residuals. In *Proceedings of the 5^{th} Asian Control Conference.*

Hofbaur, M., & Williams, B. (2004, Oct.). Hybrid estimation of complex systems. *IEEE T. Syst. Man. Cy. Part B*, *34*(5), 2178 -2191.

Karnopp, D., Margolis, D., & Rosenberg, R. (2006). *System Dynamics: Modeling and Simulation of Mechatronic Systems*. New York, NY, USA: John Wiley & Sons, Inc.

Koller, D., & Lerner, U. (2001). Sequential Monte Carlo Methods in Practice. In N. d. F. A. Doucet & N. Gordon (Eds.), (chap. Sampling in factored dynamic systems). Springer.

Koutsoukos, X., Kurien, J., & Zhao, F. (2003). Estimation of Distributed Hybrid Systems Using Particle Filtering Methods. In *Proc. of the Intl. Workshop on Hybrid Systems: Computation and Control, HSCC'03* (p. 298-313). Springer.

Moya, N., Bregon, A., Alonso-González, C., & Pulido, B. (2013). A Common Framework for Fault Diagnosis of Parametric and Discrete Faults using Possible Conflicts. In *Proc. of the 15th Intl. Conf. on Advances in AI* (pp. 239–249). Madrid, Spain: Springer-Verlag, Berlin.

Moya, N., Bregon, A., Alonso-González, C., Pulido, B., & Biswas, G. (2012, Jul-Aug). Automatic Generation of Dynamic Bayesian Networks for Hybrid Systems Fault Diagnosis. In *Proc. of the XXIII Intl. Workshop on Principles of Diagnosis, DX'12.* Great Malvern, UK.

Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. Unpublished doctoral dissertation, University of California, Berkeley.

Narasimhan, S., & Biswas, G. (2007, May). Model-Based Diagnosis of Hybrid Systems. *IEEE T. Syst. Man Cy. Part A*, *37*(3), 348-361.

Narasimhan, S., & Brownston, L. (2007). HyDE: A General Framework for Stochastic and Hybrid Model-based Diagnosis. In *Proc. the XVIII Intl. Workshop on Principles of Diagnosis, DX'07* (p. 186-193). Nashville, USA.

Podgursky, B., G. B., & Koutsoukos, X. (2010, Oct). Efficient Tracking of Behavior in Complex Hybrid Systems via Hybrid Bond Graphs. In *Proc. of the Prognostics and Health Management Conf., PHM'10.* Portland, OR, USA.

Pulido, B., & Alonso, C. (2001). Dealing with cyclical configurations in (MORDRED). In *Proc. of the 9th Intl. Conf. on Advances in AI, CAPIA'01* (p. 983-992). Gijon, Spain.

Pulido, B., & Alonso-González, C. (2004, Oct.). Possible Conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. Sys. Man Cy. Part B*, *34*(5), 2192-2206.

Pulido, B., Bregon, A., & Alonso-González, C. (2010). Analyzing the influence of differential constraints in Possible Conflicts and ARR Computation. In P. Meseguer, L. Mandow, & R. Gasca (Eds.), *Current Topics in Artificial Intelligence* (Vol. 5988, p. 11-21). Springer Berlin.

Rienmüller, T., Bayoudh, M., Hofbaur, M., & Travé-Massuyès, L. (2009). Hybrid Estimation through Synergic Mode-Set Focusing. In *Proc. of 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Safeprocess'09* (p. 1480-1485). Barcelona, Spain.

Roychoudhury, I., Biswas, G., & Koutsoukos, X. (2008, Sept.). Comprehensive Diagnosis of Continuous systems Using Dynamic Bayes Nets. In *Proc. of XIX Intl. Workshop on Principles of Diagnosis, DX'08.* Blue Mountains, Australia.

Roychoudhury, I., Daigle, M., Biswas, G., & Koutsoukos, X. (2011, June). Efficient simulation of hybrid systems: A hybrid bond graph approach. *SIMULATION: Transactions of the Society for Modeling and Simulation International*(6), 467-498.

BIOGRAPHIES

Belarmino Pulido Belarmino Pulido received his Degree, MsC degree, and PhD degree in Computer Science from the University of Valladolid, Valladolid, Spain, in 1992, 1995, and 2001 respectively. In 1994 he joined the Departamento de Informatica in the University of Valladolid, where he is Associate Professor since 2002. He has been working on Model-based reasoning and Knowledge-based reasoning, and their application to Supervision and Diagnosis. He has worked in different national and European funded projects related to Supervision and Diagnosis. He is a member of the IEEE (M'2000), ACM (M'2003), and CAEPIA (1997, part of ECCAI) professional associations. He is also the coordinator of the Spanish Network on Supervision and Diagnosis of Complex Systems since 2005. Currently he is working on model-based diagnosis and prognosis of continuous and hybrid systems, using both centralized and distributed approaches.

Noemi Moya receives her Degree, her MsC Degree, and PhD Degree in Computing Sciences from the Universidad de Valladolid, Valladolid, Spain, in 2006, 2008, and 2013, respectively. From 2008 to 2013 she has been a research and teaching assistant at the Departamento de Informatica from the Universidad de Valladolid. She has been working in Model-Based Diagnosis of continuous and hybrid systems. She has been part of a national project related to diagnosis and she has been a member of the Spanish National Network on Supervision and Diagnosis of Complex Systems.

Carlos J. Alonso-González got his degree in Sciences in 1985, and his PhD in Physics in 1990, from the University of Valladolid, Valladolid, Spain. Currently he is Associate Professor with the Department of Computer Science from the University of Valladolid. He has being working in different national funded projects related to Supervision and Diagnosis of continuous industrial environments. His main research interests are Knowledge-based systems for Supervision and Diagnosis of dynamic systems, model-based diagnosis, Knowledge Engineering and Machine Learning.

Anibal Bregon received his B.Sc., M.Sc., and Ph.D. degrees in Computer Science from the University of Valladolid, Spain, in 2005, 2007, and 2010, respectively. From September 2005 to June 2010, he was Graduate Research Assistant with the Intelligent Systems Group at the University of Valladolid, Spain. He has been visiting researcher at the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA; the Dept. of Electrical Engineering, Linkoping University, Linkoping, Sweden; and the Diagnostics and Prognostics Group, NASA Ames Research Center, Mountain View, CA, USA. Since September 2010, he has been Assistant Professor and Research Scientist at the Department of Computer Science from the University of Valladolid. Dr. Bregon is a member of the Prognostics and Health Management Society and the IEEE. His current research interests include model-based reasoning for diagnosis, prognostics, health-management, and distributed diagnosis and prognostics of complex physical systems.

Health Monitoring and Prognosis of Hybrid Systems

Said Zabi[1], Pauline Ribot[2], and Elodie Chanthery[3]

[1,2,3] *CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France*

[1,2] *Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France*
said.zabi@laas.fr
pauline.ribot@laas.fr
[3] *Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France*
elodie.chanthery@laas.fr

ABSTRACT

Maintenance and repair of complex systems are an increasing part of the total cost of final product. Efficient diagnosis and prognosis techniques have to be adopted to detect, isolate and anticipate faults. Moreover the recent industrial systems are naturally hybrid: their dynamic behavior is both continuous and discrete. This paper presents an architecture of health monitoring and prognosis for hybrid systems. By using model and experience-based approach we propose an implementation of an integrated diagnosis/prognosis process based on Weibull probabilistic model. This article focuses particularly on the prognosis algorithm description. The process has been implemented and tested on Matlab. Simulation results on a water tank system show how prognosis and diagnosis interact into the architecture.

1. INTRODUCTION

Due to new technologies, the development of industrial systems is increasingly complex. It becomes difficult to manage and anticipate the behavior of these systems, especially when they are exposed to failures. In time of economic crisis, it is particularly essential to improve the system availability by reducing maintenance and repair costs. Efficient diagnosis and prognosis techniques have to be adopted to detect, isolate and anticipate faults leading to failures. Health monitoring of industrial systems aims at determining the health state of systems at any time in order to optimize their functioning and act in case of malfunctions. Diagnosis helps to determine the current health state of a system. Malfunctions or failures may be anticipated by a prognostic reasoning on the system. The PHM community defines the prognosis as the calculation of the remaining useful life (RUL). This temporal prediction gives the date at which the system is not operational anymore and must to be repaired. When the system is in such a state, it is said to be in a *failure mode*. Before this state, it can be either in a *nominal mode*, or after the occurrence of one or several faults in a *faulty mode* or degraded mode. Prognosis requires the knowledge of the current health state of the system through a diagnosis, hence the need of a health monitoring method integrating diagnosis and prognosis.

Recent industrial systems exhibit an increasing complexity of behaviors that are both continuous and discrete. It has become difficult to ignore the fact that most systems are hybrid. Therefore this paper proposes to use the techniques of model-based diagnosis and prognosis in the framework of hybrid systems. We propose to enrich the commonly used modeling framework for hybrid systems with available knowledge about aging or degradation of the system. Systems are continuously degrading according to operational conditions. According to information available on the system, it is possible to establish physical aging laws or time-dependent fault probabilities based on the feedback. This temporal and/or stochastic information should be taken into account in the model of the hybrid system.

The main ideas on the integrated architecture for diagnosis and prognosis of hybrid systems we proposed have been published in Chanthery and Ribot (2013). This paper focuses particularly on the prognosis algorithm description. It begins with a brief overview of the related work in diagnosis and prognosis on hybrid systems. Section 3 gives an overview of our architecture for interleaving diagnosis and prognosis processes. The diagnosis process is briefly described. Then section 4 presents the hybrid system modeling for prognosis. Section 5 is the core of the article. It details the prognosis algorithm. The algorithm has been implemented. Simulation results are given on Section 6, illustrated by a water tank system. Finally, Section 7 concludes the paper and proposes some future work.

Saïd Zabi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. Related Work

There has been considerable work on diagnosis of hybrid systems on one hand, and on prognosis on the other hand. However, to the best of our knowledge, very few studies succeed in coupling diagnosis and prognosis and the authors could not find related work dealing with prognosis on hybrid systems.

A formal generic modeling framework for a complex system is presented in Ribot, Pencolé, and Combacau (2009) that encapsulates the knowledge used by diagnosis and prognosis. In this work, the authors establish a coupling of diagnosis and prognosis based on a characterization of complex system modes but no algorithm and implementation have been proposed. Another approach has been proposed in Roychoudhury and Daigle (2011). The authors propose a common framework for diagnosis and prognosis thanks to a state representation that describes the nominal behavior of the system and fault progression. However, there is absolutely no hybrid or discrete aspect in this work. The model used is a state model that specifies the system behavior in nominal modes and in faulty modes. A parameter vector and an associated evolution equation are used to represent fault progression over time. The method consists in building an observer from the nominal behavior to perform fault detection. The identification is made from a set of observers that are built for each fault. Prognosis consists in predicting the remaining useful life (RUL) for each fault using an estimator based on a fault progression model.

Most of the works on discrete event systems consider prognosis as a prediction of an event trajectory or fault event occurrences (Genc & Lafortune, 2009). The term ”predictability” of a fault event introduced by Cao (1989) is based on the system observability property. It is clearly related to the diagnosability notion in discrete event systems: ”it is certain that a critical event will take place”. Genc and Lafortune (2009) demonstrates that any predictable event is diagnosable. An extension for the prediction of event patterns is proposed in Jéron, Marchand, Genc, and Lafortune (2008). In these studies, the system model is a classical automaton in which only ordered, undated and without delay event sequences are considered.

To perform prognosis, it is required to take the temporal aspect into account to compute the RUL of the system. Only Khoumsi (2009) uses a timed automata (TA) in order to prognose a fault event on the system. Clock ticks are added to transitions of the TA to determine the dated trajectories leading to fault events. No notion of uncertainty (neither by mean of probabilities nor intervals) is taken into account in these timed automata. However, uncertainty is intrinsically linked to prognosis.

In Zemouri and Faure (2006), the evolution of the system operating state is modeled by a stochastic timed automaton (STA). A stochastic distribution $f(t)$ is associated to each transition of the automaton. The distribution $f(t)$ gives the probability of occurrence for x_{j+1} at time t_{j+1} after the occurrence of x_j at time t_j. In this study, events occurring in the system are represented in the states of the stochastic timed automaton that does not take the hybrid dynamics of the system into account.

Castaneda, Aubry, and Brinzei (2010) proposes a stochastic hybrid automaton to evaluate the system dynamic reliability. The stochastic hybrid automaton represents the possible behavioral modes of the system. The stochastic part helps to take faults and uncertainties about system knowledge into account. The system switches from one mode to another with events that may be deterministic or stochastic. Stochastic events occur when a threshold on their probability law has been reached. In this study, stochastic transitions have a constant rate. The model is simulated to obtain availability and reliability defined as the probabilistic evaluation of the hybrid system failure.

In order to complete previous works, the first issue investigated in this paper is the representation of the behavior of the hybrid system and the uncertainty of its degradation into a single model. The second issue concerns the implementation and the test of a health monitoring method coupling diagnosis and prognosis.

3. Architecture for Interleaving diagnosis and prognosis processes

In this study, a system is supposed to start operating in nominal behavior. A set of critical faults has been anticipated for the system and each one of them may occur at any time from each nominal state. Faults are supposed to be permanent: it means that once a fault has occurred, the system evolves in what is called a *faulty mode*. This degradation can evolve into a worst degraded mode. Finally, when the system is not operational anymore, it is said to be in a *failure mode*. Obviously, it is possible that the system directly evolves from a nominal mode towards a failure mode. It is a particular case of what is described in the article. Without maintenance or repair action, the evolution of a system is then supposed to be unidirectional. This evolution of the system from a nominal mode to a failure mode is illustrated in Figure 1.

The combination of faults leading to a failure can be established from a fault tree analysis (Rausand & Hoyland, 2004). We define a prognosis as a prediction of the sequence of fault dates. With the analysis of the fault tree and what we call a prognosis, it is simple to obtain the system RUL that corresponds to the remaining time until the system failure. This fault analysis allows to link our prognosis definition to the one commonly used in the PHM community.

This section begins by describing the architecture interleav-

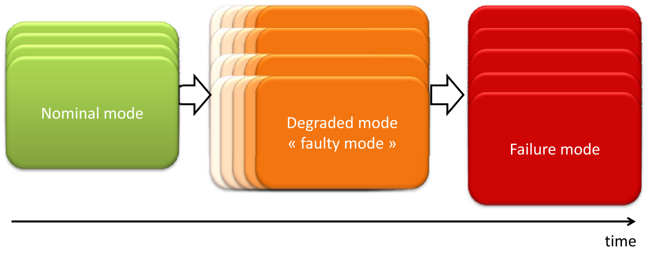


Figure 1. Unidirectional system evolution without maintenance or repair action.

ing diagnosis and prognosis. It will then go on the description of the diagnosis process.

3.1. Overview of the architecture

The originality of our work is to interleave diagnosis and prognosis processes that are too often considered separately. This section presents an architecture coupling diagnosis and prognosis into a single new process called *InterDP*. This architecture, illustrated in Figure 2, has already been described in Chanthery and Ribot (2013).

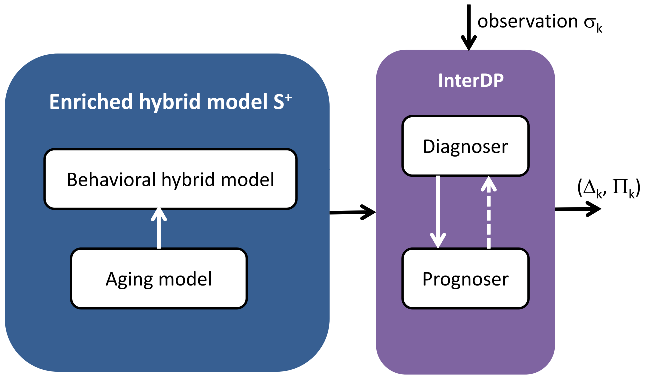


Figure 2. *InterDP* process interactions

Inputs of the *InterDP* process are an enriched hybrid model S^+ defined for our methodology and observations that will be formally defined in the next section. To put it in simple terms, the enriched hybrid model contains all the knowledge about the system behavior and degradation that is required to perform diagnosis and prognosis. Observations are a set of observable events. These events are built from information issued for example from the sensors that are embedded in the system.

The output of InterDP at each clock tick t_k is a couple (Δ_k, Π_k) of diagnosis and prognosis results. Diagnosis Δ_k is performed by a hybrid diagnoser (Bayoudh, Travé-Massuyès, & Olive, 2008). It is built off-line from the hybrid automaton part of the enriched hybrid model. On-line, it takes as input the observations and provides a vector containing all possible modes (nominal, faulty or failure modes) for the system that are consistent with observations:

$$\Delta_k = \begin{bmatrix} \Delta_k^1 \\ \Delta_k^2 \\ \vdots \\ \Delta_k^D \end{bmatrix}$$

where D is the number of diagnosis hypotheses. Δ_k can be seen as a belief state of the hybrid system. The diagnoser may take also as input information from the prognoser (dashed arrow in the figure), in order to disambiguate some diagnosis hypotheses. This link is not the core of this article and will be studied in future works.

Prognosis Π_k is performed by a prognoser. On-line, at each clock tick t_k, the prognoser takes as input the enriched hybrid model and the result of the diagnosis process and provides a vector:

$$\Pi_k = \begin{bmatrix} \Pi_k^1 \\ \Pi_k^2 \\ \vdots \\ \Pi_k^D \end{bmatrix}$$

where Π_k^i is associated to a diagnosis Δ_k^i and represents the most likely sequence of dated faulty modes leading to the system failure:

$$\Pi_k^i = (\{f_1, d_1\}, \ldots, \{f_j, d_j\}, \ldots, \{f_N, d_N\}).$$

where d_i is the date occurrence of fault f_j and N represents the number of degraded modes before the failure mode.

One hypothesis in our work is that the system is assumed to be diagnosed after each new observation, that is to say when a change in measurements is detected. As diagnosis consists in monitoring the diagnoser, the diagnosis computation duration can be considered as instantaneous. It is also supposed that between two different observations, both diagnosis and prognosis can be performed. Let t_k be an occurrence of an observation, t_{k+1} be the occurrence of the next observation and CT_p be the computation time for prognosis.
Hypothesis 1. *The computation time of the prognosis process is smaller than the interval between two different observations.*

$$CT_p \leq (t_{k+1} - t_k) \qquad (1)$$

3.2. Diagnosis process

The diagnosis process in *InterDP* has been described in detail in Chanthery and Ribot (2013) and is not the focus point of this paper. We just recall here the main steps of the process. The hybrid diagnoser is built from the hybrid behavioral automaton of S^+ that is formally defined is the next section. On-line it takes as input the set of observations on the system. The diagnosis method for hybrid systems that is adopted for our approach is the one developed in Bayoudh (2009).

It interlinks a standard diagnosis method for continuous systems, namely the parity space method, and a standard diagnosis method for DES, namely the diagnoser method (Sampath, Sengputa, Lafortune, Sinnamohideen, & Teneketsis, 1995). The diagnosis part of the methodology may be decomposed into three parts:

- diagnose the continuous part of the system,
- abstract the continuous part in terms of discrete events and enrich the discrete part of the system with discrete events that come from the abstraction of the continuous part,
- then apply the diagnoser method on the resulting discrete event system in order to build a diagnoser able to follow on-line the behavior of the system according to the observable events.

4. Hybrid system modeling for prognosis

4.1. Hybrid formalism

The modeling framework that is adopted for hybrid systems is based on a hybrid automaton (Henzinger, 1996). The hybrid automaton is defined as a quintuple $S = (\zeta, Q, \Sigma, T, C, (q_0, \zeta_0))$ where:

- ζ is the set of continuous variables that comprises input variables $u(t) \in R^{n_u}$, state variables $x(t) \in R^{n_x}$, and output variables $y(t) \in R^{n_y}$. The set of directly measured variables is denoted by ζ_{OBS}.
- Q is the set of discrete system states. Each state $q_i \in Q$ represents a behavioral mode of the system. It includes nominal and anticipated faulty modes, including failure modes. The anticipated faulty modes are faulty modes that are known to be possible on the system. The unknown mode can be added to model all the non-anticipated faulty situations.
- Σ is the set of events that correspond to discrete control inputs, autonomous mode changes and fault occurrences. $\Sigma = \Sigma_{uo} \cup \Sigma_o$, where $\Sigma_o \subseteq \Sigma$ is the set of observable events and $\Sigma_{uo} \subseteq \Sigma$ is the set of unobservable events.
- $T \subseteq Q \times \Sigma \rightarrow Q$ is the partial transition function. The transition from mode q_i to mode q_j with associated event σ_{ij} is noted $t(q_i, \sigma_{ij}, q_j)$ and we have $T(q_i, \sigma_{ij}) = q_j$. T also denotes the set of transitions.
- $C = \bigcup_i C_i$ is the set of system constraints linking continuous variables. C_i denotes the set of constraints associated to the mode q_i. C represents the set of differential and algebraic equations modeling the continuous behavior of the system. The continuous behavior in each mode is assumed to be linear.
- $(\zeta_0, q_0) \in \zeta \times Q$, is the initial condition.

The occurrence of a fault is modeled by a discrete event $f_i \in \Sigma_F$. Σ_F is the set of fault events associated to the anticipated faults of F. Without loss of generality it is assumed that $\Sigma_F \subseteq \Sigma_{uo}$. The discrete part of the hybrid automaton is given by $M = (Q, \Sigma, T, q_0)$, which is called the *underlying discrete event system (DES)* and the continuous behavior of the hybrid system is modeled by the so-called *underlying multi-mode system* $\Xi = (\zeta, Q, C, \zeta_0)$. An example of a hybrid system is given in Figure 3.

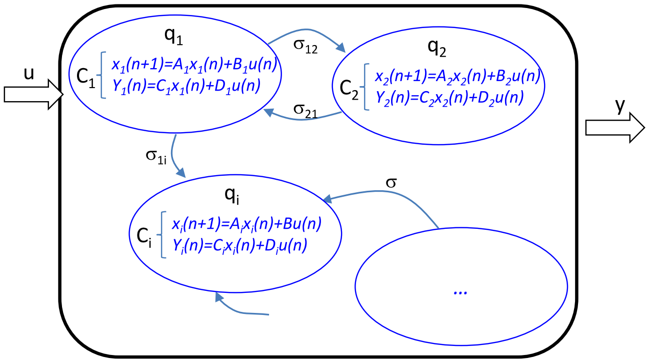


Figure 3. Example of an hybrid system

This hybrid automaton describes the set of knowledge useful to achieve model-based diagnosis. In order to perform prognosis, it is necessary to enrich the hybrid model by adding the available knowledge about the aging or the degradation of the system. A way to take the uncertainty on the degradation function into account is to introduce probability measures for each state that represents a mode of the system.

4.2. Aging modeling

The modeling framework that is adopted for the system degradation is based on the Weibull probabilistic model (Ribot & Bensana, 2011). A particular way for representing the remaining useful life of systems is to establish a fault probability from reliability analyses at different stress levels (operating conditions) (Hall & Strutt, 2003; Vachtsevanos, Lewis, Roemer, Hess, & Wu, 2006). Stress is defined as the set of internal and external conditions/factors that may have an impact on the system behavior. The parametrized Weibull model is often used in reliability for life data analyses due to its flexibility (Ferreiro & Arnaiz, 2008):

$$W(t, \beta, \eta, \gamma) = \frac{\beta}{\eta}\left(\frac{t-\gamma}{\eta}\right)^{(\beta-1)} e^{-(\frac{t-\gamma}{\eta})^{\beta}} \qquad (2)$$

where $t \geq 0$, $\beta \geq 0$, $\eta \geq 0$ and $\gamma \in [-\infty; \infty]$. The scale characteristic η defines the characteristic life of the system and corresponds to the mean life expectancy for a studied population sample. The shape characteristic β modifies the probability density function (pdf) nature and allows to model the different life phases of a system defined by the idealized bathtub curve of reliability. The location characteristic γ shifts the curve from the origin. It defines the system minimal life. The case $\gamma > 0$ means that the fault probability is zero until a date γ. In most cases, we assume $\gamma = 0$. The characteristic η is stress-dependent while β is assumed to remain constant across different stress levels.

Weibull characteristics $\beta_j^{q_i}, \eta_j^{q_i}, \gamma_j^{q_i}$ model the aging evolution of a system that leads to a fault f_j in a behavioral mode q_i and totally define the fault probability distribution $f_j^{q_i}$:

$$f_j^{q_i}(t) = \int_0^t W(t, \beta_j^{q_i}, \eta_j^{q_i}, \gamma_j^{q_i})dt. \tag{3}$$

The fault probability density function $W(t, \beta_j^{q_i}, \eta_j^{q_i}, \gamma_j^{q_i})$ has to give at any time the probability that the fault f_j occurs in the system from a mode q_i. Weibull characteristics $\beta_j^{q_i}$ and $\eta_j^{q_i}$ are fixed by the mode q_i of the system. The location characteristic $\gamma_j^{q_i}$ can be used to memorize the degradation evolution of the system in the past modes from the operation start of the system (Ribot & Bensana, 2011). At first, the system is in a mode q_0. If the system has never been used, q_0 obviously represents the nominal mode and we suppose that $\forall f_j, \quad \gamma_j^{q_0} = 0$ as previously explained. This characteristic $\gamma_j^{q_i}$ will be modified to take degradation in each behavioral mode into account during the system operation.

The occurrence date d_{f_j} of a fault event f_j for the system in mode q_i can be determined from a decision criterion $P_{max_{f_j}}$ that corresponds to a probability threshold beyond which the risk becomes unacceptable:

$$d_{f_j} \text{ such that } \int_0^{d_{f_j}} W(t, \beta_j^{q_i}, \eta_j^{q_i}, \gamma_j^{q_i})dt = P_{max_{f_j}}. \tag{4}$$

4.3. Enriched hybrid model

In each mode q_i, the system is subject to different aging laws. The set of aging laws is supposed to be accurately known.

Hypothesis 2. *An aging law of a system is supposed to be continuous over time.*

The consequence of this hypothesis is that the initial condition for an aging law at time t_{k+1}^+ is the value at t_{k+1}^-, when the system has not yet commuted between two modes.

To take into account the different aging laws, the hybrid system is then described as an enriched hybrid automaton $S^+ = (\zeta, Q, \Sigma, T, C, \mathcal{F}, (q_0, \zeta_0))$, where $\mathcal{F} = \{F^{q_i}, i \in \{1, \ldots, card(Q)\}\}$ is the set of aging laws associated to behavioral modes or the system. F^{q_i} is a vector of aging laws for each anticipated fault in the mode q_i. For example, in a system where N_F faults are considered:

$$F^{q_i}(t) = \begin{bmatrix} f_1^{q_i}(t) \\ f_2^{q_i}(t) \\ \ldots \\ f_{N_F}^{q_i}(t) \end{bmatrix} \tag{5}$$

where $f_j^{q_i}$ represents the probability distribution of the fault f_j at any time in mode q_i.

It can be noticed that as opposed to Ribot and Bensana (2011), the hybrid automaton we propose represents behavioral modes and not operational modes based on function availability.

For example, for a system with two nominal modes q_{01}, q_{02}, two possible actions a_1, a_2 that are observable events, and two faults f_1 and f_2, a possible model is given in Figure 4. This system is in a failure mode when f_1 and f_2 have occurred. If only one fault occurred, then the system is in a faulty mode.

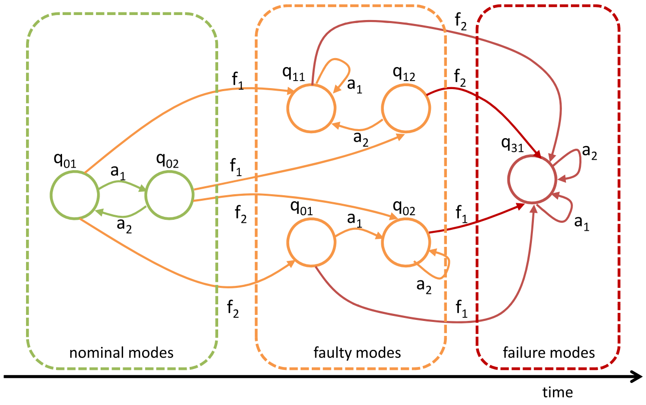


Figure 4. Example for a system with 2 nominal modes

5. Prognosis

The focus of this article is on the prognosis process. It consists, at each clock tick t_k, in computing the most likely fault sequence Π_k until the system failure. Algorithm 1 describes how the prognosis process is structured and introduces the three main functions of the process.

Prognosis takes as input information on aging laws in S^+ associated to the set of anticipated faults $\mathcal{F}$ and the set of behavioral models Q. It takes also as input the current result of the diagnosis Δ_k to update on-line the system aging laws according to the operation time in each behavioral mode. After each observable event, the appropriate aging laws are selected (line 11) according to the mode that is estimated by the diagnosis and the fault probability value reached in previous modes. Then the prognosis process predicts the most probable fault sequence (lines 6 to 14) supposing that the system

Algorithm 1: Prognosis of a hybrid system

```
Inputs: enriched model $S^+$, on-line diagnosis $\Delta_k$
Outputs: $RUL$, fault sequences $\Pi_k$

1: k = 0
2: $q- \leftarrow q_0$ {Mode initialization}
3: for each anticipated fault $f_j \in \Sigma_f$ do
4:     $(f_j^{q-}, d_{f_j}) \leftarrow$ InitializeAgingParameters($q-$)
5: end for
6: $(\Pi_0, RUL) \leftarrow$ PredictFaultSequence($S^+$, $q-$, $\{d_{f_j}\}$)
7: repeat
8:     k++
9:     if $\Delta_k \neq q-$ then
10:        $q+ \leftarrow \Delta_k$
11:        $\{(f_j^{q+}, d_{f_j})\} \leftarrow$
           UpdateAgingParameters($S^+, q-, q+$)
12:        $q- \leftarrow q+$
13:        $(\Pi_k, RUL) \leftarrow$ PredictFaultSequence($S^+$, $q-$,
           $\{d_{f_j}\}$)
14:    end if
15: until $RUL = 0$
```

remains in the current mode. It is possible to "disconnect" prognosis from the diagnosis in the architecture and to execute the prognoser module independently. Obviously in this case the system is supposed to remain in the current mode.

Thus three main functions may be distinguished: the aging parameters initialization (line 3), the aging parameters update (line 11) and the fault sequence prediction (lines 6 to 14). The following subsections describe precisely how the prognosis process is built and interleaved with diagnosis.

5.1. Aging function parameters initialization

The system is degrading in different ways leading to fault occurrences that may provoke a failure. This degradation depends on the mode of the hybrid system. For each mode, the degradation embodies the impact of stress factors. We recall that the aging dynamic of the system exposed to a fault f_j in mode q_i is modeled by a set of Weibull parameters $\beta_j^{q_i}$,$\eta_j^{q_i}$ and $\gamma_j^{q_i}$ in the enriched model S^+.

For the simplicity of the presentation, we first assume that there is no problem of diagnosability in the system that is studied. It means that at each clock tick, the system mode is totally known, i-e. non ambiguous and is given by the diagnosis process. This is a high hypothesis, and the case of ambiguity in the system state has to be studied in the future. Nevertheless, an easy solution to transform an ambiguous case into a non ambiguous one is to consider that the system is in its most probable state.

When the prognosis process is started, the system is in initial state (q_0, ζ_0). The aging law $f_j^{q_0}(t)$ associated to each anticipated fault f_j is initialized. Parameters $\beta_j^{q_0}$ and $\eta_j^{q_0}$ are assumed to be fixed and derived from reliability analyses. If the studied system has never been used before, q_0 obviously represents the nominal mode and we suppose that $\forall f_j,\ \gamma_j^{q_0} = 0$ as previously explained. If q_0 is not nominal, the enriched model has to give information about the initial values of $\gamma_j^{q_0}$.

The occurrence date d_{f_j} of each anticipated fault f_j for the system in mode q_0 is then determined from a decision criterion $P_{max_{f_j}}$:

$$\int_0^{d_{f_j}} W(t, \beta_j^{q_0}, \eta_j^{q_0}, \gamma_j^{q_0}) dt = P_{max_{f_j}}. \tag{6}$$

All along its operation, the aging probability of the system exposed to a fault f_j is denoted by P_{f_j}. It is evaluated on-line with diagnosis and predicted to determine the RUL. Knowing the aging dynamic of the system in the initial mode and the different aging threshold, the prognosis process predicts the most probable fault sequence until the failure mode. This prediction process is done each time the diagnosis process updates the current mode of the system. The next two sections describe how the aging parameters need to be updated after a new diagnosis result and how fault sequences are then predicted.

5.2. Aging parameters update

Algorithm 2 describes how the aging functions are updated after each new diagnosis result.

Algorithm 2: UpdateAgingParameters

```
Inputs: Enriched model $S^+$, previous behavioral mode $q-$,
    new behavioral mode $q+$ provided by diagnosis
Outputs: New aging laws $f_j^{q+}$ for each anticipated fault $f_j$,
    new dates of fault occurrences $d_{f_j}$
for each anticipated fault $f_j \in \Sigma_f$ do
    $P_{f_j} \leftarrow$ ComputeAgingProbability($f_j^{q-}$) {with
    Equation (7)}
    $\gamma_j^{q+} \leftarrow$ ComputeLocationParameter($P_{f_j}$,$q+$) {with
    Equations (8) (9)}
end for
```

On receipt of a new mode estimation $q+$ at time t^+, the Weibull aging functions associated to faults in the mode $q+$ are updated according to the time spent by the system in previous mode $q-$. The aging probability associated to a fault f_j that the system has reached in past mode(s) at t^+ is computed with

$$P_{f_j} = \int_0^{t^+} W(t, \beta_j^{q-}, \eta_j^{q-}, \gamma_j^{q-}) \mathrm{d}t. \tag{7}$$

To memorize this aging probability P_{f_j}, a new value for characteristic γ_j^{q+} of aging model associated to the fault f_j in the new mode $q+$ is computed, knowing the values of β_j^{q+} and

η_j^{q+} thanks to the enriched model:

$$\gamma_j^{q+} = (t^+ - \delta) \text{ such that}$$
$$\int_0^{\delta} W(t, \beta_j^{q+}, \eta_j^{q+}, 0)\mathrm{d}t = P_{f_j}. \quad (8)$$

With the above equation, we introduce a mathematical manipulation to memorize the aging probability P_{f_j} reached in past modes into new aging models for mode $q+$ from t^+. By this calculation, the continuity of any aging function is guaranteed in all mode change points and

$$\int_0^{t^+} W(t, \beta_j^{q+}, \eta_j^{q+}, \gamma_j^{q+})\mathrm{d}t = P_{f_j} \quad (9)$$

Figure 5 illustrates how Weibull pdf associated to a fault f_1 evolves in order to describe the fault aging in two behavioral modes q_{01} and q_{02}. The two curves on the top illustrate the Weibull pdf associated to fault f_1 in mode q_{01} (on the left) and in mode q_{02} (on the right). Let suppose that the system is in mode q_{01} at $t = 0$, then P_{f_1} corresponds to the area under the curve. When the system switches into mode q_{02}, the continuity condition on the aging law (Hypothesis 2) implies that the area under the first curve is equal to the area under the second one. So the second curve is translated on the left.

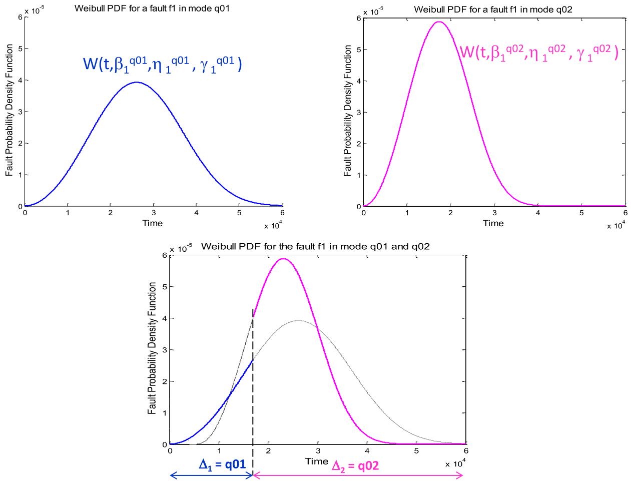


Figure 5. Weibull pdf associated to a fault f_1 for a system in two behavioral modes

5.3. Fault sequence prediction

To determine the sequence of fault events that leads to the system failure, we need to define a discrete fault event automaton (line 4) to extract the system faulty evolutions. This automaton describes all the possible fault transitions between operating modes until the system failure. The Discrete Fault Event System (DFES) is obtained by the projection of the underlying DES on the fault events (see Section 4.1). It corresponds to an abstraction of the hybrid system which contains fault events only. The DFES is a finite state machine $M_f = (Q_f, \Sigma_f, T_f, q_{f0})$ formally defined as follows:

- Q_f is the set of discrete states of the system,
- Σ_f is set of the fault events,
- $T_f \subseteq (Q_f \times \Sigma_f \rightarrow Q_f)$ is the partial transition function,
- $q_{f0} = q_0$ is the initial state corresponding to the nominal mode of the system.

The DFES of the example described in Figure 4 is illustrated by Figure 6.

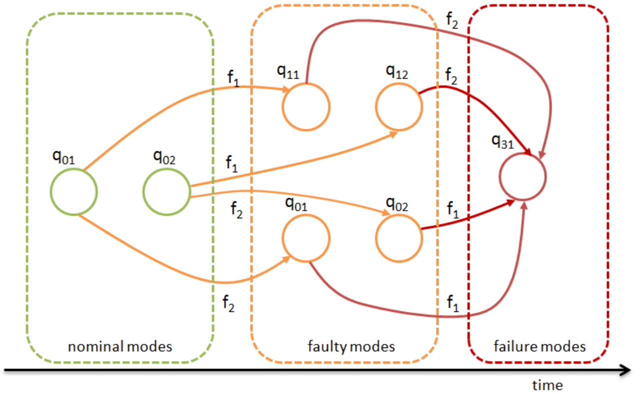


Figure 6. The DFES

Once a mode change is detected by diagnosis, the idea is to run through the DFES and predict the fault sequences until the system failure. A state q_c in a DFES is a deadlock if

$$\forall f \in \Sigma_f,\ T_f(q_c, f) = \emptyset \quad (10)$$

The stop criterion for the prediction function is a dead-lock (for example mode q_{31} in Figure 6).

For each fault f_j that has not occurred yet, the algorithm computes the date of occurrence of f_j. The minimum value in this set of dates is denoted d_{min_1}: it corresponds to the date of occurrence of the next fault, denoted f_{min_1} (line 6). The set of faults whose occurrence has not been predicted yet is denoted by Σ_{ff}, then Σ_{ff} is updated and $\Sigma_{ff} = \Sigma_f \backslash f_{min_1}$ (line 7). It then contains faults whose occurrence date is superior to d_{min_1}. At d_{min_1}, the system is predicted to switch into fault mode $q_{f_{min_1}}$. New aging models in fault mode $q_{f_{min_1}}$ (described by the Weibull pdf $W(t, \beta_j^{q_{f_{min_1}}}, \eta_j^{q_{f_{min_1}}}, \gamma_j^{q_{f_{min_1}}})$) have to be updated for each fault f_j in Σ_{ff} (line 12). The mode change predicted at d_{min_1} may result in a modification of fault dates $\{d_{f_j}\}$ that have been previously computed.

As for an aging parameter update resulting from a change in diagnosis, characteristic $\gamma_j^{q_{f_{min_1}}}$ of aging models in future mode $q_{f_{min_1}}$ has to be computed from the fault probability $P_{f_j}^1$ the system could have reached at predicted commutation time d_{min_1}. Let q_c denotes the current system mode, for each fault f_j in Σ_{ff} :

$$P_{f_j} = \int_0^{d_{min_1}} W(t, \beta_j^{q_c}, \eta_j^{q_c}, \gamma_j^{q_c})\mathrm{d}t, \quad (11)$$

and $\gamma_j^{q_{f_{min_1}}} = (d_{min_1} - \delta)$ such that

$$\int_0^{\delta} W(t, \beta_j^{q_{f_{min_1}}}, \eta_j^{q_{f_{min_1}}}, 0)\mathrm{d}t = P_{f_j}. \quad (12)$$

Characteristic $\gamma_j^{q_{f_{min_1}}}$ allows to memorize the system aging in all past modes from q_0 and guarantee the continuity of aging laws. The date d_{f_j} of fault occurrences in Σ_{ff} are modified as follows:

$$\int_0^{d_{f_j}} W(t, \beta_j^{q_{f_{min_1}}}, \eta_j^{q_{f_{min_1}}}, \gamma_j^{q_{f_{min_1}}})dt = P_{max_j}. \quad (13)$$

The next possible fault f_{min_2} after f_{min_1} is determined from the minimal predicted fault date d_{min_2} for faults in Σ_{ff}. Then $\Sigma_{ff} = \Sigma_{ff} \backslash f_{min_2}$. Fault propagation is studied as explained above to compute $\gamma_j^{q_{f_{min_2}}}$ for faults that have not reached their probability threshold at d_{min_2} using new aging models for mode $q_{f_{min_2}}$ and the process reiterates.

The prognosis process computes the most likely future sequence Π_k of dated fault events according to a diagnosis Δ_k:

$$\Pi_k = (\{f_{min_1}, d_{min_1}\}, \{f_{min_2}, d_{min_2}\}, \ldots, \{f_{min_N}, d_{min_N}\}). \quad (14)$$

Algorithm 3 sums up the procedure of fault sequence prediction.

6. Experimental results

HYDIAG is a software program on MATLAB developed by the DISCO team. It performs diagnosis of hybrid systems Bayoudh et al. (2008). The idea was to enrich it with Weibull aging models to performs prognosis, to implement the prognosis algorithm and interleave diagnosis and prognosis processes into a single one module named *InterDP*. This has been implemented and tested on a water tank system.

6.1. Modeling of a Water tank system

InterDP has been applied to the case study of a water tank system (Figure 7). This system is composed of one tank with two hydraulic pumps (P_1, P_2) providing water with different rates. Water flows through a valve at the bottom of the tank depending on the system control. Three sensors located at different tank levels (h_1, h_2, h_{max}) detect the water level and allows to set the control of the pumps (on/off). If the water level h is between h_1 and h_2, both pumps P_1 and P_2 are turned on. If $h_2 < h < h_{max}$, only P_1 is on and when $h \leq h_{max}$, the pumps are turned off. It is assumed that the pumps may fail only if they are on.

Algorithm 3: PredictFaultSequence

```
Inputs: Enriched model $S^+$, Current mode $q_c$, Dates of
    fault occurrence $\{d_{f_j}\}$
Outputs: Fault sequence $\Pi_k$, RUL
$\Pi_k = \emptyset$
$\Sigma_{ff} \leftarrow \Sigma_f$ {$\Sigma_{ff}$ is the set of faults to be predicted}
$i \leftarrow 1$
ConstructDFES($S^+$)
while $\exists f \in \Sigma_{ff} \mid T_f(q_c, f) \neq \emptyset$ {$q_c$ is not a dead-lock}
do
    $(f_{min}(i), d_{min}(i)) \leftarrow$ PredictNextFault($\Sigma_{ff}$, $\{d_{f_j}\}$)
    $\Sigma_{ff} \leftarrow \Sigma_{ff} \backslash f_{min}$
    $\Pi_k = \Pi_k \cup \{(f_{min}(i), d_{min}(i)\}$
    RUL $\leftarrow d_{min}(i)$
    $q_f \leftarrow T_f(q_c, f_{min}(i))$ {System is predicted to switch
    in mode $q_f$ at $d_{min}$}
    for each anticipated fault $f_j \in \Sigma_{ff}$ do
        $(f_j^{q_f}, d_{f_j}) \leftarrow$ UpdateAgingParameters($q_c$,$q_f$) {with
        Equations (11) (12) (13)}
    end for
    $i \leftarrow i + 1$
    $q_c \leftarrow q_f$
end while
```

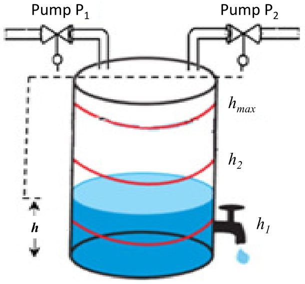
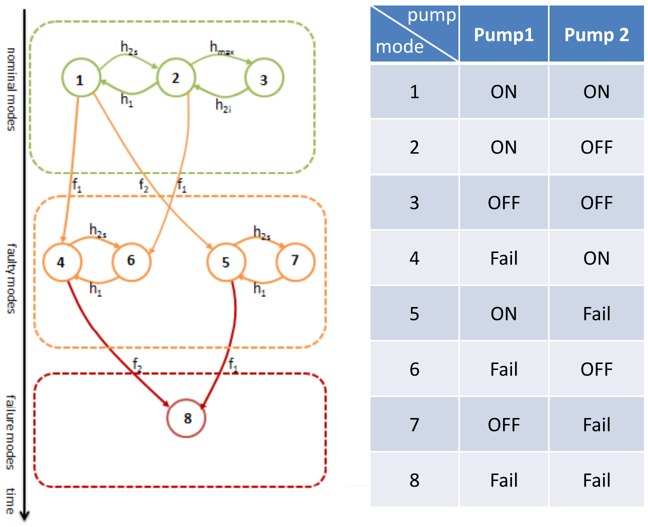


Figure 7. Water tank system

The discrete model of water tank and the controls of pumps are given in Figure 8. Eight discrete behavioral modes has been identified $Q = \{q_1, q_2, \ldots, q_8\}$. Discrete events in $\Sigma = \{h_1, h_{2s}, h_{2i}, h_{max}, f_1, f_2\}$ allow the system to switch into different modes. Observable events are $\Sigma_o = \{h_1, h_{2s}, h_{2i}, h_{max}\}$. They result from sensor measurements of water level in the tank. Two faults are anticipated $\Sigma_f = \{f_1, f_2\}$ and are not observable: f_1 corresponds to the failure of P_1 and f_2 is the failure of P_2. The parameter values of aging models $\mathcal{F} = \{F^{q_i}\}$ associated to every behavioral mode q_i are reported in Table 1. Let us recall that the Weibull characteristics γ is assumed to be equal to zero at the system start.

The underlying continuous behavior of every discrete mode



mode \ pump	Pump1	Pump 2
1	ON	ON
2	ON	OFF
3	OFF	OFF
4	Fail	ON
5	ON	Fail
6	Fail	OFF
7	OFF	Fail
8	Fail	Fail

Figure 8. Water tank DES model

Table 1. Weibull parameters of aging models

Aging laws		β	η
F^{q_1}	$f_1^{q_1}$	1.5	3000
	$f_2^{q_1}$	1.5	4000
F^{q_2}	$f_1^{q_2}$	2	3000
	$f_2^{q_2}$	1	7000
F^{q_3}	$f_1^{q_3}$	1	8000
	$f_2^{q_3}$	1	7000
F^{q_4}	$f_1^{q_4}$	NaN	NaN
	$f_2^{q_4}$	2	4000
F^{q_5}	$f_1^{q_5}$	2	3000
	$f_2^{q_5}$	NaN	NaN
F^{q_6}	$f_1^{q_6}$	NaN	NaN
	$f_2^{q_6}$	1	7000
F^{q_7}	$f_1^{q_7}$	1	8000
	$f_2^{q_7}$	NaN	NaN
F^{q_8}	$f_1^{q_8}$	NaN	NaN
	$f_2^{q_8}$	NaN	NaN

q_i for $i \in \{1..8\}$ is represented by the same state space:

$$\begin{cases} X(k+1) &= AX(k) + BU(k) \\ Y(k) &= CX(k) + DU(k) \end{cases} \tag{15}$$

where the state variable X is the water level in the tank, continuous inputs U are the flows delivered by the pumps P_1, P_2 and the flow going through the valve, $A = (1)$, $B = \begin{pmatrix} eTe/S \\ eTe/S \\ eTe/S \end{pmatrix}$ with Te the sample time, S the tank base area and $e_i = 1$ (resp. 0) if the pump is turned on (resp. turned off), $C = (1)$ and $D = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.

The continuous behavior is abstracted to build new observable discrete events Rox_y using the parity space approach. The enriched discrete event model of the hybrid system is used to build the diagnoser that will allow to track the system mode after each new observation.

The process *InterDP* was tested on this water tank hybrid model. Both diagnosis and prognosis are performed.

6.2. Simulation results

6.2.1. Simulation parameters

The time horizon is fixed at $T_{sim} = 4000$h, the sampling period is $T_s = 36$s and the filter sensitivity for the diagnosis is set as $T_{filter} = 3$min. The residual threshold is 10^{-12} as in Bayoudh (2009).

The scenarios involve a variant use of water (max flow rate = 1200L/h) depending on user needs during 4000h. Pumps are automatically controlled to satisfy the specifications indicated above. Flow rate of P_1 and P_2 are respectively 750L/h and 500L/h.

The diagnoser issued from the diagnosis process is given in Figure 9. Its computation is done off-line. Each state of the diagnoser indicates the belief state in the model enriched by the abstraction of the continuous part of the system, with a tag that gives the set of faults that have occurred on the system. This set is empty in case of nominal mode. This diagnoser shows that the tested system is diagnosable.

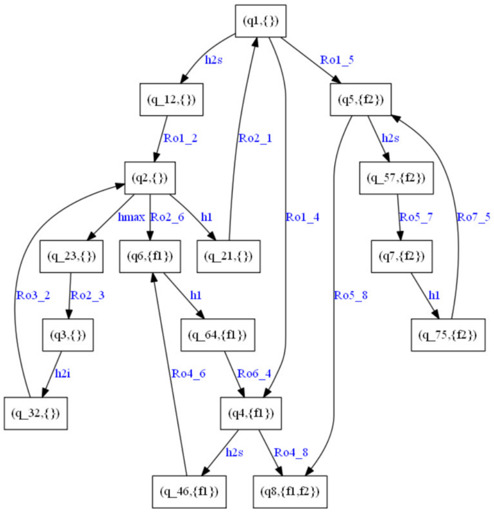


Figure 9. Diagnoser state tracker

Two fault scenarios have been simulated. In the Scenario 1, fault f_1 on the pump P_1 was injected after 3500h, fault f_2 is not injected. In the Scenario 2, fault f_2 is injected after 2000h, fault f_1 is not injected.

6.2.2. Scenario 1

Figure 10 shows the diagnoser belief state for Scenario 1 just before and after the fault f_1 occurrence. Results are consistent with the scenario: before 3500h, the belief states of the diagnoser are always tagged with a nominal diagnosis. After 3500h, all the states are tagged with f_1.

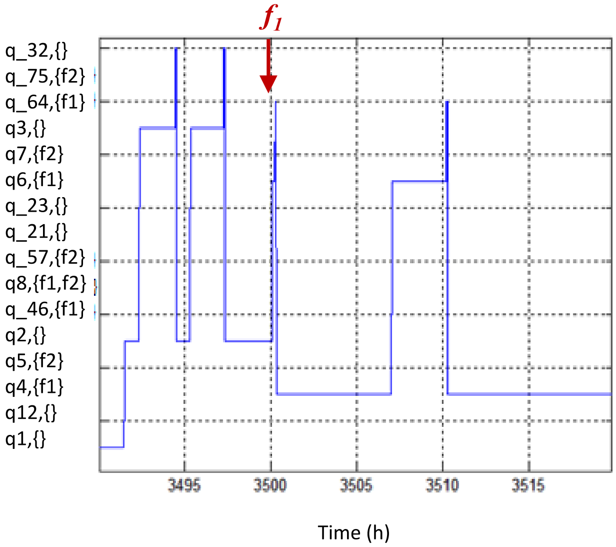


Figure 10. Diagnoser belief state for scenario 1

Figure 11 illustrates the predicted date of fault occurrence (d_{f_1} and d_{f_2}). At the beginning of the process, the prognosis result is: $\Pi_0 = (\{f_1, 4120\}, \{f_2, 5105\})$. It can be noted that the predicted dates d_{f_1} and d_{f_2} of f_1 and f_2 globally increase. Indeed, the system oscillates between stressful modes and less stressful modes. To make it simple, we can consider that in some modes, the system does not degrade, so the predicted dates of f_1 and f_2 are postponed.

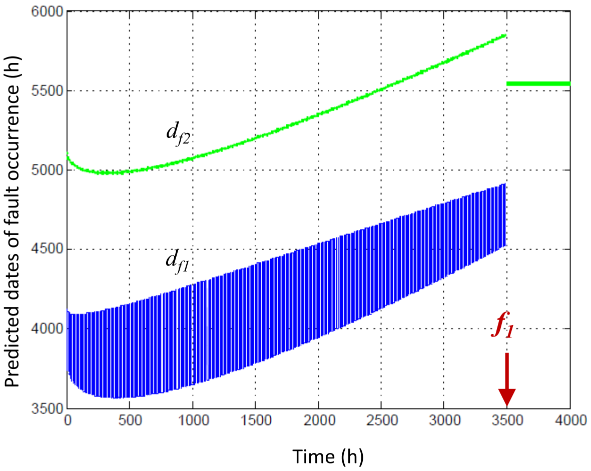


Figure 11. Prognosis results for scenario 1: d_{f_1}, d_{f_2}

Before 3500h, the predicted date of f_1 is lower than the one of f_2. After 3500h, the predicted date of f_2 is updated, knowing that the system is in a degraded mode. The prognosis result is $\Pi_{3501} = (\{f_2, 5541\})$.

Figure 12 shows the RUL of the system. Because of the oscillation of the system between stressful modes and less stressful modes, the RUL value oscillates between two values that globally decrease. The maximum bound corresponds to the less stressful mode whereas the minimum bound corresponds to the more stressful one. After the occurrence of the fault, the RUL is updated.

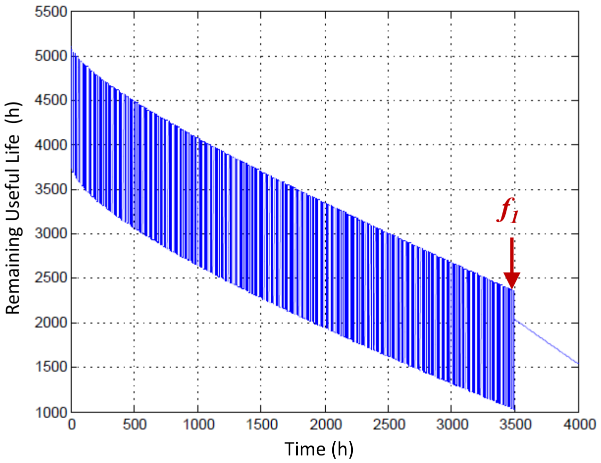


Figure 12. Prognosis results for scenario 1: RUL

6.2.3. Scenario 2

Figure 13 shows the diagnoser belief state for Scenario 2 just before and after the fault f_2 occurrence. Results are coherent with the Scenario: before 2000h, the belief states of the diagnoser are always tagged with a nominal diagnosis. After 2000h, all the states are tagged with f_2.

Figure 14 illustrates the predicted date of fault occurrence (d_{f_1} and d_{f_2}). At the beginning of the process, the prognosis result is: $\Pi_0 = (\{f_1, 4120\}, \{f_2, 5105\})$.

Before 2000h, the predicted date of f_1 is lower than the one of f_2. The prognosis result is After 2000h, the predicted date of f_1 is updated, knowing that the system is in a degraded mode. The prognosis result is $\Pi_{2010} = (\{f_1, 3947\})$.

Figure 15 shows the RUL of the system.

7. Conclusion and Future Work

This paper proposed an architecture of health monitoring and prognosis for hybrid systems. Diagnosis and prognosis are interleaved into a single process called *InterDP*. An original enriched hybrid model is proposed. It is used both for describing the hybrid behavior dynamic and for modeling the

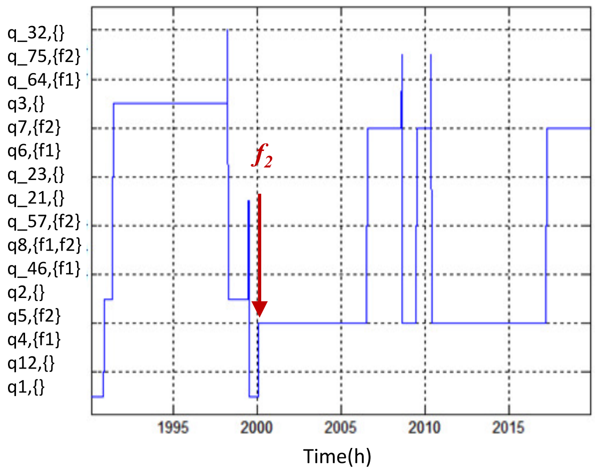


Figure 13. Diagnoser belief state for scenario 2

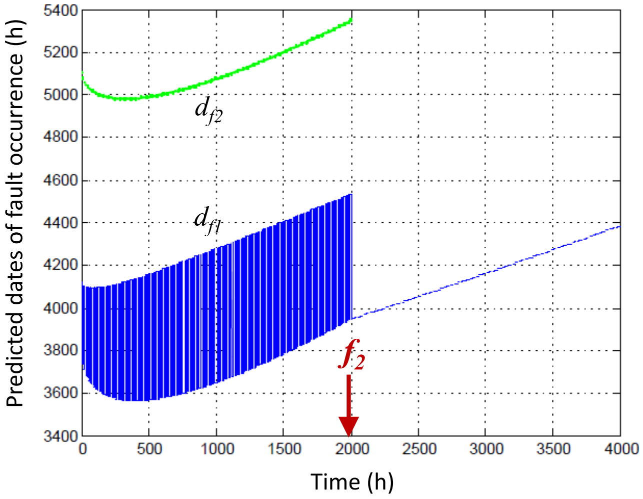


Figure 14. Prognosis results for scenario 2: d_{f_1}, d_{f_2}

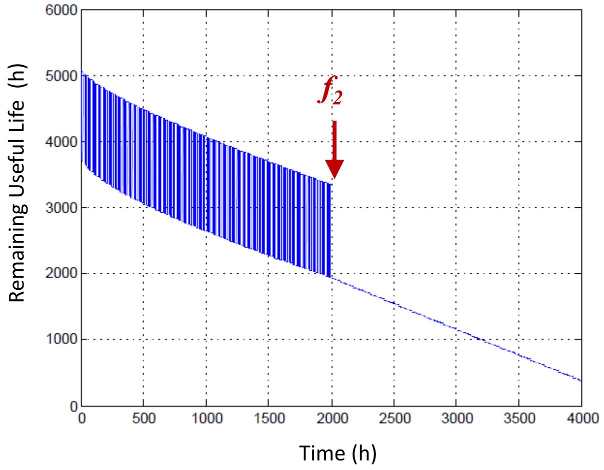


Figure 15. Prognosis results for scenario 2: RUL

degradation of the system. This paper is focused on the prognosis process whose algorithm is given in detail. An implementation has been done in Matlab and *InterDP* has been tested on a water tank system. Simulation results show how the prognosis is updated thanks to the diagnosis.

In future work, we will study the case of a non diagnosable system, where the diagnoser provides a set of belief states. This will detail the arrow from the prognoser to the diagnoser. It will lead to a complexity problem both for diagnosis and prognosis processes. If a fault sequence has to be predicted for each belief state, this could combinatorially explode. Another complexity problem concerns the number of anticipated faults in each behavioral state. For all these reasons, it seems rather complicated to fulfill the hypothesis of running both diagnosis and prognosis between two observations. A solution that will be developed is to do what we call ”any-time diagnosis”. It consists in elaborating a diagnosis in constrained time. The diagnoser gives its best result at the time it has to give a result. In the same time, the prognoser could also be improved by reducing the model size of the hybrid system, for example if only the most critical faults in each behavioral mode are taken into account.

Another perspective is a study on the model itself. A more detailed analysis of how to get a sufficient amount of information for the enriched model is required. Taking into account uncertainties on the available knowledge is a solution that will be investigated. Finally, future work will apply this approach to real-word examples. We plan to compare our approach with some others from the literature.

References

Bayoudh, M. (2009). *Active diagnosis of hybrid systems guided by diagnosability properties - application to autonomous satellites*. Unpublished doctoral dissertation, Université de Toulouse, France.

Bayoudh, M., Travé-Massuyès, L., & Olive, X. (2008). Hybrid systems diagnosis by coupling continuous and discrete event techniques. In *Proceedings of the 17th IFAC World Congress* (pp. 7265–7270). Korea, South. doi: 10.3182/20080706-5-KR-1001.01229

Cao, X. (1989). The predictability of discrete event systems. *IEEE Transactions Automatic Control*, *34*(11), 1168–1171. doi: 10.1109/9.40745

Castaneda, G.-A. P., Aubry, J.-F., & Brinzei, N. (2010). DyReIA (Dynamic Reliability and Assessment). In *Proceedings of the 1st Workshop on DYnamic Aspects in DEpendability Models for Fault-Tolerant Systems (DYADEM-FTS).* Valencia, Spain. doi: 10.1145/1772630.1772642

Chanthery, E., & Ribot, P. (2013). An integrated framework for diagnosis and prognosis of hybrid systems. In *Pro-*

ceedings of the 3^{rd} *Workshop on Hybrid Autonomous System (HAS).* Roma, Italy.

Ferreiro, S., & Arnaiz, A. (2008). Prognosis based on probabilistic models and reliability analysis to improve aircraft maintenance. In *Proceedings of the International Conference on Prognostics and Health Management (PHM).* Denver, USA.

Genc, S., & Lafortune, S. (2009). Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, *45*(2), 301–311. doi: 10.1016/j.automatica.2008.06.022

Hall, P. L., & Strutt, J. E. (2003). Probabilistic physics-of-failure models for component reliabilities using monte carlo simulation and weibull analysis: a parametric study. *Reliability Engineering and System Safety*, *80*, 233–242. doi: 10.1016/S0951-8320(03)00032-2

Henzinger, T. (1996). The theory of hybrid automata. In *Proceedings of the* 11^{th} *Annual IEEE Symposium on Logic in Computer Science* (pp. 278–292). doi: 10.1109/LICS.1996.561342

Jéron, T., Marchand, H., Genc, S., & Lafortune, S. (2008). Predictability of sequence patterns in discrete event systems. In *Proceedings of the* 17^{th} *IFAC World Congress* (pp. 537–543). Korea, South. doi: 10.3182/20080706-5-KR-1001.00091

Khoumsi, A. (2009). Fault prognosis in real-time discrete event systems. In *Proceedings of the International Workshop on Principles of Diagnosis (DX)* (p. 259). Stockholm, Sweden.

Rausand, M., & Hoyland, A. (2004). *System reliability theory: models, statistical methods and applications*. Wiley. doi: 10.1002/9780470316900

Ribot, P., & Bensana, E. (2011). A generic adaptative prognostic function for heterogeneous multi-cmoponent systems: application to helicopters. In *Proceedings of the European Safety & Reliability Conference (ESREL).* Troyes, France. doi: 10.1201/b11433-53

Ribot, P., Pencolé, Y., & Combacau, M. (2009). Diagnosis and prognosis for the maintenance of complex systems. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC).* San Antonio, USA. doi: 10.1109/ICSMC.2009.5346718

Roychoudhury, I., & Daigle, M. (2011, October 4-7). An integrated model-based diagnostic and prognostic framework. In *Proceedings of the* 22^{nd} *International Workshop on Principle of Diagnosis (DX'11).* Murnau, Germany.

Sampath, M., Sengputa, R., Lafortune, S., Sinnamohideen, K., & Teneketsis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, *40*, 1555-1575. doi: 10.1109/9.412626

Vachtsevanos, G., Lewis, L., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. Wiley. doi: 10.1002/9780470117842.index

Zemouri, R., & Faure, J.-M. (2006, October 4-6). Diagnosis of discrete event system by stochastic timed automata. In *the IEEE International Conference on Control Applications* (pp. 1861–1866). Munich, Germany. doi: 10.1109/CACSD-CCA-ISIC.2006.4776924

An Efficient Model-based Diagnosis Engine for Hybrid Systems using Structural Model Decomposition

Anibal Bregon[1], Sriram Narasimhan[2], Indranil Roychoudhury[3], Matthew Daigle[4], and Belarmino Pulido[5]

[1,5] *Dept. of Computer Science, University of Valladolid, Spain*
{anibal,belar}@infor.uva.es

[2] *Univ. of California Santa Cruz, NASA Ames Research Center. Moffett Field, CA 94035, USA*
sriram.narasimhan-1@nasa.gov

[3] *SGT Inc., NASA Ames Research Center. Moffett Field, CA 94035, USA*
indranil.roychoudhury@nasa.gov

[4] *NASA Ames Research Center. Moffett Field, CA 94035, USA*
matthew.j.daigle@nasa.gov

Abstract

Complex hybrid systems are present in a large range of engineering applications, like mechanical systems, electrical circuits, and embedded computation systems. The behavior of these systems is made up of continuous and discrete event dynamics that increase the difficulties for accurate and timely online fault diagnosis. The Hybrid Diagnosis Engine (HyDE) architecture offers flexibility to the diagnosis application designer to choose the modeling paradigm and the reasoning algorithms. The HyDE architecture supports the use of multiple modeling paradigms at the component and system level. However, HyDE faces some problems regarding performance in terms of time and space complexity. This paper focuses on developing efficient model-based methodologies for online fault diagnosis in complex hybrid systems. To do this, we propose a diagnosis framework where structural model decomposition is integrated within the HyDE diagnosis framework to reduce the computational complexity associated with the fault diagnosis of hybrid systems. As a case study, we apply our approach to a diagnostic benchmark problem, the Advanced Diagnostics and Prognostics Testbed (ADAPT), using real data.

1. Introduction

Nowadays, complex hybrid systems are present in many engineering applications, from electrical circuits to embedded computation systems. Their behavior is made up of continuous and discrete event dynamics, making accurate and timely online fault diagnosis more difficult. This paper focuses on developing efficient model-based methodologies for online fault diagnosis in complex hybrid systems. Hybrid systems modeling and diagnosis have been approached by the DX community, and several proposals have been made based on hybrid modeling (Mosterman & Biswas, 1999), hybrid state estimation (Hofbaur & Williams, 2004), or a combination of on-line state tracking and residual evaluation (Benazera & Travé-Massuyès, 2009; Bayoudh et al., 2008). In all cases, the solution requires to somehow model and eventually fully or approximately estimate the set of possible states, and to diagnose the current set of consistent modes. A major restriction, however, is that each technique uses its own modeling paradigm and the reasoning algorithms implement a single strategy. This does not facilitate the generation of flexible, integrated, reasoning solutions by the inclusion of additional diagnosis strategies, thus restricting the diagnostic capabilities of the hybrid diagnoser.

In (Narasimhan & Brownston, 2007), the authors proposed a general framework for stochastic and hybrid model-based diagnosis called Hybrid Diagnosis Engine (HyDE). HyDE offers flexibility to the diagnosis application designer to choose the modeling paradigm and the

Anibal Bregon et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

reasoning algorithms. The HyDE architecture supports the use of multiple modeling paradigms at the component and system level. Several alternative algorithms are available for the various steps in diagnostic reasoning. This approach is extensible, with support for the addition of new modeling paradigms as well as diagnostic reasoning algorithms for existing or new modeling paradigms. However, HyDE faces some problems regarding performance in terms of space and time complexity.

Recently, we have proposed to use structural model decomposition for efficient fault diagnosis and prognosis in continuous systems (Bregon, Biswas, & Pulido, 2012; Daigle et al., 2011a, 2012). In (Roychoudhury et al., 2013), we generalized those ideas and proposed a common model decomposition framework, where we solve the model decomposition problems for three separate system health management tasks, namely, estimation (used for residual generation that is usually required for fault detection and fault identification), fault isolation, and prediction (used for fault prognostics). The basic idea of the approach is to partition the global system model into submodels based on the set of measurements. This way, we will have submodels that are smaller than the global system model, leading to efficiency improvements and potential for concurrent computation.

In this paper, we integrate structural model decomposition as in (Roychoudhury et al., 2013) within the HyDE diagnosis framework. Structural model decomposition is used to decompose the HyDE models, thus reducing the computational complexity associated with the fault diagnosis of hybrid systems. This work contributes in two different aspects. First, we propose an online diagnosis approach for hybrid systems where the system model is partitioned into submodels, which are implemented using the HyDE modeling framework. Then, the global diagnosis result is provided by the combination of the local diagnosis results corresponding to the submodels. Second, we apply our approach to a real system, the Advanced Diagnostics and Prognostics Testbed (ADAPT) with satisfactory results.

The rest of the paper is organized as follows. Section 2 presents the HyDE diagnosis framework. Section 3 discusses the basic ideas of structural model decomposition. Section 4 proposes an integrated framework where structural model decomposition is used to reduce HyDE's computational burden. Section 5 shows results for the case study. Section 6 reviews the related work and current approaches for hybrid systems fault diagnosis and structural model decomposition. Finally, Section 7 concludes the paper.

2. HyDE

HyDE (Hybrid Diagnosis Engine) (Narasimhan & Brownston, 2007) combines ideas from consistency-based, control-theory-based and stochastic diagnosis approaches to provide a general, flexible and extensible architecture for stochastic and hybrid diagnosis. HyDE supports the use of multiple modeling paradigms and is extensible to support new paradigms. HyDE also offers a library of algorithms to be used in the various steps of the diagnostic reasoning process. The key features of HyDE are:

- Diagnosis of multiple discrete faults.
- Support for hybrid models, including autonomous and commanded discrete switching.
- Support for stochastic models and stochastic reasoning.
- Capability for handling time delay in the propagation of fault effects.

Next we present the HyDE modeling approach and reasoning procedure.

2.1. HyDE Models

HyDE models have two parts, the transition model and the behavior model. The transition model describes the components that make up the system, the various operating modes of the system (including faulty ones), and the conditions for transitions between the operating modes. The behavior model specifies the behavior evolution and has three parts: the propagation model, integration model, and dependency model. The information in the propagation model allows the estimation of unknown variable values from known variable values. The dependency model captures information about the dependencies between variables, models, and components. The integration model describes how the variables' values are propagated across time steps. HyDE supports the representation of each of the behavior models in more than one paradigm.

2.2. HyDE Reasoning

HyDE reasoning is the maintenance of a set K of weighted candidates (k_i, w_i). A candidate represents the hypothesized trajectory of the system inferred from the transition and behavior models, knowledge of the initial operating modes of all components and initial values of all variables, and the sensor observations reported to HyDE. The candidates' weights are a way of ranking them and depend on several factors, including prior probabilities of transitions and the degree of fit between model predictions and observations. Although weights are in the range [0, 1], weight is not a probability measure.

Each candidate contains a possible trajectory of system behavior evolution represented in the form of a hybrid state history and transition history. The hybrid state is a snapshot of the entire system state at any single instant. It associates all components with their current operating modes and all variables with their current values. Applications run HyDE at discrete time steps, typically but not necessarily when observations are available. Time steps need not be periodic. For each time step that HyDE reasons about, a candidate contains two hybrid states, one at the beginning of the time step and one at the end, as well as the set of transitions taken by the system between the previous and current time steps.

At time step 0, the candidate set is initialized with candidate(s) derived from the initial hybrid state of the system. Once the initial candidate set has been created, HyDE's reasoning process uses the same sequence of operations for each time step. The reasoning process can be divided into three categories of operations (Narasimhan & Brownston, 2007):

1. *Candidate Set Management* maintains the candidate set. The operations include updating the weights of all candidates, pruning candidates that do not satisfy minimum weight requirements, adding new candidates (the next best ones from the candidate generator) when necessary, and optionally re-sampling or normalizing the distribution of weights.
2. *Candidate Testing* deals with operations on a single candidate. The operations include determining the occurrence of any transitions, estimating the hybrid states at the beginning and end of a time step, comparing against observations to update weight of the candidate as well as reporting inconsistencies.
3. *Candidate Generation* creates candidate generators from inconsistencies reported by Candidate Testing and supplies the next-best potential (untested) candidate to Candidate Set Management when requested. This is achieved using a conflict directed search. First reported inconsistencies are used to generate conflicts, i.e., the subset of operating modes that cannot all be true at the same time. The conflicts are then used to guide a search for new candidates by optimizing some candidate property (typically weight or size).

As we have mentioned, the size of the system model (HyDE uses the global model of the complete system) directly affects the computational complexity for each one of the steps in the reasoning process. Our proposal on this work is to use structural model decomposition to divide the global system model into minimal submodels such that the complexity in the reasoning process is reduced. The next section describes our structural model decomposition approach to compute minimal submodels. Then, in Section 4 we will show in detail how these minimal submodels are integrated within the HyDE framework.

3. Structural Model Decomposition

In this section, we briefly present our structural model decomposition framework (Roychoudhury et al., 2013). We begin with the definition of a *model*.

Definition 1 (Model)**.** A *model* $\mathcal{M}$ is a tuple $\mathcal{M} = (V, C)$, where V is a set of variables, and C is set of constraints. V consists of five disjoint sets, namely, the set of state variables, X; the set of parameters, Θ; the set of inputs, U; the set of outputs, Y; and the set of auxiliary variables, A. Each constraint $c = (\varepsilon_c, V_c) \in C$ consists of an equation ε_c involving variables $V_c \in V$.

Input variables $u \in U$ are known/measured; and the output variables $y \in Y$ correspond to (measured) sensor signals. Parameters $\theta \in \Theta$ include explicit model parameters that are used in the model constraints. Θ does not need to include all parameters in the equations, only those that must be included explicitly (e.g., for joint state-parameter estimation or fault isolation). These parameters, by definition, are not computed in terms of any other variables, and, in this way, appear as inputs. Since the state variables X are, by definition, enough to describe the future behavior of the system, the auxiliary variables $a \in A$ are not strictly needed, however, they make the model easier to parse, develop, and implement.

As shown in Defn. 1, a constraint $c = (\varepsilon_c, V_c)$ includes an equation ε_c over the set of variables V_c. Note that c does not impose any computational causality on the variables V_c, i.e., although ε_c captures the information about how to compute a variable $v \in V_c$ in terms of all other variables in V_c, the constraint does not specify which $v \in V_c$ is the dependent variable in equation ε_c. We write a constraint $c_1 = (\varepsilon_{c_1}, V_{c_1})$ by its equation, e.g., as follows:

$$a + b = c + d \qquad (c_1)$$

where $V_{c_1} = \{a, b, c, d\}$.

In order to define for a constraint c which variable $v \in V_c$ is the dependent variable that is computed by the others using the constraint, we require the notion of a *causal assignment*.

Definition 2 (Causal Assignment)**.** A *causal assignment* α to a constraint $c = (\varepsilon_c, V_c)$ is a tuple $\alpha =$

(c, v_c^{out}), where $v_c^{out} \in V_c$ is assigned as the dependent variable in equation ε_c.

Unlike a constraint, a causal assignment defines a computational causality (or computational direction) to a particular variable $v_c^{out} \in V_c$ in the constraint in which it can be computed in terms of all other variables in V_c. We write a causal assignment of a constraint using the constraint's equation in a causal form. For example, for constraint c_1 above choosing $v_{c_1}^{out} = d$:

$$d := a + b - c \qquad (\alpha_1)$$

where Constraint c_1 is rewritten with a := symbol to explicitly denote that the direction of computation is from variables a, b, and c to d.

We say that a set of causal assignments $\mathcal{A}$, for a model $\mathcal{M}$ is *valid* if

- For all $v \in U \cup \Theta$, $\mathcal{A}$ does not contain any α such that $\alpha = (c, v)$, i.e., U and Θ are not computed in terms of any other variables.
- For all $v \in Y$, $\mathcal{A}$ does not contain any $\alpha = (c, v_c^{out})$ where $v \in V_c - \{v_c^{out}\}$, i.e., no variable is computed in terms of any $y \in Y$.
- For all $v \in V - U - \Theta$, $\mathcal{A}$ contains exactly one $\alpha = (c, v)$, i.e., other than the variables in U and Θ, every variable must have exactly one constraint to compute it.

A *causal model* is a model extended with a valid set of causal assignments.

Definition 3 (Causal Model). Given a model $\mathcal{M}^* = (V, C)$, a *causal model* for $\mathcal{M}^*$ is a tuple $\mathcal{M} = (V, C, \mathcal{A})$, where $\mathcal{A}$ is a set of valid causal assignments.

Given a model, we generate submodels that allow for the computation of a given set of variables using only local inputs. Given a definition of the local inputs (in general, selected from V) and the set of variables we wish to be computed by the submodel (selected from $V - U - \Theta$), we create from a causal model $\mathcal{M}$ a causal submodel $\mathcal{M}_i$. We obtain a submodel in which only a subset of the variables in V are computed using only a subset of the constraints in C. In this way, each submodel computes its variable values independently from all other submodels. A submodel can be defined as follows.

Definition 4 (Causal Submodel). A *causal submodel* $\mathcal{M}_i$ of a causal model $\mathcal{M} = (V, C, \mathcal{A})$ is a tuple $\mathcal{M}_i = (V_i, C_i, \mathcal{A}_i)$, where $V_i \subseteq V$, $C_i \subseteq C$, and $\mathcal{A}_i$ is a set of (valid) causal assignments for $\mathcal{M}_i$.

Note that, in general, $\mathcal{A}_i$ is not a subset of $\mathcal{A}$, because since we allow to select local inputs from Y, these variables become local inputs, i.e., appear in U_i, and the causal assignment in $\mathcal{A}$ that computes these variables is changed to a form where some other variable in the corresponding constraint is selected as the dependent variable. As a result, these causal assignments will be different, but the rest of the causal assignments in $\mathcal{A}_i$ will still be found in $\mathcal{A}$.

The procedure for generating a submodel from a causal model is given as Algorithm 1 (Roychoudhury et al., 2013). Given a causal model $\mathcal{M}$, a set of variables $U^* \supseteq U$ that includes the input variables in $\mathcal{M}$ as well as some other variables previously not in U that are considered as local inputs, and a set of variables to be computed V^*, and a preferences list, P (explained below), the `GenerateSubmodel` algorithm derives a causal submodel $\mathcal{M}_i$ that computes V^* using a subset of U^*.

In the following we briefly describe the algorithm, see (Roychoudhury et al., 2013) for additional details. In Algorithm 1, the queue, *variables*, represents the set of variables that have been added to the submodel but have not yet been resolved, i.e., they cannot yet be computed by the submodel. This queue is initialized to V^*, the set of variables that must be computed by the submodel. The algorithm then loops until this queue has been emptied, i.e., the submodel can compute all variables in V^* using only variables in U^*. Within the loop, the next variable v is popped off the queue. We then determine the best constraint to use to resolve this variable with the `GetBestConstraint` subroutine (Subroutine 2). We add the constraint to the submodel and the causal assignment for the constraint in the form that computes v. We then need to resolve all the variables being used to compute v, i.e., all its predecessors in the causal graph. Each of these variables that have not already been visited (not already in V_i), are not parameters (not in Θ), and are not local inputs (not in U^*) must be resolved and so are added to the queue. Then the variables are added to the submodel and the loop continues until the queue is emptied.

The goal of the `GetBestConstraint` subroutine is to find the best constraint to resolve v. The subroutine constructs a set C that is the set of constraints that can completely resolve the variable, i.e., resolves v without further backward propagation (all other variables involved in the constraint are in $V_i \cup \Theta \cup U^*$), and then chooses the best according to a preferences list P. If no such constraint exists, then the constraint that computes v in the current causal assignment is chosen, and further backward propagation will be necessary. Here, we are preferring minimal resolutions of v, i.e., those that do not require backward propagation, because then the submodel will be minimal in the number of variables and constraints needed to compute V^*.

Algorithm 1 $\mathcal{M}_i = \texttt{GenerateSubmodel}(\mathcal{M}, U^*, V^*, P)$

```
$V_i \leftarrow V^*$
$C_i \leftarrow \varnothing$
$\mathcal{A}_i \leftarrow \varnothing$
$variables \leftarrow V^*$
while $variables \neq \varnothing$ do
    $v \leftarrow \texttt{pop}(variables)$
    $c \leftarrow \texttt{GetBestConstraint}(v, V_i, U^*, \mathcal{A}, P)$
    $C_i \leftarrow C_i \cup \{c\}$
    $\mathcal{A}_i \leftarrow \mathcal{A}_i \cup \{(c, v)\}$
    for all $v' \in V_c$ do
        if $v' \notin V_i$ and $v' \notin \Theta$ and $v' \notin U^*$ then
            $variables \leftarrow variables \cup \{v'\}$
        end if
        $V_i \leftarrow V_i \cup \{v'\}$
    end for
end while
$\mathcal{M}_i \leftarrow (V_i, C_i, \mathcal{A}_i)$
```

In general, a variable v is involved in many constraints, however, exactly one of these constraints, in the given causal assignment, computes v. If this constraint does not completely resolve v, we find the constraints in which v is used to compute some output variable $y \in Y \cap U^*$. We consider modifying the causal assignment so that such a y (used now as an input) is used to compute v, instead of v being used to compute y. This can only be performed if, for the causal assignment in which y is being used to compute v, all other variables involved in the constraint are in $V_i \cup \Theta \cup U^*$, in which case this constraint in this new causal assignment can completely resolve v. If no constraint can be found that completely resolves v, then the constraint that in the current causal assignment computes v will have to be used, and backward propagation will be necessary. Otherwise, we select the most preferable constraint that completely resolves v. Preference among constraints (in which an output would be transformed to an input) is computed using a preferences list P, that contains a partial ordering of all the outputs in the model of the form $y_i \lhd y_j$, meaning that y_j is preferred over y_i. The subroutine goes through every pair of constraints and removes from the list of most preferable constraints, C', any constraint that uses a measured variable that is less preferable to one involved in another constraint. Of those remaining, an arbitrary choice is made. The preferences list can be used to prefer measured variables with less noise over those with more noise.

In the following sections, we show how this model decomposition approach can be integrated within the HyDE diagnosis framework to reduce the computational complexity associated with the diagnosis of faults in hybrid systems.

Subroutine 2 $c = \texttt{GetBestConstraint}(v, V_i, U^*, \mathcal{A}, P)$

```
$C \leftarrow \varnothing$
$c_v \leftarrow$ find $c$ where $(c, v) \in \mathcal{A}$
if $(V_{c_v} - v) \subseteq V_i \cup U^*$ then
    $C \leftarrow C \cup \{c_v\}$
end if
for all $y \in Y \cap U^*$ do
    $c_y \leftarrow$ find $c$ where $(c, y) \in \mathcal{A}$
    if $v \in V_{c_y}$ and $(V_{c_y} - v) \subseteq V_i \cup U^*$ then
        $C \leftarrow C \cup \{c_y\}$
    end if
end for
if $C = \varnothing$ then
    $c \leftarrow c_v$
else if $c_v \in C$ then
    $c \leftarrow c_v$
else
    $C' \leftarrow C$
    for all $c_1, c_2 \in C$ where $c_1 \neq c_2$ do
        $y_1 \leftarrow$ find $y$ where $(c_1, y_1) \in \mathcal{A}$
        $y_2 \leftarrow$ find $y$ where $(c_2, y_2) \in \mathcal{A}$
        if $(y_1 \lhd y_2) \in P$ then
            $C' \leftarrow C' - \{c_1\}$
        end if
    end for
    $c \leftarrow \texttt{first}(C')$
end if
```

4. Integration Proposal

The three main steps in the reasoning process of HyDE are simulation, comparison and candidate generation. These steps are performed for each currently consistent candidate in the candidate set. In this section, we show how the inclusion of structural model decomposition affects each one of these steps, thus proposing a framework where decomposed models can be implemented within HyDE.

In the simulation step, the behavior of the system is simulated using the global model of the system. The goal of the simulation step is to predict expected values of variables in the model that correspond to sensed observations. The main problem regarding this simulation step in HyDE is related to the time and memory performance of HyDE. Our proposal here is to use structural model decomposition so several smaller simulation tasks can be run. The advantage of using minimal submodels for simulation is its smaller size when compared to the size of the global model. However, as we will explain later, computing HyDE models from minimal submodels will affect the comparison and the candidate generation steps in the reasoning process of HyDE as well.

In order to implement minimal submodels in HyDE, we have to look at the models used by HyDE, which are similar to simulation models. They describe the expected behavior of the system under nominal and fault conditions. The model can be constructed in modular and hierarchical fashion by building component subsystem models (which may themselves contain component subsystem models) and linking them through shared variables/parameters. The component model is expressed as operating modes of the component and conditions for transitions between these various modes. Faults are modeled as transitions whose conditions for transitions are unknown (and have to be inferred through the reasoning process). Finally, the behavior of the components is expressed as a set of variables/parameters and relations governing the interaction among them (for example, equations). The relation between HyDE components and our structural decomposition framework is summarized as follows:

- HyDE model variables are related to variables V in our model.
- The propagation model is specified as constraint predicates over model variables. Constraints may be Boolean expressions if the variables are Boolean; algebraic and ordinary differential equations for interval- and real-valued variables, and equality or inequality for all variables. These are related to the constraints, C, and causal assignments, $\mathcal{A}$, in our model description.
- Candidates k_i in HyDE are related to parameters θ_i in our model.
- The integration model in HyDE is related to variables X in our model.

The comparison step then takes the predictions from the simulation step and the sensed observations and determines if they are consistent with each other or not. This step is performed only for those variables specified to be output variables (some sensed variables are designated inputs and will not be involved in the comparison step). Typically the percentage difference is compared to a threshold defined in the noise characteristics for each sensor specified when building the HyDE model. When HyDE is run without model decomposition only a subset of the sensed variables (those designated as output) are used in comparisons, while with minimal submodels all sensed variables will be used in comparisons. However this overhead is quite insignificant when compared to computational complexity of the simulation and candidate generation steps.

The third and final step is candidate generation, which is typically the most computationally intensive step. When the comparison step results in inconsistencies, a best first search is performed over the unknown transition space to identify potential candidates. When predicted values and sensed observations for a set of variables do not match, then all unknown transitions that could have influenced those inconsistent variables are considered suspects. There are two such flavors of dependencies. A component may have behavioral constraints in the current mode that affect the inconsistent variables and unknown transitions take the component to a different mode that influences the inconsistent variables in a different way. For this a dependency graph that maps dependencies between variables of the system through currently active behavioral constraints is generated. Back propagation through this graph starting from the inconsistent variable, identifies all suspected components. For each suspected component, all unknown transitions from the current mode of that component are selected as potential candidates. Among these transitions those that lead to component modes that influence the inconsistent variable(s) in the same way as the current component mode are eliminated.

The second flavor of influences are from components that do not affect the inconsistent variables in the current mode but have unknown transitions to modes that do influence the inconsistent variables. To identify such components a global dependency graph is generated that maps all dependencies in all modes of all components. Back propagation through this graph would then identify additional potential candidates that could possibly fix the inconsistencies.

When HyDE is used without model decomposition, the dependency graphs and candidate generation represent the entire model, which results in complexity that is exponential in the total number of unknown transitions that influence in the model. After model decomposition the HyDE model is decomposed into independent submodels each of which has its own dependency graph that is not connected to the other submodels. As a result, the candidate state space is significantly reduced. While this approach works for nonsensor faults, sensor faults pose a problem when using a decomposed model. Since a sensed observation can be used as input in other submodels a sensor fault would result in inconsistent variables in all of the submodels involving the sensor as an input or an output. In such cases we need a mechanism to report a single sensor fault instead of a fault from each submodel.

Such a mechanism is implemented in HyDE by representing the sensor as a single component. However inside the component there will be a variable for each submodel that the sensor appears in. When the sensor is used as an observation then its corresponding variable in the HyDE model is marked as an output variable, whereas if the observation is used as an input in the decomposition the corresponding variable is marked as an input variable in the HyDE model. The modes of the sensor component (that include nominal faulty modes) are shared by all of these variables. In other words these variables are connected to the rest of the variables in their submodels through independent behavioral constraints in the sensor component's modes. This would result in nonconnected dependency graphs but referring to shared component modes. As a result the back propagation would identify the shared component as a suspect.

Example 1. Consider a sensor component $S1$ with an associated variable $v1$ that appears in two submodels $M1$ and $M2$. In $M1$ it appears as an output variable $v1_o$ and in $M2$ it appears as input variable $v1_i$. Let the output variable associated with $M2$ be $v2$. When $S1$ is faulty then we will notice an inconsistency in the output $M1$ (the predicted value for $v1_o$ would be nominal, but because of the sensor fault, the observed value for $v1_0$ will not be consistent) as well as $M2$ (since we will simulate a faulty $v1$ value through $M2$, the predicted value for $v2$ will not match the observed value). The dependency graph associated with $M1$ will have edges going back from $v1_o$ to other variables represented in relations in $M1$. The edge to $v1_0$ (going back from $v1$) will be labeled as depending on $S1$ being in the nominal node (which is the current operating mode of $S1$). The dependency graph for $M2$ will go backwards from $v2$ and will ultimately reach $v1_i$ through relations represented in $M2$. In this case the edge out of $v1_i$ (going back into $v1_i$) would be labeled as depending on $S1$ too. In this case when we see $v1_o$ and $v2$ inconsistent, $S1$ will be selected as the most likely common explanation (unless there is another double fault with one component fault in $M1$ and another component fault in $M2$ that is more probable as defined by prior probabilities in the model). This example sensor component is illustrated in Fig. 1. The model inside sensor $v1$ is displayed below the $v1$ component for convenience. In the nominal and faulty modes of operation, there will be independent constraints relating $v1_{predictedo}$ with $v1_0$ and $v1_i$ with $v1_{predictedi}$. This will break the propagation path from $M1$ at $v1_o$ and start an independent propagation path from $v1_i$ to $M2$.

This approach allows us to gain the benefits of reduced computational complexity of the model decomposition without adding an additional diagnostic fusion step that might have been necessary if each submodel was completely independent.

5. Case Study

In this section we present our case study, a subset of the Advanced Diagnostics and Prognostics Testbed (ADAPT) (Poll et al., 2007), called ADAPT-Lite, which is an electrical power distribution system. We first briefly present the ADAPT-Lite system and then we show results that we obtained by using our integration approach.

5.1. ADAPT-Lite

A schematic of ADAPT-Lite is given in Fig. 2. Sensors prefixed with an "E" are voltage sensors, those with an "IT" are current sensors, and those with "ISH" or "ESH" are for states of circuit breakers and relays, respectively. TE228 is the battery temperature sensor, and ST516 is the fan speed sensor. Note that the inverter converts DC power to AC, and E265 and IT267 provide rms values of the AC waveforms. Here, v_B and i_B are the battery voltage and current, v_0 is the voltage across C_0, v_s is the voltage across C_s, e is the inverter efficiency, v_{inv} is the inverter voltage on the DC side, R_{inv} is the DC resistance of the inverter, R_{dc} is the DC load resistance, J_{fan} is the fan inertia, and B_{fan} is a damping parameter. Additional details on ADAPT-Lite may be found in (Daigle & Roychoudhury, 2010).

5.2. Diagnosis Results

For the case study we used test scenarios generated for the Diagnostic Competition 2011 (DXC 2011) (Poll et al., 2011). Specifically we used all of the 30 nominal scenarios and picked 66 fault scenarios that considered only discrete, abrupt and persistent faults. For these scenarios we ran the full HyDE model (we will call it HyDE) and the decomposed HyDE model (we will call it HyDE+SMD).[1]. Equations for the ADAPT model and its submodels can be found in (Daigle et al., 2011b). We then compared the diagnosis as well as the number of candidates that were tested before arriving at the diagnosis. For the nominal scenarios both models performed about the same with HyDE+SMD using less computational time. However this time saving was very insignificant (order of milliseconds). One of the reasons for this is that the full ADAPT model is relatively small and behavioral constraints were mostly algebraic.

Both models were tuned to not generate any false positives when run on the nominal scenarios. The results of running the faulty scenarios are presented in Table 1. Each row in the table represents a fault in ADAPT. Regarding the columns, the first column identifies the faulty component and the kind of fault; the second and third columns indicate the time of fault injection and its magnitude; the fourth (resp. seventh) column shows the HyDE (resp. HyDE+SMD) diagnosis result; the fifth (resp. eighth) column indicates the number of candidates that HyDE (resp. HyDE+SMD) needs to explore immediately after the fault detection; the sixth (resp. ninth) column shows the HyDE (resp. HyDE+SMD) classification errors (either a false positive or a false negative); finally, the tenth column shows the difference in the number of fault candidates considered for each one

[1] SMD stands for Structural Model Decomposition.

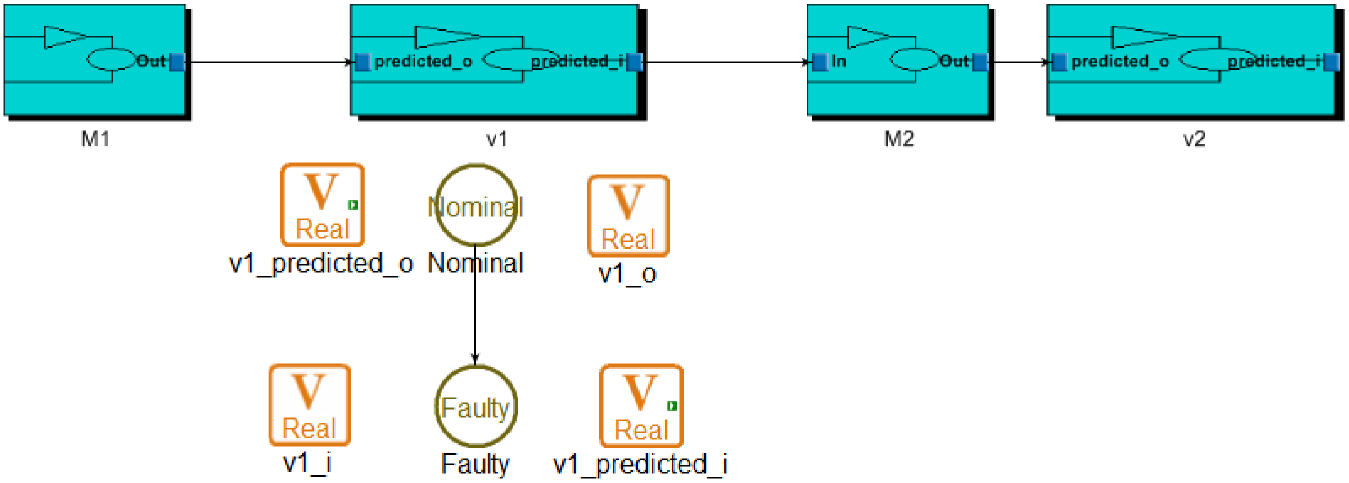


Figure 1. HyDE PC Sensor Model.

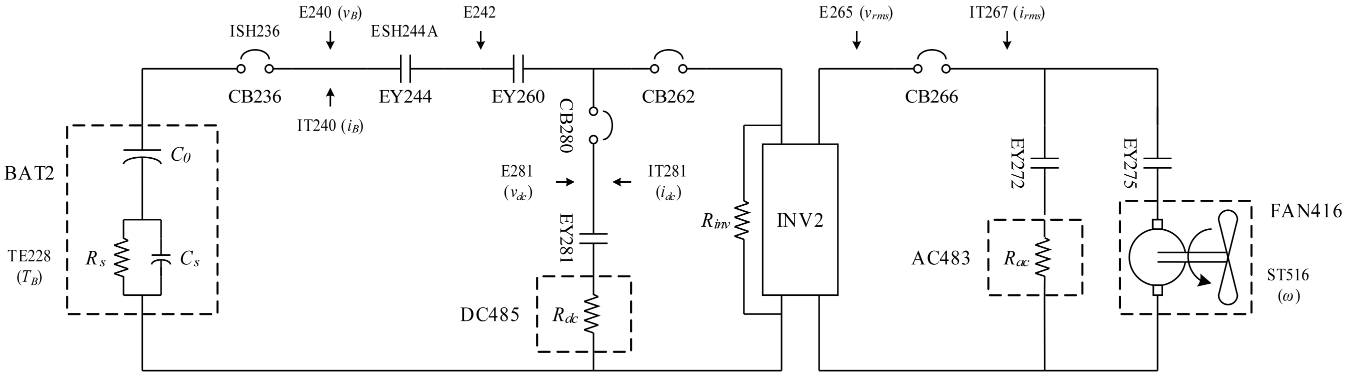


Figure 2. ADAPT-Lite schematic.

of the approaches. For an easier evaluation of the results obtained, Table 2 summarizes these results by giving the total number of candidates tried and classification errors for both of the approaches. Table 2 distinguishes between sensor and nonsensor faults.

Since the candidate generation takes a significant amount of time (order of seconds), the computational time can be considered to be directly proportional to the number of candidates tested. From the results we can see that there are two main advantages from combining HyDE with structural model decomposition.

First we see that the number of errors is reduced from 19 to 11. The reason for this will be apparent when we see how the simulation step is performed in the two cases. When only HyDE is used, the full model is simulated and any errors introduced because of model approximations (parameters in the model are estimated from data and are based on the best fit available and hence are approximate) get propagated through the model and accumulate. As a result at the comparison step some variables are incorrectly determined to be inconsistent when they are not (false positives). This problem can be addressed by increasing the threshold used for comparison but that would lead to some valid inconsistencies to not be detected at all (false negatives). When using HyDE+SMD this problem is substantially mitigated by the fact that simulation results (and any associated errors) do not get propagated to other submodels (instead the actual sensed input values are used). This results in more accurate predictions (assuming sensor values used as inputs are not too noisy) which leads to better diagnostic accuracy.

The second advantage is that fewer candidates are tested in the candidate generation step. As shown in the results, a total of 277 candidates for sensor faults and 44 candidates for nonsensor faults are tested when using HyDE. On the other hand, when HyDE+SMD is used, a total number of 54 candidates are tested for sensor faults and 20 for candidate faults. The reason for this is that the candidate generation step does not have to

Table 1. Diagnosis Results

Fault	At Time	Magnitude	HyDE Diagnosis	HyDE Candidates Tried	HyDE errors	HyDE+SMD Diagnosis	HyDE+SMD Candidates Tried	HyDE+SMD errors	Difference in Candidates Tried
IT240.Offset	72.00	5.40	IT240.Offset	14	0	IT240.Offset	1	0	13
IT240.Offset	101.00	0.30	NONE	0	1	IT240.Offset	1	0	0
E242.Offset	158.00	−2.00	NONE	0	1	E242.Offset	1	0	0
IT240.Stuck	83.00	16.88	IT240.Stuck	15	0	IT240.Stuck	2	0	13
IT267.Offset	192.00	−0.20	NONE	0	1	NONE	0	1	0
IT281.Offset	101.00	1.80	IT281.Offset	7	0	IT281.Offset	1	0	6
ESH244A.Stuck	49.00	0.00	ESH244A.Stuck	2	0	ESH244A.Stuck	2	0	0
IT267.Offset	104.00	0.70	IT267.Offset	11	0	IT267.Offset	1	0	10
IT281.Offset	47.00	0.20	NONE	0	1	NONE	0	1	0
ST516.Offset	168.00	90.00	ST516.Offset	9	0	ST516.Offset	1	0	8
ST516.Offset	121.00	−30.00	NONE	0	1	NONE	0	1	0
ST516.Stuck	58.00	0.00	ST516.Stuck	10	0	ST516.Stuck	2	0	8
ISH236.Stuck	41.00	0.00	ISH236.Stuck	2	0	ISH236.Stuck	2	0	0
ST516.Offset	203.00	−300.00	ST516.Offset	9	0	ST516.Offset	1	0	8
E240.Stuck	102.00	23.90	NONE	0	1	E240.Offset	1	1	0
E242.Stuck	173.00	0.00	E242.Stuck	4	0	E242.Stuck	2	0	2
E265.Stuck	41.00	0.00	E265.Stuck	7	0	E265.Stuck	2	0	5
IT281.Offset	101.00	−0.70	NONE	0	1	IT281.Offset	1	0	0
ST516.Offset	112.00	240.00	ST516.Offset	9	0	ST516.Offset	1	0	8
IT267.Offset	174.00	0.10	NONE	0	1	NONE	0	1	0
E240.Offset	138.00	−5.10	E240.Offset	2	0	E240.Offset	1	0	1
IT267.Offset	187.00	−1.40	IT267.Offset	11	0	ERROR	3	1	8
IT267.Stuck	49.00	2.38	IT267.Stuck	12	0	IT267.Stuck	2	0	10
IT240.Offset	199.00	−1.70	IT240.Offset	14	0	IT240.Offset	1	0	13
IT281.Offset	132.00	−0.05	NONE	0	1	NONE	0	1	0
E281.Stuck	80.00	21.38	ERROR	6	1	E281.Stuck	2	0	4
IT240.Offset	69.00	−4.20	IT240.Offset	14	0	IT240.Offset	1	0	13
IT281.Stuck	152.00	0.00	ERROR	8	1	IT281.Stuck	1	0	7
TE228.Offset	175.00	5.00	TE228.Offset	1	0	TE228.Offset	1	0	0
E265.Offset	39.00	8.00	E265.Offset	6	0	E265.Offset	1	0	5
AC483.FailedOff	79.88	*N/A*	EY272.StuckOpen	1	1	EY272.StuckOpen	1	1	0
DC485.FailedOff	51.73	*N/A*	EY284.StuckOpen	1	1	EY284.StuckOpen	1	1	0
FAN416.FailedOff	87.92	*N/A*	FAN416.FailedOff	1	0	FAN416.FailedOff	1	0	0
INV2.FailedOff	167.99	*N/A*	INV2.FailedOff	1	0	INV2.FailedOff	1	0	0
CB236.FailedOpen	170.97	*N/A*	CB236.FailedOpen	1	0	CB236.FailedOpen	1	0	0
CB262.FailedOpen	188.72	*N/A*	CB262.FailedOpen	1	0	CB262.FailedOpen	1	0	0
CB266.FailedOpen	129.80	*N/A*	ERROR	13	1	CB266.FailedOpen	1	0	12
CB280.FailedOpen	135.03	*N/A*	CB280.FailedOpen	1	0	CB280.FailedOpen	1	0	0
EY244.StuckOpen	35.35	*N/A*	EY244.StuckOpen	1	0	EY244.StuckOpen	1	0	0
EY260.StuckOpen	176.83	*N/A*	EY260.StuckOpen	1	0	EY260.StuckOpen	1	0	0
EY272.StuckOpen	62.87	*N/A*	EY272.StuckOpen	1	0	EY272.StuckOpen	1	0	0
EY275.StuckOpen	141.90	*N/A*	EY275.StuckOpen	1	0	EY275.StuckOpen	1	0	0
EY284.StuckOpen	83.83	*N/A*	EY284.StuckOpen	1	0	EY284.StuckOpen	1	0	0
DC485.FailedOff	59.08	*N/A*	EY284.StuckOpen	1	1	EY284.StuckOpen	1	1	0
FAN416.FailedOff	105.22	*N/A*	FAN416.FailedOff	1	0	FAN416.FailedOff	1	0	0
INV2.FailedOff	120.70	*N/A*	INV2.FailedOff	1	0	INV2.FailedOff	1	0	0
CB236.FailedOpen	35.66	*N/A*	CB236.FailedOpen	1	0	CB236.FailedOpen	1	0	0
CB266.FailedOpen	60.89	*N/A*	ERROR	13	1	CB266.FailedOpen	1	0	12
EY260.StuckOpen	80.06	*N/A*	EY260.StuckOpen	1	0	EY260.StuckOpen	1	0	0
EY272.StuckOpen	39.27	*N/A*	EY272.StuckOpen	1	0	EY272.StuckOpen	1	0	0
ISH236.Stuck	46.00	0.00	ISH236.Stuck	2	0	ISH236.Stuck	2	0	0
ST516.Offset	187.00	−243.00	ST516.Offset	9	0	ST516.Offset	1	0	8
TE228.Offset	101.00	21.00	TE228.Offset	1	0	TE228.Offset	1	0	0
IT240.Offset	203.00	−2.30	IT240.Offset	14	0	IT240.Offset	1	0	13
ST516.Offset	188.00	420.00	ST516.Offset	9	0	ST516.Offset	1	0	8
IT281.Offset	99.00	1.70	IT281.Offset	7	0	IT281.Offset	1	0	6
IT267.Offset	163.00	0.20	NONE	0	1	IT267.Offset	1	0	0
IT267.Offset	146.00	−0.30	IT267.Offset	11	0	IT267.Offset	1	0	10
IT281.Stuck	140.00	0.00	ERROR	8	1	ERROR	2	1	6
IT240.Stuck	95.00	18.26	IT240.Stuck	15	0	IT240.Stuck	2	0	13
E242.Offset	138.00	−3.00	E242.Offset	3	0	E242.Offset	1	0	2
E281.Stuck	83.00	23.42	E281.Stuck	5	0	E281.Stuck	2	0	3
IT240.Offset	178.00	1.50	NONE	0	1	IT240.Offset	1	0	0
IT267.Offset	172.00	−2.00	IT267.Offset	11	0	IT267.Offset	1	0	10
ST516.Offset	131.00	−300.00	ST516.Offset	9	0	ST516.Offset	1	0	8

Table 2. Summary of Diagnosis Results

Kind of Fault	Sum of HyDE Candidates Tried	Sum of HyDE errors	Sum of HyDE+SMD Candidates Tried	Sum of HyDE+SMD errors
Nonsensor faults	44	5	20	3
Sensor faults	277	14	54	8

back propagate past submodel boundaries when using HyDE+SMD. To understand this further first we look at how the unknown transition probabilities are set up. All nonsensor faults are considered to have the same probability and have higher probabilities than sensor faults. Among sensor faults (we consider only offset and stuck) the offset fault is considered more probable than stuck fault. In the full HyDE model when we see some inconsistent variables all components upstream of the sensors have to be considered suspect. However in the case of HyDE+SMD all components upstream of the sensor only in that submodel have to be considered suspect. For sensor faults we see an even more marked improvement in performance because of the special mechanism used to represent sensors in HyDE+SMD. In this case when we see two submodels to have inconsistent variables, the first explanation is the sensor that appears as output in one and input in the other. In the HyDE case all nonsensor faults upstream have to be considered before the sensor fault is considered, resulting in more candidates being tested. For HyDE+SMD we notice that we always test 1 (if actual fault is offset) or 2 (if actual fault is stuck then offset is tested first and then stuck is selected) candidates only.

As examples we will consider one nonsensor fault (DC485 Failed) and one sensor fault (IT281 Offset). The HyDE and HyDE+SMD model fragments containing these two components are illustrated in Fig. 3 and Fig. 4. For the DC485 Failed scenario using only HyDE we see that IT281 and IT240 are inconsistent and HyDE identifies EY284, DC485, CB280, EY260, EY244 and CB236 as possible suspects (based on the intersection of what is upstream of IT240 and IT281). When EY284 is tested it is consistent (EY284 and DC485 failures cannot be distinguished because they do not have any sensors in between them). When using HyDE+SMD only IT281 is detected to be inconsistent and now only EY284 and DC485 are picked as suspects since only those 2 components are present in the submodel that contains IT281 as output. In this case also EY284 is tested first and found to be consistent (resulting in the same diagnostic error due to lack of diagnosability).

When we consider the IT281 Offset scenario, HyDE generates EY284, DC485, CB280, EY260, EY244, CB236 and IT281 as suspects. Since nonsensor faults have higher probability, it considers the 6 nonsensor faults first, but they do not provide consistent predictions. Finally IT281 Offset is selected as a candidate which results in consistency. When HyDE+SMD model is used, IT281 and IT240 are found to be inconsistent. In this case the only intersection when searching for suspects is the IT281 component. Testing the IT281 Offset (because it has higher probability than IT281 Stuck) results in consistency.

6. Related work

Hybrid systems diagnosis has been tackled in different ways. Approaches based in a pure DES following the proposition by (Sampath et al., 1995) model the system as a set of automata, one for each working mode, that tries to track the discrete state, while performing diagnosis as a state-estimation process (Hofbaur & Williams, 2004; Benazera & Travé-Massuyès, 2009). The obvious difference and advantage with HyDE is that it does not need to pre-enumerate modes because they are generated on the fly. Moreover it is not required to generate, track and confirm any potential new discrete state given the ability to track continuous behavior.

Decompositional approaches for continuous systems diagnosis, such as PCs (Pulido & Alonso-González, 2004), ARRs (Staroswiecki & Declerck, 1989), and MSOs (Krysander et al., 2008), have been extended for hybrid systems following somewhat the proposal by (Cocquempot et al., 2004), and their concept of parameterized ARRs (Bayoudh et al., 2009; Moya et al., 2012). The set of ARRs or PCs for any mode must be generated off-line, and the active PCs or ARRs must be derived on-line. The obvious disadvantage is the need to model every potential transition in terms of known or estimated system variables.

There is also the option to combine ARRs and hybrid mode tracking as in (Rienmuller et al., 2013). This work combines hybrid estate estimation which is based on activated or non-activated residuals derived from ARRs for the current system. As in previous approaches, the set of potential states must be taken into account and two different diagnosis processes must be done at the same time to avoid tracking multiple discrete modes.

To avoid enumeration of potential modes, approaches based on Hybrid Bond Graphs, HBGs, adapt the model of the current continuous state by activating/deactivating switching junctions in a Bond-Graph model, and quickly providing a valid causal assignment (Narasimhan & Biswas, 2007). That approach can be combined with system model decomposition such as PCs, in the Hybrid PCs approach, providing a set of subsystems that can track the continuous behavior, while adapting to mode changes thanks to the underlying hybrid bond-graph modeling (Bregon, Alonso, et al., 2012). These HBG based approaches avoid enumeration of modes, but are still linked to one kind of diagnosis algorithm.

Summarizing a main difference between HyDE and the mentioned approaches is that all of them are linked to

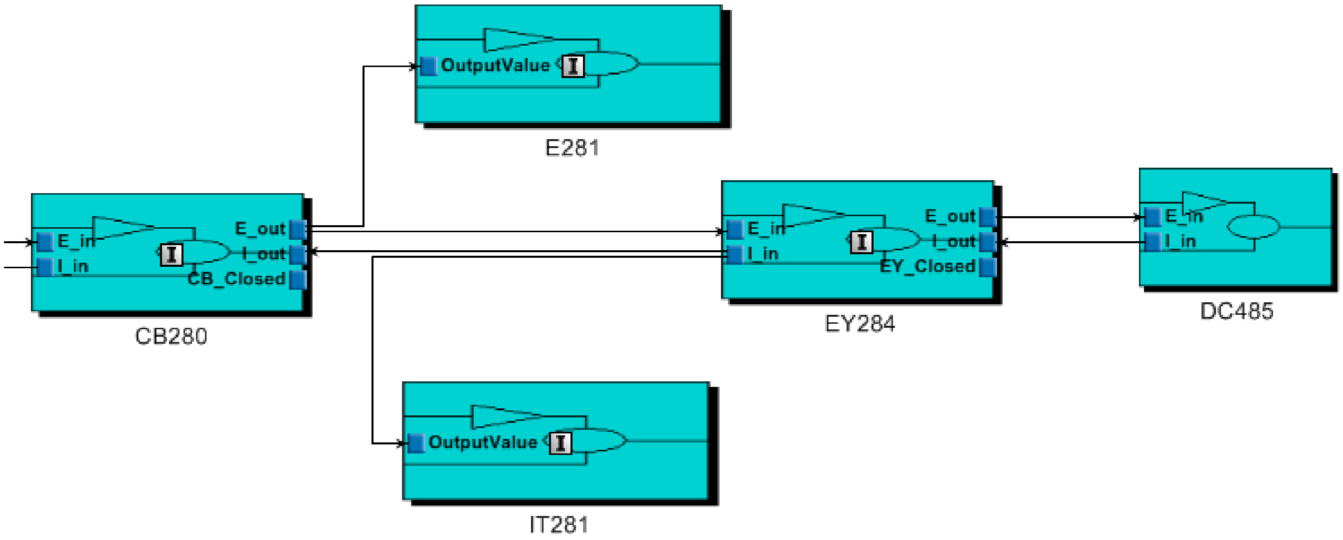


Figure 3. HyDE SMD Sensor Model.

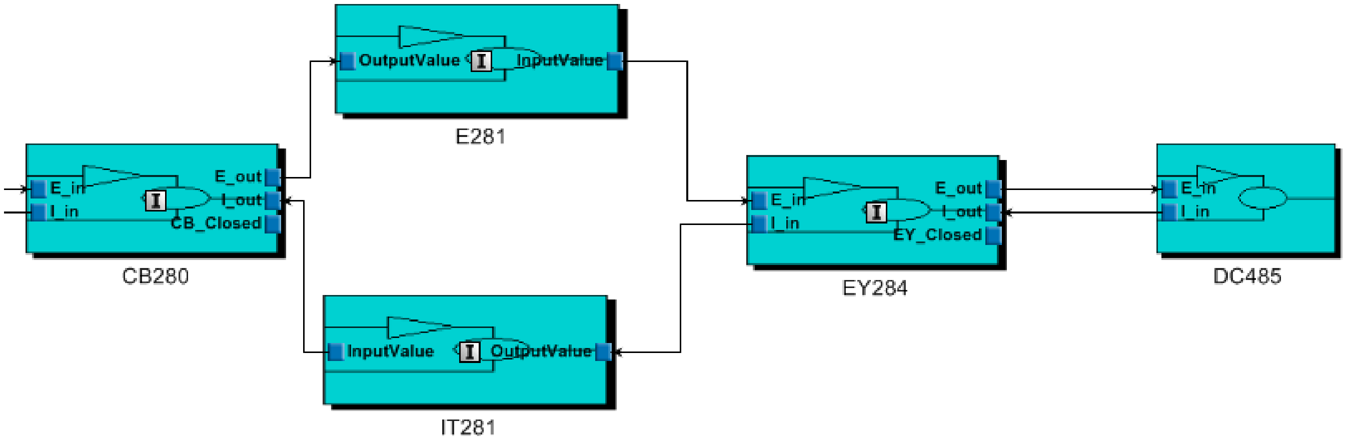


Figure 4. HyDE SMD Sensor Model.

one (or at most two) modeling paradigms, and integrates one diagnosis algorithm.

An implicit assumption in the integration of HyDE and structural model decomposition, due to the potential presence of output sensors as input in the submodels is that sensor noise should not be too high. This is an issue with all model decomposition approaches, because the additional introduction of noisy sensor values as input. This fact provokes sometimes a delay in the detection time, needing a longer period to be sure that the difference in the residual is not related to noise. But this is a common problem in almost any approach to model-based diagnosis, including those without model decomposition.

7. Conclusions

In this paper we presented a method of combining HyDE and structural model decomposition that lets us improve the performance of HyDE under assumptions that sensor noise is not too high. The combined approach results it better diagnosis accuracy as well as reduced computational complexity. We demonstrated this on an electrical testbed at NASA Ames Research Center that has published nominal and faulty data sets as part of the Diagnostic Competition series. In future work we would like to apply this method to other systems, more datasets, and further characterize the improvement in performance. Of particular interest would be multiple fault and increased sensor noise scenarios.

Acknowledgment

A. Bregon and B. Pulido's funding for this work was provided by the Spanish MCI TIN2009-11326 grant. S. Narasimhan, I. Roychoudhury and M. Daigle's funding for this work was provided by the NASA System-wide Safety and Assurance Technologies (SSAT) Project.

References

Bayoudh, M., Travé-Massuyès, L., & Olive, X. (2008). Coupling Continuous and Discrete Event System Techniques for Hybrid System Diagnosability Analysis. In *Proceeding of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence* (pp. 219–223). Amsterdam, The Netherlands, The Netherlands: IOS Press.

Bayoudh, M., Trave-Massuyes, L., & Olive, X. (2009). On-line analytic redundancy relations instantiation guided by component discrete-dynamics for a class of non-linear hybrid systems. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on* (p. 6970-6975).

Benazera, E., & Travé-Massuyès, L. (2009, October). Set-theoretic estimation of hybrid system configurations. *Trans. Sys. Man Cyber. Part B*, *39*, 1277–1291.

Bregon, A., Alonso, C., Biswas, G., Pulido, B., & Moya, N. (2012). Fault Diagnosis in Hybrid Systems using Possible Conficts. In *Proc. of Safeprocess'2012.* Mexico City, Mexico.

Bregon, A., Biswas, G., & Pulido, B. (2012, May). A Decomposition Method for Nonlinear Parameter Estimation in TRANSCEND. *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, *42*(3), 751-763.

Cocquempot, V., El Mezyani, T., & Staroswiecki, M. (2004, july). Fault detection and isolation for hybrid systems using structured parity residuals. In *Control Conference, 2004. 5th Asian* (Vol. 2, p. 1204 - 1212 Vol.2).

Daigle, M., Bregon, A., & Roychoudhury, I. (2011a, September). Distributed Damage Estimation for Prognostics Based on Structural Model Decomposition. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2011* (p. 198-208).

Daigle, M., Bregon, A., & Roychoudhury, I. (2011b, Oct). Qualitative Event-based Diagnosis with Possible Conflicts: Case Study on the Third International Diagnostic Competition. In *Proceedings of the 22nd International Workshop on Principles of Diagnosis* (p. 285-292). Murnau, Germany.

Daigle, M., Bregon, A., & Roychoudhury, I. (2012, September). A Distributed Approach to System-Level Prognostics. In *Annual Conference of the Prognostics and Health Management Society 2012* (p. 71-82).

Daigle, M., & Roychoudhury, I. (2010, October). Qualitative Event-based Diagnosis: Case Study on the Second International Diagnostic Competition. In *Proceedings of the 21st International Workshop on Principles of Diagnosis* (pp. 371–378).

Hofbaur, M., & Williams, B. (2004, oct.). Hybrid estimation of complex systems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *34*(5), 2178 -2191.

Krysander, M., Åslund, J., & Nyberg, M. (2008). An Efficient Algorithm for Finding Minimal Overconstrained Sub-systems for Model-based Diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics, Part A*, *38*(1).

Mosterman, P. J., & Biswas, G. (1999). Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, *29*(6), 554-565.

Moya, N., Pulido, B., Alonso-González, C., Bregon, A., & Rubio, D. (2012). Automatic generation of Dynamic Bayesian Networks for hybrid systems fault diagnosis. In *Proceeding of Intl. Workshop on principles of Diagnosis, DX, 2012.* Great Malvern, U.K..

Narasimhan, S., & Biswas, G. (2007, may). Model-Based Diagnosis of Hybrid Systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *37*(3), 348 -361.

Narasimhan, S., & Brownston, L. (2007, May 29-31). HyDE - A General Framework for Stochastic and Hybrid Model-based Diagnosis. In *Proceedings of the 18th International Workshop on Principles of Diagnosis, DX07* (p. 186-193). Nashville, TN, USA.

Poll, S., de Kleer, J., Abreau, R., Daigle, M., Feldman, A., Garcia, D., et al. (2011, October). Third International Diagnostics Competition – DXC'11. In *Proc. of the 22nd International Workshop on Principles of Diagnosis* (pp. 267–278).

Poll, S., et al. (2007, May). Evaluation, Selection, and Application of Model-Based Diagnosis Tools and Approaches. In *AIAA Infotech@Aerospace 2007 Conference and Exhibit.*

Pulido, B., & Alonso-González, C. (2004). Possible Conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics, Part B, Special Issue on Diagno-*

sis of Complex Systems, *34*(5), 2192-2206.

Rienmuller, T., Hofbaur, M., Trave-Massuyes, L., & Bayoudh, M. (2013, March). Mode set focused hybrid estimation. *International Journal of Applied Mathematics and Computer Science*, *23*(1), 131-144.

Roychoudhury, I., Daigle, M., Bregon, A., & Pulido, B. (2013, March). A Structural Model Decomposition Framework for Systems Health Management. In *Proceedings of the 2013 IEEE Aerospace Conference.*

Sampath, M., Sengputa, R., Lafortune, S., Sinnamohideen, K., & Teneketsis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*.

Staroswiecki, M., & Declerck, P. (1989, July). Analytical redundancy in nonlinear interconnected systems by means of structural analysis. In *IFAC Symp. on Advanced Information Processing in Automatic Control.*

Biographies

Anibal Bregon received his B.Sc., M.Sc., and Ph.D. degrees in Computer Science from the University of Valladolid, Spain, in 2005, 2007, and 2010, respectively. From September 2005 to June 2010, he was Graduate Research Assistant with the Intelligent Systems Group at the University of Valladolid, Spain. He has been visiting researcher at the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA; the Dept. of Electrical Engineering, Linkoping University, Linkoping, Sweden; and the Diagnostics and Prognostics Group, NASA Ames Research Center, Mountain View, CA, USA. Since September 2010, he has been Assistant Professor and Research Scientist at the Department of Computer Science from the University of Valladolid. Dr. Bregon is a member of the Prognostics and Health Management Society and the IEEE. His current research interests include model-based reasoning for diagnosis, prognostics, health-management, and distributed diagnosis and prognostics of complex physical systems.

Sriram Narasimhan is a Project Scientist with University of California, Santa Cruz working as a contractor at NASA Ames Research Center in the Discovery and Systems Health area. His research interests are in model-based diagnosis with a focus on hybrid and stochastic systems. He is the technical lead for the Hybrid Diagnosis Engine (HyDE) project. He received his M.S and Ph.D. in Electrical Engineering and Computer Science from Vanderbilt University. He also has a M.S in Economics from Birla Institute of Technology and Science.

Indranil Roychoudhury received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, Rajasthan, India in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, Tennessee, USA, in 2006 and 2009, respectively. Since August 2009, he has been with SGT, Inc., at NASA Ames Research Center as a Computer Scientist. Dr. Roychoudhury is a member of the Prognostics and Health Management Society and the IEEE. His research interests include hybrid systems modeling, model-based diagnostics and prognostics, distributed diagnostics and prognostics, and Bayesian diagnostics of complex physical systems.

Matthew Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. During the summers of 2006 and 2007, he was an intern with Mission Critical Technologies, Inc., at NASA Ames Research Center. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems. Dr. Daigle is a member of the Prognostics and Health Management Society and the IEEE.

Belarmino Pulido Belarmino Pulido received his Degree, MsC degree, and PhD degree in Computer Science from the University of Valladolid, Valladolid, Spain, in 1992, 1995, and 2001 respectively. In 1994 he joined the Departamento de Informatica in the University of Valladolid, where he is Associate Professor since 2002. He has been working on Model-based reasoning and Knowledge-based reasoning, and their application to Supervision and Diagnosis. He has worked in different national and European funded projects related to Supervision and Diagnosis. He is a member of the IEEE (M'2000), ACM (M'2003), and CAEPIA (1997, part of ECCAI) professional associations. He is also the coordinator of the Spanish Network on Supervision and Diagnosis of Complex Systems since 2005. Currently he is working on model-based diagnosis and prognosis of continuous and hybrid systems, using both centralized and distributed approaches.

SIL/HIL Replication of Electric Aircraft Powertrain Dynamics and Inner-Loop Control for V&V of System Health Management Routines

Brian Bole[1], Christopher Teubert[2], Cuong Chi Quach[3], Edward Hogge[4], Sixto Vazquez[5], Kai Goebel[6], George Vachtsevanos[7]

[1] *Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332*
bbole3@gatech.edu

[2] *SGT, Inc., NASA Ames Research Center, Moffett Field, CA, 94035*
Christopher.A.Teubert@nasa.gov

[3] *NASA Langley Research Center, Hampton, VA 23681*
cuong.c.quach@nasa.gov

[4] *Northrop Grumman Technical Services, Hampton, VA 23681*
edward.f.hogge@nasa.gov

[5] *NASA Langley Research Center, Hampton, VA 23681*
sixto.l.vazquez@nasa.gov

[6] *NASA Ames Research Center, Moffett Field, CA 94035*
kai.goebel@nasa.gov

[7] *Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332*
george.vachtsevanos@ece.gatech.edu

ABSTRACT

Software-in-the-loop and hardware-in-the-loop testing of failure prognostics and decision making tools for aircraft systems will facilitate more comprehensive and cost-effective testing than what is practical to conduct with flight tests. A framework is described for the offline recreation of dynamic loads on simulated or physical aircraft powertrain components based on a real-time simulation of airframe dynamics running on a flight simulator, an inner-loop flight control policy executed by either an autopilot routine or a human pilot, and a supervisory fault management control policy. The offline testing framework is described for the example of battery charge depletion failure scenarios onboard a prototype electric unmanned aerial vehicle.

1. INTRODUCTION

An early investment of resources into the development of an offline verification and validation (V&V) testing infrastructure for prognostics and supervisory health management algorithms is easily justified for complex systems in which online testing is substantially more time consuming and costly than offline testing. The V&V process is used to confirm that algorithms meet requirements and perform in a way that is consistent with stakeholder expectations. Flight tests prior to algorithm V&V can be dangerous to the vehicle, pilot, and ground crew. Offline tests to V&V algorithms in a laboratory setting prior to flight tests will not only improve flight test safety, but, as many issues can be resolved during offline tests, it reduces the number of real flight tests required for V&V, therefore reducing cost and development time.

Offline V&V tests of supervisory failure prognosis and decision making routines will allow developed supervisory algorithms to interact with onboard flight controllers and measured flight data exactly as they would during flight tests. The offline testing of health management algorithms may be con-

Brian Bole et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ducted using software-in-the-loop (SIL) or hardware-in-the-loop (HIL) procedures. SIL testing refers to tests conducted using only software simulations of system physics and embedded control routines. HIL testing refers to tests that include some hardware components from the target system.

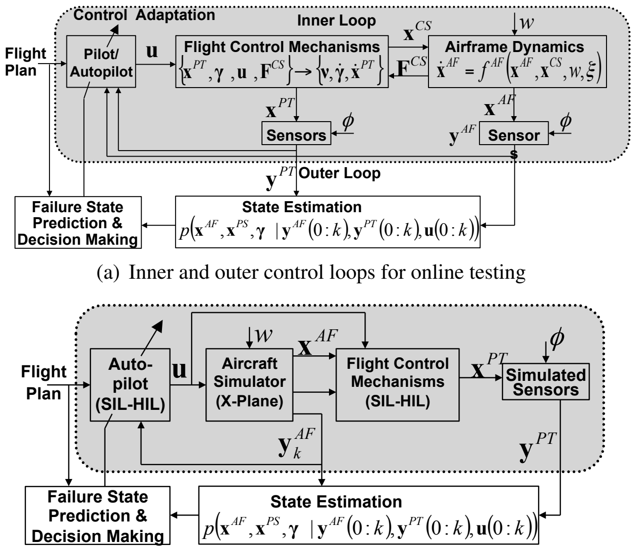
The SIL/HIL testing framework described in this paper uses the X-Plane[1] flight simulator package and an X-Plane Toolbox for MATLAB to facilitate prognostic based control algorithm V&V over a range of potential operating conductions. Examples of other offline testbeds making use of X-Plane for aerodynamics simulation and Matlab/Simulink softwares for simulation of control routines are found in (Ibeiro & Oliveira, 2010; Brown & Garcia, 2009; Sagoo et al., 2010). The SIL/HIL testbed presented in this paper improves present capabilities for performing offline testing with X-Plane aerodynamics simulations, by including a structure for simulating internal aircraft dynamics and component fault scenarios. The communications architecture developed to interface supervisory control routines running in MATLAB to SIL/HIL tests and an aerodynamics simulation running in X-Plane is intended to be distributed open-source in the near future.

When conducting V&V of supervisory control algorithms, injecting faults and testing to failure can provide valuable knowledge of the algorithm's behavior during potential failure scenarios. It is often not feasible to test to failure during flight tests without compromising the safety of the vehicle, onboard crew (for manned aircraft), or the ground crew. It is therefore valuable to have a method for the offline V&V of algorithm performance during failure scenarios. The offline V&V framework will also facilitate testing over a wide range of potential environmental conditions, including extreme conditions that are rarely encountered in practice. That said however, offline V&V testing is limited by the accuracy of SIL and HIL replications of nominal and off-nominal system dynamics, and flight testing is still a necessary part of the algorithm development and V&V process.

The general framework for SIL/HIL testing is described in Section 2. The development of an SIL/HIL simulation structure for the offline testing of battery charge management algorithms onboard an Edge-540T flight vehicle is presented in Section 3.

2. General Framework

This section introduces an SIL/HIL framework for testing component failure prognostics and real-time supervisory decision making algorithms that are intended to run onboard a flight vehicle. Block diagrams illustrating the structure of control loops used for online and offline flight testing are shown in Figure 1. The symbols used in Figure 1 and elsewhere in the paper are defined in the Nomenclature table at the end of the paper. An inner-loop controller is assumed to update flight control inputs based on a known flight plan and observations of the system state. The inner-loop controller is denoted by the 'Pilot/Autopilot' blocks in Figures 1(a) and 1(b). Failure prognostics and supervisory decision making operations are performed by an outer-loop process, denoted by the 'State Estimation', 'Failure State Prediction', and 'Decision Making' blocks in the figure. Both the inner-loop and outer-loop controllers would be unchanged in either online or offline flight testing.


Control Adaptation
Inner Loop
Flight Plan
Pilot/ Autopilot
u
Flight Control Mechanisms
$\{x^{PT}, \gamma, u, F^{CS}\} \rightarrow \{v, \dot{\gamma}, \dot{x}^{PT}\}$
x^{CS}
F^{CS}
w
Airframe Dynamics
$\dot{x}^{AF} = f^{AF}(x^{AF}, x^{CS}, w, \xi)$
x^{PT}
Sensors
ϕ
x^{AF}
y^{AF}
Sensor
ϕ
y^{PT}
Outer Loop
Failure State Prediction & Decision Making
State Estimation
$p(x^{AF}, x^{PS}, \gamma \mid y^{AF}(0:k), y^{PT}(0:k), u(0:k))$
Flight Plan
Auto-pilot (SIL-HIL)
u
w
Aircraft Simulator (X-Plane)
x^{AF}
Flight Control Mechanisms (SIL-HIL)
x^{PT}
ϕ
Simulated Sensors
y_k^{AF}
y^{PT}
Failure State Prediction & Decision Making
State Estimation
$p(x^{AF}, x^{PS}, \gamma \mid y^{AF}(0:k), y^{PT}(0:k), u(0:k))$

(a) Inner and outer control loops for online testing

(b) Inner and outer control loops for offline testing

Figure 1. Inner and outer control loops for online and offline controls testing

2.1. Inner-Loop Control Dynamics

Inner-loop control of the aircraft is assumed to be provided by either a human pilot or a pre-programmed autopilot. Both human pilot and autopilot will henceforth be referred to as just 'the pilot', for convenience. The pilot will update the control vector, $\boldsymbol{u}$, based on the observed states of the aircraft and a desired system state directed by a given flight plan.

The 'Flight Control Mechanisms' block shown in Figures 1(a) and 1(b) represents the internal electrical and mechanical dynamics of the vehicle's powertrain. The inner-loop control signals sent by the pilot, and the aerodynamic forces exerted on the vehicle's control surfaces by the surrounding environment, $\mathbf{F}^{CS}$, are inputs to this block. The $\mathbf{F}^{CS}$ vector consists of forces like the drag on the aircraft propeller, or the torque on control flaps. These inputs result in the loading of powertrian components, represented by the vector $\boldsymbol{\nu}$, which in turn determine the dynamics of powertrain component states, $\dot{\boldsymbol{x}}^{PT}$, and the dynamics of potential fault modes. The magnitudes of potential fault modes are represented here by the fault mode vector, $\boldsymbol{\gamma}$, where fault modes are assumed

[1] www.x-plane.com/

to be measurable indicators of component degradation such as crack length, spall width, or pitting depth.

The loads exerted by the vehicle's active components at a given time index, k, are expressed as a function of control input signals, the current states of powertrain components, and the states of component fault modes that may reduce component effectiveness,

$$\boldsymbol{\nu}\left(k\right) = f^{PT}\left(\boldsymbol{u}\left(k\right), \boldsymbol{x}^{PT}\left(k\right), \gamma\left(k\right), \boldsymbol{\xi}\left(k\right)\right) \tag{1}$$

where the additional $\boldsymbol{\xi}$ term is used to represent a vector of unknown or uncertain model parameters.

The 'Airframe Dynamics' block shown in Figure 1(a) represents the aerodynamic interactions between the vehicle airframe, vehicle control surfaces, and the operating environment. The inputs to this block are the current states of aircraft control surfaces, $\boldsymbol{x}^{CS}$, and the current state of the operating environment, $\boldsymbol{w}$. Environmental states represented by $\boldsymbol{w}$ may include atmospheric pressure, air temperature, wind speed, and turbulence. The current state of the airframe is represented by, $\boldsymbol{x}^{AF}$; it includes the position, heading, linear and rotational speed, and linear and rotational acceleration of the airframe in a set coordinate system.

Vehicle control surfaces are mechanically connected to powertrain components, so they should be a known function of $\boldsymbol{x}^{PT}$,

$$x^{CS} = f^{CS}\left(x^{PT}\right) \tag{2}$$

The forces exerted on the vehicle's control surfaces due to their motion through surrounding air is represented here as a generic non-linear function of the airframe state, the states of vehicle control surfaces, and current environmental states,

$$\boldsymbol{F}^{CS}\left(k\right) = f^{F^{CS}}\left(\boldsymbol{x}^{AF}\left(k\right), \boldsymbol{x}^{CS}\left(k\right), \boldsymbol{w}\left(k\right), \boldsymbol{\xi}\left(k\right)\right) \tag{3}$$

where the additional $\boldsymbol{\xi}$ term is used to represent a vector of unknown or uncertain model parameters.

Powertrain state dynamics and airframe dynamics are generically expressed in terms of the loading vectors $\boldsymbol{\nu}$ and $\boldsymbol{F}^{CS}$ as:

$$\dot{\boldsymbol{x}}^{PT}\left(k\right) = f^{PT}\left(\boldsymbol{x}^{PT}\left(k\right), \boldsymbol{\nu}\left(k\right), \boldsymbol{F}^{CS}\left(k\right), \boldsymbol{\xi}\left(k\right)\right) \tag{4}$$

$$\boldsymbol{y}^{PT}\left(k\right) = h^{PT}\left(\boldsymbol{x}^{PT}\left(k\right), \boldsymbol{\phi}\left(k\right)\right) \tag{5}$$

$$\dot{\boldsymbol{x}}^{AF}\left(k\right) = f^{AF}\left(\boldsymbol{x}^{AF}\left(k\right), \boldsymbol{x}^{CS}\left(k\right), \boldsymbol{w}\left(k\right), \boldsymbol{\xi}\left(k\right)\right) \tag{6}$$

$$\boldsymbol{y}^{AF}\left(k\right) = h^{AF}\left(\boldsymbol{x}^{AF}\left(k\right), \boldsymbol{\phi}\left(k\right)\right) \tag{7}$$

The progression of component health degradation is represented as:

$$\dot{\gamma}\left(k\right) = f^{\gamma}\left(\boldsymbol{x}^{PT}\left(k\right), \gamma\left(k\right), \boldsymbol{\nu}\left(k\right), \boldsymbol{\xi}\left(k\right)\right) \tag{8}$$

where component failure is considered to occur when fault magnitudes exceed a defined threshold that renders the component ineffective. The deterioration of control surfaces and electromechanical components in aircraft powertrains as a function mechanical loading forces has been a topic of study for some time; examples include: electromechanical actuators (Balaban et al., 2010) and composite wing structures (Gobbato et al., 2012), to name a few. The degradation and failure of electrical components as a function of electrical power loading has also been examined for aircraft components such as batteries (Saha et al., 2009) and power electronics (Celaya et al., 2011).

Measurements from simulated vehicle powertrain components, $\boldsymbol{y}^{PT}$, are generated in offline testing using models for both the underlying component dynamics, and sensor response dynamics. Data from actual system hardware is obtained in offline testing by applying mechanical loads to hardware components in real-time, in accordance with the environmental loads, $\mathbf{F}^{CS}$, reported in by the X-Plane simulator.

A hardware-only recreation of the 'Flight Control Mechanisms' portion of the inner-loop vehicle dynamics, illustrated in Figure 1(b), could be accomplished in a laboratory setting using an aircraft battery pack, power electronic motor/actuator drivers, electromechanical components, and associated interconnection cabling. Pilot controls could be sent directly to an electrical power distribution system assembled in the laboratory, and additional loading hardware could be used to apply mechanical loads to the electromechanical components of the powertrain in order to recreate the environmental loads estimated by the aircraft simulator. This approach is similar in nature to dynamometer testing commonly performed in the testing of automotive systems (Kelly et al., 2002; Tsang et al., 1985). Software models may be switched in for some or all of the hardware components in this setup; however, small errors in modeling the behavior of a given component may have outsized effects in observed system behavior over long time periods.

Measurements of the airframe states are represented by the vector, $\boldsymbol{y}^{AF}$. The X-Plane simulator, used in offline testing, includes sensor models to generate $\boldsymbol{y}^{AF}$, with the possibility of injecting sensor error and sensor noise. Communication between an autopilot board and the framework is facilitated by the open-source program APM Mission Planner[2]. The X-Plane Toolbox for MATLAB was used to communicate with APM Mission Planner, X-Plane, and outer-loop supervisory control routines running in MATLAB. The toolbox, currently being developed at NASA Ames Research Center, provides various MATLAB functions that allow for UDP communication with an associated X-Plane plug-in and APM Mission Planner. The team developing the toolbox intends to release it open-source upon completion.

Adequate control of aircraft does not in most cases require a pilot to understand environmental dynamics or the inter-

[2] http://code.google.com/p/ardupilot-mega/wiki/Mission

nal dynamics of the flight vehicle in great detail. Pilots are autopilots are considered here to make decisions based on an internal decision making policy that maps observations of $\boldsymbol{y}^{AF}(k)$ and $\boldsymbol{y}^{PT}(k)$ at time-index k to appropriate control outputs, $\boldsymbol{u}(k)$. An autopilot will use an embedded control policy to map $\left(\boldsymbol{y}^{AF}(k), \boldsymbol{y}^{PT}(k)\right) \rightarrow \boldsymbol{u}$. For human pilots, the mapping $\left(\boldsymbol{y}^{AF}(k), \boldsymbol{y}^{PT}(k)\right) \rightarrow \boldsymbol{u}$ will be determined by the pilot's situational awareness and judgment. The mechanism for interaction between an autopilot and supervisory failure prognostics and decision making routines can be for the decision making routines to directly update the autopilot's control policy. Policy updates for human pilots could be prompted indirectly by presenting the pilot with system health information and suggested risk mitigating actions, as described in (Bukov, Chernyshov, Kirk, & Schagaev, 2007).

2.2. Outer-loop Failure Prognostics and Decision Making

Supervisory outer-loop control routines make use of sensor measurements to estimate current and future system states given approximations of system state dynamics and physics of failure models. Probability distributions for belief in the current states of $\mathbf{x}^{PT}$, $\mathbf{x}^{AF}$, and γ, based on a history of observations of $\mathbf{y}^{PT}$, $\mathbf{y}^{AF}$, and $\mathbf{u}^{AF}$ are generically represented in Figure 1 as:

$$p\left(\mathbf{x}^{PT}, \mathbf{x}^{AF}\gamma | \mathbf{y}^{PT}(0:k), \mathbf{y}^{AF}(0:k), \mathbf{u}(0:k)\right) \quad (9)$$

Many Bayesian and machine learning methods have been published for the estimation of such probability distributions in the aviation domain (Lopez & Sarigul-Klijn, 2010; Napolitano et al., 1998).

Prediction of the evolution of future system states may be performed by propagating input uncertainty, model uncertainty, and state uncertainty forward in time. Estimates of component remaining useful life (RUL) are generated by extending predictions of the evolution of component loads and corresponding predictions of fault state evolution into the future until there is sufficient confidence in the occurrence of either component failure or completion of a prescribed flight plan. Particle filtering (Arulampalam et al., 2002), extended Kalman filtering (Ray & Tangirala, 1996), and Markov modeling (Guidaa & Pulcini, 2011) are examples of predictive filtering techniques used to propagate current state and model uncertainties forward in time.

Because inner-loop control policies may be modified by outer-loop supervisory control actions, the outer-loop prognostics and decision making routines could also be factored into the computation of future component load estimates. (Bole et al., 2012) describes the incorporation of outer-loop control policies into inner-loop fault growth predictions. The role of stochastic estimates of future loading in prognostic predictions is described in (Sankararaman et al., 2013; Tang et al., 2009). Stochastic beliefs about the manner in which the environment or system will evolve over time may be validated in offline simulations against repeated randomized simulations of flight scenarios.

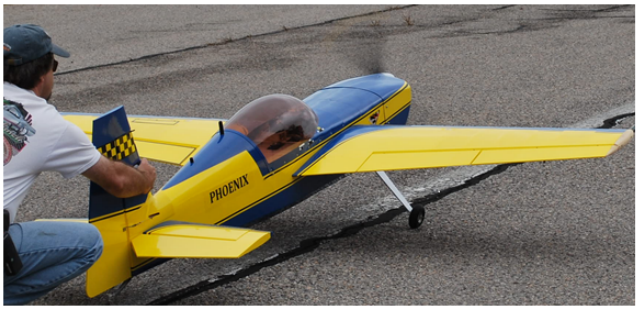


Figure 2. Edge-540T on runway

3. A Case Study: UAV Battery Charge Depletion Modeling

The implementation of the proposed SIL/HIL framework for the offline simulation of battery charge depletion onboard a prototype electric UAV platform is discussed here. The aircraft platform used for this case study is a commercial-off-the-shelf (COTS) 33% scale model of the Zivko Edge 540T airplane, pictured in Figure 2. The wingspan of the Edge 540T is 87 inches. The mass of the instrumented vehicle is approximately 44lbs. The propeller of the UAV is driven by two tandem mounted outrunner brushless DC motors that are each powered by a series connection of two lithium polymer battery packs. Each of the battery packs consist of five series connections of two 4.2V 3900mAh lithium polymer pouch cells wired in parallel.

The electrical and mechanical connections in the UAV powertrain are illustrated in Figure 3. Power flow from the battery packs to the driving motors is controlled by a Jeti 90 Pro Opto electric speed controller (ESC). The ESC sends synchronized voltages to the propeller motors at a duty cycle determined by a throttle input. The throttle input is either sent by remote control from a pilot, or by an onboard autopilot.

During both remote control and autonomous flight, a human pilot will maintain line of sight with the aircraft, and stand ready to execute a landing maneuver when the command is given by other operators on the ground. The gound operators assissting the pilot monitor the battery end-of-discharge prognostic estimates and decision making outputs generated in real-time by outer loop supervisory routines.

Charge estimation and end of charge prediction for UAV powertrain batteries have previously been examined in several publications by Bhaskar Saha at NASA ARC, Quach Chong Chi at NASA LaRC, and others (Saha, Quach, & Goebel, 2011; Saha, Koshimoto, et al., 2011). A separate battery system is used to power the data acquisition and other flight com-

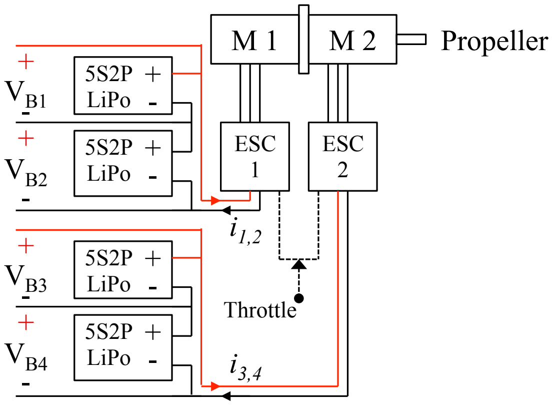


Figure 3. Electrical and mechanical connections of an Edge-540T UAV powertrain

munications and control hardware. The two battery systems are sized such that it is very likely that the batteries powering the propeller motors will be the first to be depleted. For that reason, onboard battery discharge prognostic algorithms and supervisory decision making actions are considered to only be concerned with the propeller driving batteries.

The offline replication of Edge 540T powertrain dynamics is described in three parts: inner-loop controls, battery demand modeling, and equivalent circuit battery modeling.

3.1. The Inner-Loop Controller

Vehicle flight plans are considered to be given in terms of an ordered set of 3D coordinates to be visited by the UAV, and a desired airspeed for making the translation from one waypoint to the next. Autonomous control of the Edge 540T is performed using an ArduPilot board. The ArduPilot sends control commands to the aircraft ESCs and flight control surfaces based on a set of proportional integral derivative (PID) control parameters that are tuned prior to flight, and periodic measurements of vehicle airspeed, heading, and position.

As was described in Section 2.1, the X-Plane simulator is used to simulate vehicle aerodynamics in offline simulations. Plane Maker, a design tool within the X-Plane package, was used to specify the aircraft mass, balance, and geometry for use in X-Plane aerodynamic simulations. There is some unavoidable error between the actual geometry, drag, and mass distribution of the aircraft and that used in the X-Plane aerodynamics models; however, because the control system is closed-loop small errors in simulating aircraft aerodynamics will not typically accumulate into large errors.

X-Plane can simulate various weather conditions and hardware configurations, and the ArduPilot can be tested with various flight plans. This configuration allows for the thorough testing of algorithm performance and safety before conducting flight tests.

3.2. Battery Demand Modeling

The proposed SIL/HIL testbed separates the simulation of aerodynamics and powertrain dynamics into two functional blocks. Connecting these two blocks requires that the airframe loads reported by the aerodynamics simulation be translated into loads on the system's powertrain components. It is difficult to collect direct measurements of airframe loads such as component forces and torques in flight. This measurement difficulty makes validating the load mapping used in offline simulations a complex proposition.

The tuning and validation of a propeller load mapping function is separated into two steps in this paper. First, a series of characterization experiments are performed in X-Plane to identify a nonlinear mapping between propeller output power and aircraft angle of climb, speed, and acceleration. Second, the modeled propeller power is mapped to a required battery power using a fixed power conversion efficiency coefficient and a proportional drag correction coefficient.

The nonlinear relationship between propeller output power and aircraft angel of climb, speed, and acceleration is observed for a flight simulator representation of an aircraft, by simulating a series of climbing and descending maneuvers at various angle of climb and throttle setpoints. Unlike actual flight tests, there is no difficulty in observing the precise loads on aircraft components in simulated flight tests. The results of repeated experiments at different throttle and angel of climb setpoints are used to fit a general set of aircraft aerodynamics and energy conservation equations, presented below.

The equations below are developed using the following assumptions: 1) the propeller is mounted on the aircraft nose; 2) the angle between the thrust vector generated by the propeller and the velocity vector of the aircraft is small; 3) Aircraft turning forces are small in comparison to the thrust and drag forces on the aircraft in its direction of travel.

The sum of the forces acting in the aircraft direction of travel is:

$$T_{x_w} = D(v) + m \cdot g \cdot \sin(\alpha) + m \cdot \dot{v} \tag{10}$$

where T_{x_w} represents the thrust produced by the aircraft in the direction of travel, D represents the drag force acting in the opposite direction of aircraft motion, v represents the aircraft speed, $\dot{v}$ represents acceleration, α represents angle of climb, m represents the vehicle mass, and g represents the earth's gravity.

The drag force on the airframe is represented by the following polynomial function of airspeed and angle of climb.

$$D(v, \gamma) = c_1 + c_2 \cdot v + c_3 \cdot v^2 + c_4 \cdot \alpha \tag{11}$$

Figure 4(a) shows a fit of the drag model to the averaged drag force reported by the X-Plane simulator over several steady speed climbing and descending maneuvers. The fitted parameter values are: $c_1 = 13.47, c_2 = -0.6, c_3 = 0.019, c_4 = 0.14$. During take-off and landing maneuvers when the aircraft speed is less than $15m/s$ the drag force is approximated as $D = 3 \cdot v$.

The measured and estimated propeller thrust versus airspeed is shown in Figure 4(b). The plot was generated using the thrust model given in Eq. (10), and the drag model given in Eq. (11).

The aircraft ESC throttle command is fit as a linear function of thrust and airspeed,

$$\text{Throttle} = a_1 + a_2 \cdot T_{x_w}(v, \dot{v}, \gamma) + a_3 \cdot v \tag{12}$$

where the fitted parameter values are: $a_1 = -19.64$, $a_2 = 0.95$, $a_3 = 1$.

Figure 4(c) shows the measured and estimated steady state ESC throttle commands observed in X-Plane simulations of various steady state climbing and descending maneuvers.

The product of thrust and airspeed gives the motive power exerted by the aircraft,

$$P_p = \frac{1}{\eta_p} \cdot T_{x_w} \cdot v \tag{13}$$

where P_p represents propeller output power and η_p represents the approximate propeller output power conversion efficiency.

Figure 4(d) shows the modeled propeller power and that reported by the X-Plane simulator for several steady speed climbing and descending maneuvers. The η_p parameter for the modeled aircraft is fit to $\eta_p = 0.7652$.

A fixed battery power conversion efficiency is assumed here for the aircraft motors and power electronics. Conversion between the net propeller output power and the net battery output power required to maintain a particular airspeed and angle of climb setpoint is achieved by applying a fixed power factor,

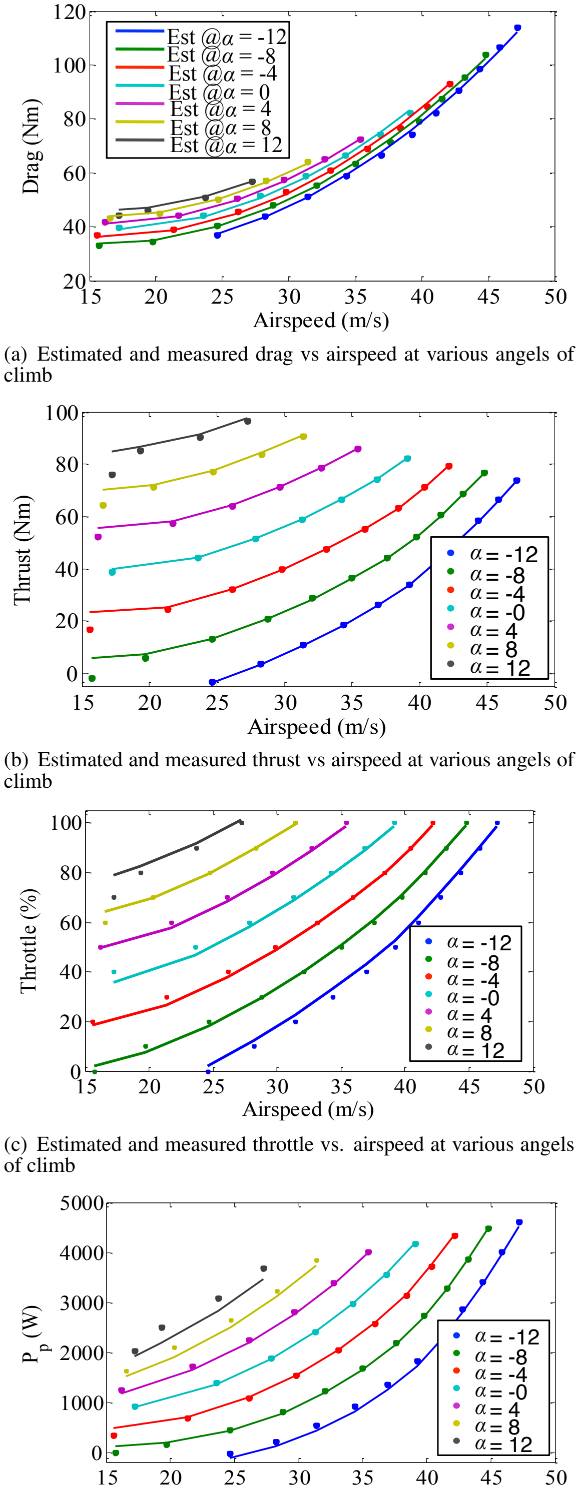
$$P_b = \eta_e \cdot P_p \tag{14}$$

where η_e represents power conversion efficiency and P_b represents net battery output power.

A proportional factor is also introduced here to correct for discrepancies between the drag model given in Eqn. 11, and the drag forces estimated for the actual aircraft. The corrective factor is expressed as:

$$D_A(v, \gamma) = \lambda_D \cdot D_M(v, \alpha) \tag{15}$$

where D_A and D_M represent the drag force estimated for the actual aircraft and drag force estimated for the X-Plane model respectively. λ_D represents a constant corrective factor that may be fitted by comparing modeled and actual aircraft powertrain load dynamics over sample flights.



(a) Estimated and measured drag vs airspeed at various angels of climb

(b) Estimated and measured thrust vs airspeed at various angels of climb

(c) Estimated and measured throttle vs. airspeed at various angels of climb

(d) Estimated and measured propeller power output vs airspeed at various angels of climb

Figure 4. Model fitting results for X-Plane flight load characterization tests

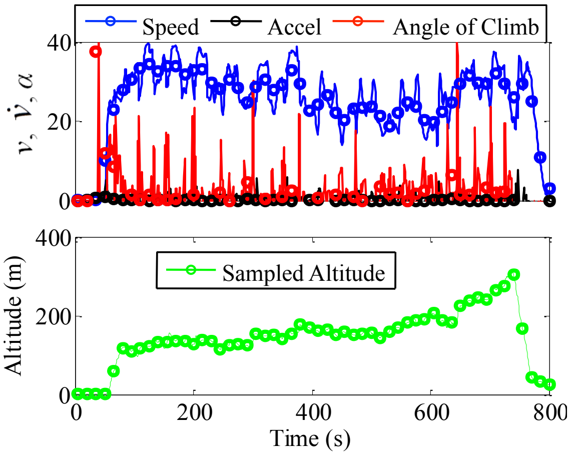


Figure 5. Approximate aircraft airspeed, acceleration, and angle of climb measurements derived from GPS samples

A roughly proportional deviation between the modeled and actual drag force is attributed to slight errors in modeling the aircraft geometry and surface aberrations. Small errors in modeling aircraft drag will cause only small effects on the aircraft handling from the perspective of a pilot or an autopilot, thus the drag correction need not necessarily be made for the SIL testing of inner-loop controllers. However, small errors in approximating loads on onboard energy storage devices will accumulate into large errors over a simulated flight.

Substitution of Eqns. (14) and (15) into Eqns. (10)-(13) yield the approximate battery power required to fly at a particular airspeed and angle of climb.

$$\begin{aligned} P_B &= \frac{1}{\eta_e \eta_p} \cdot T_{x_w} \cdot v \\ P_B &= \frac{v}{\eta_e \eta_p} \cdot \left(D_A(v,\alpha) + mg \cdot \sin(\alpha) + m\dot{v}\right) \\ P_B &= \frac{v}{\eta_e \eta_p} \cdot \left(\lambda_D D_M(v,\alpha) + mg \cdot \sin(\alpha) + m\dot{v}\right) \end{aligned} \tag{16}$$

Figure 5 shows approximate aircraft airspeed, acceleration, and angle of climb measurements derived from GPS samples, taken at 1 second intervals, over a sample aircraft flight. Figure 6 shows the predicted and measured battery power draw over the recoreded aircraft flight, using the average airspeed, acceleration, and angle of climb values falling within fifteen second intervals over the flight. The proportional constants η_e and λ_D used in Eq. 16 were fitted to $\eta_e = 0.85$ and $\lambda_D = 0.9$ to obtain the model fit shown in Figure 6.

The battery power predictions shown in Figure 6 are seen to lay on top of the observed power draw over the sample flight, aside from an apparent under prediction of battery power required during takeoff, which occurs from about 50 seconds to 100 seconds in the plots given. The under prediction of power required during takeoff could arise in part from the assumption that the angle between the trust vector and the velocity vector is small, which is not necessarily the case during takeoff. The battery power demand modeling used here also does not account for the fact that motor power conversion efficiency is typically very low during initial spin up.

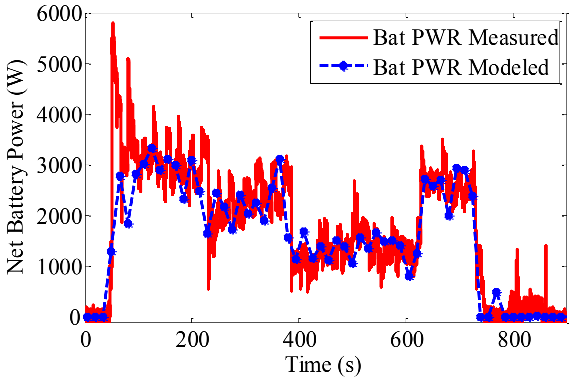


Figure 6. Measured and modeled battery power output

The net battery power, P_B, input to aircraft ESCs is given by the summation of battery power input to ESC1 and ESC2. The proportion of the net battery output powers that goes to each ESC is represented by:

$$\lambda_{ESC} = \frac{P_1 + P_2}{P_3 + P_4} \tag{17}$$

where λ_{ESC} represents the ratio of battery power drawn by each of the onboard ESCs.

The power output from the two strings of series connected battery packs is equal to the product of current and voltage,

$$P_B = \quad I_{1,2} \cdot (V_{B1} + V_{B2}) + I_{3,4} \cdot (V_{B3} + V_{B4}) \tag{18}$$

where V_{Bi} represents the terminal voltage of battery i, $I_{1,2}$ and $I_{3,4}$ represents the current flowing through the two sets of series connected batteries.

Substitution of Eq. (18) into Eq. (17), and solving for I gives:

$$\begin{aligned} I_{1,2} &= \frac{\lambda_{ESC} P_b}{(\lambda_{ESC}+1)\cdot(V_{B1}+V_{B2})} \\ I_{3,4} &= \frac{P_b}{(\lambda_{ESC}+1)\cdot(V_{B3}+V_{B4})} \end{aligned} \tag{19}$$

Figure 7 shows the observed ratio of battery power drawn from each of the onboard ESCs over a sample flight. The ratio of ESC power draw is currently uncontrolled, and it is seen to drift around a value of $\lambda_{ESC} \approx 0.7$ over the sample flight. The approximation for λ_{ESC} to be used in SIL and HIL testing of the vehicle powertrain may be improved in future work by incorporating possible dependencies on time, battery pack voltage, throttle command, and other inputs control inputs.

3.3. An Equivalent Circuit Battery Model

Battery voltage-current dynamics may be recreated over simulated flights in a laboratory by loading real or simulated batteries with a current indicative of flight loads. It should be

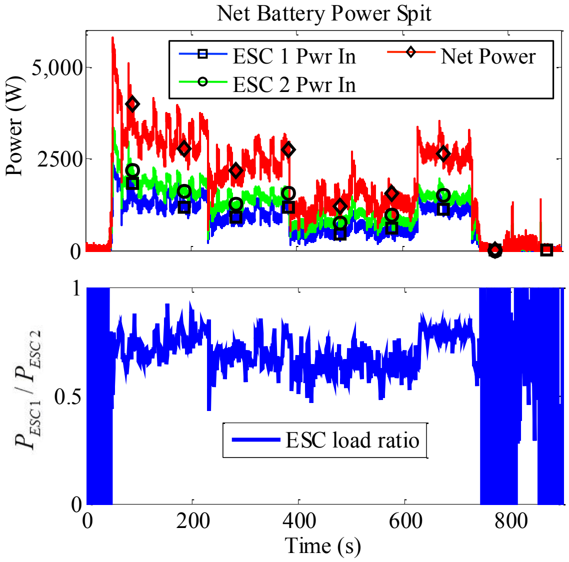


Figure 7. Measured battery power input to ESCs (Top) and observed ESC power ratio over a sample flight (bottom)

noted however that battery dynamics will vary substantially as a function battery health and temperature (Jossen, 2006). Differences in state of health and thermal loading of real and simulated batteries may cause results from SIL/HIL cycling of batteries in a laboratory to diverge from the observed battery dynamics in flight test.

Aircraft powertrain batteries are simulated in SIL testing using the equivalent circuit model shown in Figure 8. The equivalent circuit model used here is an extended version of the model explained in (Daigle, Saxena, & Goebel, 2012). This battery model uses six electrical components that are tuned to recreate the observed current-voltage dynamics of Edge 540T powertrain batteries. Battery charge is stored in the capacitor, C_b. The R_s, C_s and R_{cp}, C_{cp} circuit element pairs capture battery internal resistance drops and concentration polarization effects, respectively. The resistor R_p accounts for the slow battery self-discharge that is seen to occur over weeks or months of storage.

The current and voltage dynamics of the equivalent circuit model are defined as:

$$x^B = \begin{bmatrix} q_b & q_{cp} & q_{Cs} \end{bmatrix}^T \tag{20}$$

$$\dot{x}^B = \begin{bmatrix} -\frac{1}{C_b R_p} & \frac{1}{C_{cp} R_p} & \frac{1}{C_s R_p} \\ \frac{1}{C_b R_p} & -\frac{1}{C_{cp} R_p R_{cp}} & \frac{1}{C_s R_p} \\ \frac{1}{C_b R_p} & \frac{1}{C_{cp} R_p} & \frac{1}{C_s R_p} \end{bmatrix} x + \begin{bmatrix} i \\ i \\ i \end{bmatrix} \tag{21}$$

$$y^B = V_p = \begin{bmatrix} \frac{1}{C_b} & \frac{1}{C_{cp}} & \frac{1}{C_s} \end{bmatrix} \cdot x \tag{22}$$

where q_b, q_{cp}, and q_{cs} represent the charge stored in capacitors C_b, C_{cp}, and C_{cs} respectively. The total voltage drop across the battery terminals, V_p, is given by the sum of the voltage drops across the each of the three capacitors in the equivalent circuit model.

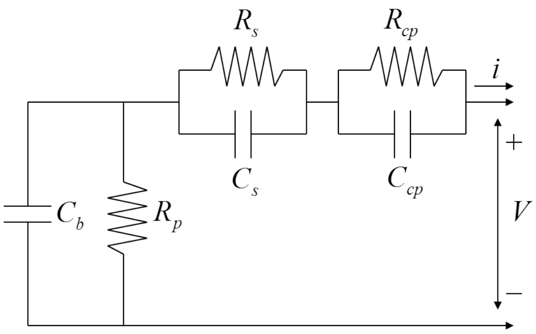


Figure 8. Equivalent circuit battery model

Because battery current-voltage dynamics are known to vary as a function of battery SOC, some of the resistive and capacitive (RC) components in the equivalent circuit model must be parameterized as functions of battery state of charge (SOC) (Zhang & Chow, 2010). It was decided based on qualitative observation that defining C_b, C_{cp}, and R_{cp} as parameterized functions of battery SOC gave an acceptable trade-off between the number of parameters to be identified and model error.

Battery SOC is defined as:

$$\text{SOC} = 1 - \frac{q_{max} - q_b}{C_{max}} \tag{23}$$

where q_b is the charge stored in the battery, q_{max} is the maximum charge of the battery, and C_{max} is the maximum charge that can be drawn from the battery. The term coulombic efficiency is used to refer to the portion of stored charge that can be withdrawn over repeated charge and discharge cycling of a battery. Resting a battery can temporarily unlock some of its lost charge storage capacity, however the overall trend is inevitably downward.

The C_b, C_{cp} and R_{cp} terms in the equivalent circuit battery model are parameterized as:

$$C_b = C_{Cb0} + C_{Cb1} \cdot \text{SOC} + C_{Cb2} \cdot \text{SOC}^2 + C_{Cb3} \cdot \text{SOC}^3 \tag{24}$$

$$C_{cp} = C_{cp0} + C_{cp1} \cdot \exp\left(C_{cp2}\left(1 - \text{SOC}\right)\right) \tag{25}$$

$$R_{cp} = R_{cp0} + R_{cp1} \cdot \exp\left(R_{cp2}\left(1 - \text{SOC}\right)\right) \tag{26}$$

Each battery pack used in Edge 540T flight tests should be characterized individually prior to testing, in order to account for any manufacturing and SOH variations. Two battery characterization experiments are used to identify the RC parameters in the battery equivalent circuit model.

The first experiment is a low current discharge of a battery from a fully charged state until a cutoff voltage of 17.5V is

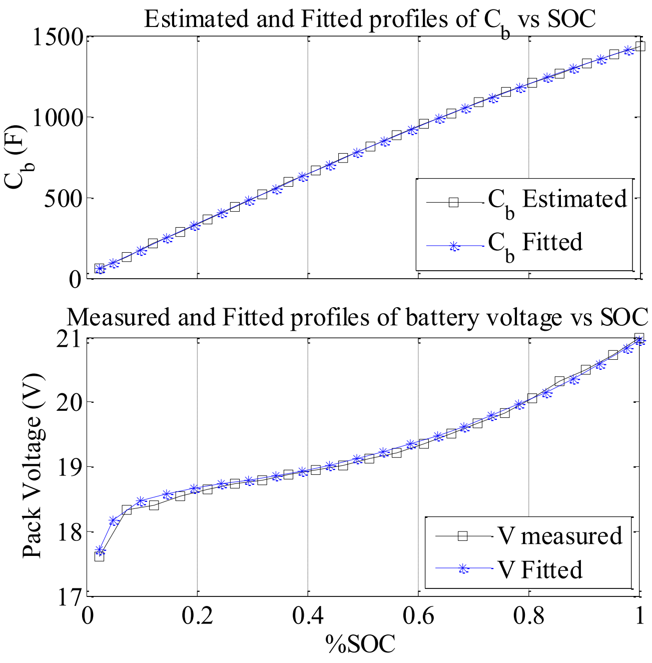


Figure 9. Measured and fitted profiles for C_b and battery voltage

reached. This type of discharge is mostly affected by the C_b, q_b, q_{max}, and C_{max} parameters in the model. Figure 9 shows a polynomial fit of C_b as a function of SOC, and the battery voltage fit for the tuned parameter values for C_{Cb0}, C_{Cb1}, C_{Cb2}, C_{Cb3}, q_{max}, and C_{max}.

Next, a pulsed loading experiment is used to fit the remaining parameters in the equivalent circuit model to the observed changes in battery hysteresis behavior as a function of SOC. A gradient descent search is used to identify the remaining model coefficients using the pulsed loading data. Figure 10 shows the battery voltage fit over a pulsed loading profile, using the tuned parameters identified in the low current experiment and the newly tuned values of R_s, C_s, R_{cp0}, R_{cp1}, R_{cp2}, C_{cp0}, C_{cp1}, and C_{cp2}. Values for all of the RC components and parameterization coefficients used in the equivalent circuit model of an Edge battery are defined in Table 1.

Observed battery loading over a piloted flight of the Edge 540T is shown in Figure 11. An asymmetric loading of the two propeller motors over the sample flight is apparent from the battery loads given in Figure 11. Motor M2 is known to consistently draw more current than motor M1 on the Edge 540, due to unregulated coupling of the two motor ESCs. Predicted and measured voltage profiles for batteries B1 and B3 using the recorded battery current profiles are shown in Figure 11. The close match between observed battery voltages and open-loop predictions over a given loading profile provides a measure of the validity of the software model.

The tuned battery model may be used to estimate the internal SOC of powertrain batteries based on sampled voltage and current data. The output of model based filtering approaches such as Kalman filtering will be much less susceptible to initialization and measurement errors than the Coulomb counting method currently used in many battery monitoring systems (Dai, Wei, & Sun, 2006).

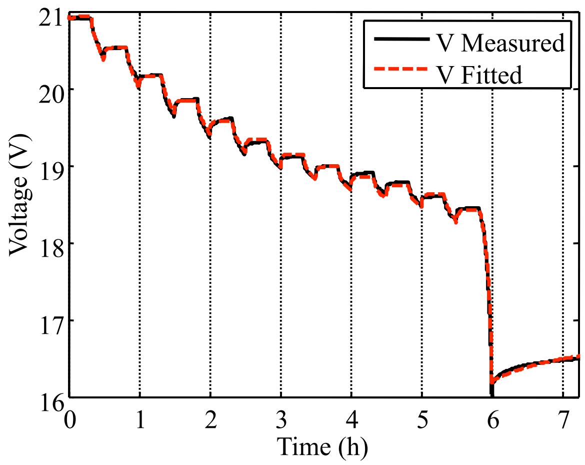


Figure 10. Measured and fitted profiles for battery voltage during pulsed loading

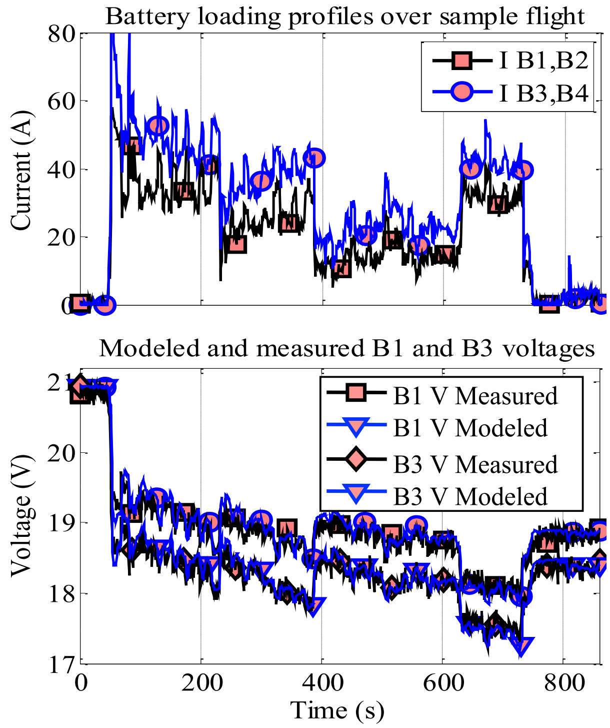


Figure 11. Modeled and measured voltages of batteries B1 and B3 for a sample flight loading profile

Table 1. Parameter values used in equivalent circuit model

Parameter	Value	Parameter	Value
q_{max}	2.88×10^4 C	C_s	89.3 F
C_{max}	2.85×10^4 C	R_{cp0}	$1.60 \times 10^{-3}\ \Omega$
C_{Cb0}	19.4 F	R_{cp1}	8.45
C_{Cb1}	1576 F	R_{cp2}	-61.9
C_{Cb2}	41.7 F	C_{cp0}	2689 F
C_{Cb3}	-203 F	C_{cp1}	$-2285\ F$
R_s	2.77×10^{-2}	C_{cp2}	-0.73 F

4. Conclusions

A framework is described for the offline recreation of dynamic loads on simulated or physical aircraft powertrain components based on a real-time simulation of airframe dynamics, an inner-loop flight control policy executed by either an autopilot routine or a human pilot, and a supervisory outer-loop control policy. The creation of an offline framework for verifying and validating supervisory outer-loop prognostics and decision making routines is described for the example of battery charge depletion failure scenarios onboard a prototype Edge 540T UAV with electric propulsion. The SIL/HIL testbed described in this paper is intended to be used to perform much more comprehensive and cost-effective testing of aircraft fault prognostics and decision making tools than would be practical to conduct in flight testing.

Acknowledgment

The project support by AvSafe/SSAT and OCT/ACLO are respectfully acknowledged.

Nomenclature

$\boldsymbol{x}^{AF}$	airframe state vector
$\boldsymbol{y}^{AF}$	observation of airframe state vector
$\boldsymbol{x}^{PT}$	electrical power dist. system state vector
$\boldsymbol{y}^{PT}$	observation of $\boldsymbol{x}^{PT}$ states
$\boldsymbol{u}$	pilot or autopilot control output vector
$\boldsymbol{\nu}$	mechanical loads on electromechanical components
$\boldsymbol{F^{CS}}$	net mechanical loads exerted by airframe
$\boldsymbol{w}$	environmental state parameter vector
γ	magnutude state vector for potential faults modes
$\boldsymbol{\xi}$	captures uncertainties in physics of failure models
$\boldsymbol{\phi}$	captures noise in sensor measurements

References

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174-188.

Balaban, E., Saxena, A., Narasimhan, S., Roychoudhury, I., Goebel, K., & Koopmans, M. (2010). Airborne electro-mechanical actuator test stand for development of prognostic health management systems. In *Annual conference of the prognostics and health management society.*

Bole, B., Goebel, K., & Vachtsevanos, G. (2012). Stochastic modeling of component fault growth over a derived domain of fiesible output control effort modifications. In *Annual conference of the prognostics and health management society.*

Brown, A., & Garcia, R. (2009). Concepts and validation of a small-scale rotorcraft proportional integral derivative (pid) controller in a unique simulation environment. *Unmanned Aircraft Systems*, *1*, 511-532.

Bukov, V., Chernyshov, V., Kirk, B., & Schagaev, I. (2007). Principle of active system safety for aviation: Challenges, supportive theory, implementation, application and future. In *ASTEC.*

Celaya, J., Kulkarni, C., Biswas, G., & Goebel, K. (2011). A model-based prognostics methodology for electrolytic capacitors based on electrical overstress accelerated aging. In *Proceedings of the annual conference of the PHM society.*

Dai, H., Wei, X., & Sun, Z. (2006). Online SOC estimation of high-power lithium-ion batteries used on HEVs. In *IEEE international conference on vehicular electronics and safety.*

Daigle, M., Saxena, A., & Goebel, K. (2012). An efficient deterministic approach to model-based prediction uncertainty. In *Annual conference of the prognostics and health management society.*

Gobbato, M., Conte, J., Kosmatka, J., & Farrar, C. (2012). A reliability-based framework for fatigue damage prognosis of composite aircraft structures. *Probabilistic Engineering Mechanics*, *29*, 176-188.

Guidaa, M., & Pulcini, G. (2011). A continuous-state Markov model for age- and state-dependent degradation processes. *Structural Safety*, *33*(6), 354-366.

Ibeiro, L., & Oliveira, N. M. (2010). UAV autopilot controllers test platform using MATLAB/Simulink and X-Plane. In *Frontiers in education conference (FIE).*

Jossen, A. (2006). Fundamentals of battery dynamics. *Journal of Power Sources*, *154*, 530-538.

Kelly, K., Mihalic, M., & Zolot, M. (2002). Battery usage and thermal performance of the toyota prius and honda insight during chassis dynamometer testing. In *Seventeenth annual IEEE battery conference on applications and advances.*

Lopez, I., & Sarigul-Klijn, N. (2010). A review of uncertainty in flight vehicle structural damage monitoring, diagnosis and control: Challenges and opportunities. *Progress in Aerospace Sciences*, *46*, 247-273.

Napolitano, M., Windon, D., Casanova, J., Innocenti, M., & Silvestri, G. (1998). Kalman filters and neural-network schemes for sensor validation in flight control systems. *IEEE Transactions on Control Systems Technologyy.*

Ray, A., & Tangirala, S. (1996). Stochastic modeling of fatigue crack dynamics for on-line failure prognostics. *IEEE Transactions on Control Systems Technology*, *4*, 443-451.

Sagoo, G., Gururajan, S., Seanor, B., Napolitano, M., Perhinschi, M., Gu, Y., & Campa, G. (2010). Evaluation of a fault-tolerant scheme in a six-degree-of-freedom motion flight simulator. *Journal of Aerospace Computing, Information, and Communication*, *7*, 47-67.

Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009). Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Transactions on Instrumentation and Measurement*, *58*(2), 291-296.

Saha, B., Koshimoto, E., Quach, C., Hogge, E., Strom, T., Hil, B., ... Goebel, K. (2011). Battery health management system for electric UAVs. In *IEEE aerospace conference.*

Saha, B., Quach, P., & Goebel, K. (2011). Exploring the model design space for battery health management. In *Annual conference of the prognostics and health managment society.*

Sankararaman, S., Daigle, M., Saxena, A., & Goebel, K. (2013). Analytical algorithms to quantify the uncertainty in remaining useful life prediction. In *IEEE aerospace conference.*

Tang, L., Kacprzynski, G., Goebel, K., & Vachtsevanos, G. (2009). Methodologies for uncertainty management in prognostics. In *IEEE aerospace conference.*

Tsang, P., Jacko, M., & Rhee, S. (1985). Comparison of chase and inertial brake dynamometer testing of automotive friction materials. *Wear*, *103*, 217-232.

Zhang, H., & Chow, M.-Y. (2010). Comprehensive dynamic battery modeling for PHEV applications. In *IEEE power and energy society general meeting.*

BIOGRAPHIES

Brian M. Bole graduated from the FSU-FAMU School of Engineering in 2008 with a B.S. in Electrical and Computer Engineering and a B.S. in Applied Math. Brian received a M.S. degree in Electrical Engineering from the Georgia Institute of Technology in 2011, and he is currently pursuing a Ph.D. Brian's research interests include: analysis of stochastic processes, risk analysis, and optimization of stochastic systems. Brian is currently investigating the use of risk management and stochastic optimization techniques for optimal adaptation of active component load allocations in robotic and aviation applications. In a previous project, Brian worked with the Georgia Tech EcoCar team to develop an energy management controller for optimizing the fuel economy of a charge sustaining hybrid electric vehicle.

Christopher Teubert received his B.S. in Aerospace Engineering from Iowa State University in 2012. While at Iowa State University, he conducted research on asteroid deflection mission design and asteroid fragment propagation for Iowa State University's Asteroid Deflection Research Center (ADRC). Previous to his current position he worked as a spacecraft systems engineer for a Mars sample return mission as part of the NASA Academy Program. He is currently researching systems and algorithms for diagnostics, prognostics, and system health management for Stinger Ghaffarian Technologies, Inc. at NASA Ames Research Center's Prognostic Center of Excellence (PCoE). He plans to begin pursuing a M.S. in 2014.

Quach Cuong Chi Cuong C. Quach got his M.S. from the School of Physics and Computer Sciences at Christopher Newport University in 1997. He is a staff researcher in the Safety Critical Avionics Systems Branch at NASA Langley Research Center. His research areas include development and testing of software for airframe diagnosis and strategic flight path conflict detection.

Edward Hogge received a B.S. in Physics from the College of William and Mary in 1977. He has provided engineering services to the government and currently is employed by Northrop Grumman Technical Services. He has recently been supporting aviation safety research through the implementation of electronic systems for subscale remotely piloted aircraft and through commercial aircraft simulation. He is a member of the American Institute of Aeronautics and Astronautics.

Sixto Vazquez obtained his MSEE degree from Old Dominion University in 1990 and BSEE from the University of Puerto Rico in 1983. He has developed real-time 3D graphical simulations to aid in the visualization and analysis of complex sensory data. Has developed techniques to interactively process, analyze, and integrate sensory data from multiple complex, state-of-the-art sensing technologies, i.e. FMCW Coherent Laser Radar range measuring system, Bragg grating Fiber Optic Strain Sensing system, etc., into simulation.

Kai Goebel received the degree of Diplom-Ingenieur from the Technische Universitt Mnchen, Germany in 1990. He received the M.S. and Ph.D. from the University of California at Berkeley in 1993 and 1996, respectively. Dr. Goebel is a senior scientist at NASA Ames Research Center where he leads the Diagnostics & Prognostics groups in the Intelligent Systems division. In addition, he directs the Prognostics Center of Excellence and he is the Associate Principal Investigator for Prognostics of NASAs Integrated Vehicle Health Management Program. He worked at General Electrics Corporate Research Center in Niskayuna, NY from 1997 to 2006 as a senior research scientist. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion. His research interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He holds eleven patents and has published more than 100 papers in the area of systems health management.

Vachtsevanos George is a Professor Emeritus of Electrical and Computer Engineering at the Georgia Institute of Technology. He was awarded a B.E.E. degree from the City College of New York in 1962, a M.E.E. degree from New York University in 1963 and the Ph.D. degree in Electrical Engineering from the City University of New York in 1970. He directs the Intelligent Control Systems laboratory at Georgia Tech where faculty and students are conducting research in intelligent control, neurotechnology and cardiotechnology, fault diagnosis and prognosis of large-scale dynamical systems and control technologies for Unmanned Aerial Vehicles. His work is funded by government agencies and industry. He has published over 240 technical papers and is a senior member of IEEE. Dr. Vachtsevanos was awarded the IEEE Control Systems Magazine Outstanding Paper Award for the years 2002-2003 (with L. Wills and B. Heck). He was also awarded the 2002-2003 Georgia Tech School of Electrical and Computer Engineering Distinguished Professor Award and the 2003-2004 Georgia Institute of Technology Outstanding Interdisciplinary Activities Award.

Why is the Remaining Useful Life Prediction Uncertain?

Shankar Sankararaman[1] and Kai Goebel[2]

[1] *SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA*
shankar.sankararaman@nasa.gov

[2] *NASA Ames Research Center, Moffett Field, CA 94035, USA*
kai.goebel@nasa.gov

ABSTRACT

This paper discusses the significance and interpretation of uncertainty in the remaining useful life (RUL) prediction of components used in several types of engineering applications, and answers certain fundamental questions such as “Why is the RUL prediction uncertain?”, “How to interpret the uncertainty in the RUL prediction?”, and “How to compute the uncertainty in the RUL prediction?”. Prognostics and RUL prediction are affected by various sources of uncertainty. In order to make meaningful prognostics-based decision-making, it is important to analyze how these sources of uncertainty affect the remaining useful life prediction, and thereby, compute the overall uncertainty in the remaining useful life prediction. The classical (frequentist) and Bayesian (subjective) interpretations of uncertainty and their implications on prognostics are explained, and it is argued that the Bayesian interpretation of uncertainty is more suitable for remaining useful life prediction in the context of condition-based monitoring. Finally, it is demonstrated that the calculation of uncertainty in remaining useful life can be posed as an uncertainty propagation problem, and the practical challenges involved in computing the uncertainty in the remaining useful life prediction are discussed.

1. INTRODUCTION

The prediction of remaining useful life (RUL) is an important functional aspect of an efficient prognostics and health management (PHM) system. The RUL prediction is not only necessary to verify if the mission goal(s) can be accomplished but also important to aid in online decision-making activities such as fault mitigation, mission replanning, etc. Since the prediction of RUL is critical to operations and decision-making, it is imperative that the RUL be estimated accurately. Since prognostics deals with predicting the future behavior of engineering systems, there are several sources of uncertainty which influence such future prediction, and therefore, it is rarely feasible to obtain an estimate of the RUL with complete precision. In fact, it is not even meaningful to make such predictions without computing the uncertainty associated with RUL. As a result, researchers have been developing different types of approaches for quantifying the uncertainty associated with the RUL prediction and prognostics in general.

Existing methods for quantifying uncertainty in prognostics and remaining useful life prediction can be broadly classified as being applicable to two different types of situations: offline prognostics and online prognostics. Methods for offline prognostics are based on rigorous testing before and/or after operating an engineering system, whereas methods for online prognostics are based on monitoring the performance of the engineering system during operation. For example, there are several research papers which discuss uncertainty quantification in crack growth analysis (Sankararaman, Ling, Shantz, & Mahadevan, 2011; Sankararaman, Ling, & Mahadevan, 2011), structural damage prognosis (Farrar & Lieven, 2007; Coppe, Haftka, Kim, & Yuan, 2010), electronics (Gu, Barker, & Pecht, 2007), and mechanical bearings (Liao, Zhao, & Guo, 2006), primarily in the context of offline testing. Engel et. al (Engel, Gilmartin, Bongort, & Hess, 2000) discuss several issues involved in the estimation of remaining useful life in online prognostics and health monitoring. While some of the initial studies on remaining useful life prediction lacked uncertainty measures (Celaya, Saxena, Kulkarni, Saha, & Goebel, 2012), researchers have recently started investigating the impact of uncertainty on estimating the remaining useful life. For example, there have been several efforts to quantify the uncertainty in remaining useful life of batteries (Saha & Goebel, 2008) and pneumatic valves (Daigle & Goebel, 2010) in the context of online health monitoring. Different types of sampling techniques (Daigle, Saxena, & Goebel, 2012)

Shankar Sankararaman et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

and analytical methods (Sankararaman, Daigle, Saxena, & Goebel, 2013) have been proposed to predict the uncertainty in the remaining useful life.

While the importance of uncertainty quantification in prognostics and RUL estimation have been widely understood, there have been few efforts to understand and appropriately interpret such uncertainty. Celaya et al. (Celaya, Saxena, & Goebel, 2012) discussed the interpretation of RUL in the context of Kalman filtering-based prognostics techniques, and explained that it is not approriate to arbitrarily force the variance of RUL to be small. It is necessary to further delve into this topic in order to completely analyze the importance and impact of uncertainty in prognostics.

This paper poses three fundamental questions in order to understand uncertainty in prognostics, particular in the context of remaining useful life (RUL) prediction:

1. Why is the RUL prediction uncertain?
2. How do we interpret the uncertainty in RUL?
3. How do we calculate the uncertainty in RUL?

The answers to the above questions are sought from multiple points of view. First, the topic of uncertainty in prognostics is discussed from a qualitative point of view in Section 2; the various sources of uncertainty in prognostics are discussed and the different activities related to uncertainty quantification and management are outlined. Second, the interpretation of uncertainty is discussed from a statistical point of view in Section 3. While statistics and probability methods have been in existence for over 200 years, there has always been a disagreement (amongst mathematicians and statisticians alike) regarding the interpretation of probability. It is important to understand this disagreement before attempting to interpret uncertainty in prognostics. Third, the interpretation of uncertainty in prognostics and RUL prediction is analyzed in detail in Section 4, and it is explained all different interpretations of probability may not be suitable for prognostics and health monitoring purposes. Fourth, it is demonstrated that calculating the uncertainty in RUL is, fundamentally, an uncertainty propagation problem and the challenges in this regard are outlined in Section 5. In this context, it is examined whether it is possible to analytically construct the probability distribution of remaining useful life prediction in certain simple example problems (consisting of linear models and Gaussian variables) and it is demonstrated that it is impossible to estimate closed-form analytical solutions without rigourous mathematical considerations even for such simple example problems.

2. Uncertainty in Prognostics

Prognostics is the art of predicting future component/system behavior, identifying possible failure models, and thereby computing the remaining useful life of the component/system. There are several sources of uncertainty which affect the prediction of future behavior, and in turn, the remaining useful life prediction. As a result of these sources of uncertainty, it is practically impossible to precisely estimate the remaining useful life prediction. In order to make meaningful prognostics-based decision-making, it is important to analyze how the various sources of uncertainty affect the remaining useful life prediction and compute the overall uncertainty in the remaining useful life prediction.

2.1. Activities Related to Uncertainty in PHM

In the context of prognostics and health management, uncertainties have been discussed from representation, quantification, and management points of view (deNeufville, R., 2004; Hastings, D. and McManus, H., 2004; Ng & Abramson, 1990; Orchard, Kacprzynski, Goebel, Saha, & Vachtsevanos, 2008; Tang, Kacprzynski, Goebel, & Vachtsevanos, 2009). While these three are different processes, they are often confused with each other and interchangeably used. In this paper, the various tasks related to uncertainty quantification and management are classified into four, as explained below. These four tasks need to performed in order to accurately estimate the uncertainty in the RUL prediction and inform the decision-maker regarding such uncertainty.

1. **Uncertainty Representation and Interpretation:** The first step is uncertainty representation and interpretation, which in many practical applications, is guided by the choice of modeling and simulation frameworks. There are several methods for uncertainty representation that vary in the level of granularity and detail. Some common theories include classical set theory, probability theory, fuzzy set theory, fuzzy measure (plausibility and belief) theory, rough set (upper and lower approximations) theory, etc. Amongst these theories, probability theory has been widely used in the PHM domain (Celaya, Saxena, & Goebel, 2012); even within the context of probabilistic methods, uncertainty can be interpreted and perceived in two different ways: frequentist (classical) versus subjectivist (Bayesian). Sections 3 and 4 outline the differences between these two schools of thought and argues that the Bayesian approach provides a more suitable interpretation for uncertainty in PHM.
2. **Uncertainty Quantification:** The second step is uncertainty quantification, that deals with identifying and characterizing the various sources of uncertainty that may affect prognostics and RUL estimation. It is important that these sources of uncertainty are incorporated into models and simulations as accurately as possible. The common sources of uncertainty in a typical PHM application include modeling errors, model parameters, sensor noise and measurement errors, state estimates (at the time at which prediction needs to be performed), future loading,

operating and environmental conditions, etc. The goal in this step is to address each of these uncertainties separately and quantify them using probabilistic/statistical methods. The Kalman filter is essentially a Bayesian tool for uncertainty quantification, where the uncertainty in the states is estimated continuously as a function of time, based on data which is also typically available continuously as a function of time.

3. **Uncertainty Propagation:** The third step is uncertainty propagation and is most relevant to prognostics, since it accounts for all the previously quantified uncertainties and uses this information to predict (1) future states and the associated uncertainty; and (2) remaining useful life and the associated uncertainty. The former is computed by propagating the various sources of uncertainty through the prediction model. The latter is computed using the estimated uncertainty in the future states along with a Boolean threshold function which is used to indicate end-of-life. In this step, it is important to understand that the future states and remaining useful life predictions are simply dependent upon the various uncertainties characterized in the previous step, and therefore, the distribution type and distribution parameters of future states and remaining useful life should not be arbitrarily chosen. Sometimes, a normal (Gaussian) distribution has been assigned to the remaining useful life prediction; such an assignment is erroneous and the true probability distribution of RUL needs to be estimated though rigorous uncertainty propagation of the various sources of uncertainty through the state space model and the EOL threshold function, both of which may be non-linear in practice.

4. **Uncertainty Management:** The fourth and final step is uncertainty management, and it is unfortunate that, in several articles, the term "Uncertainty Management" has been used instead of uncertainty quantification and/or propagation. Uncertainty management is a general term used to refer to different activities which aid in managing uncertainty in condition-based maintenance during real-time operation. There are several aspects of uncertainty management. One aspect of uncertainty management attempts to answer the question: "Is it possible to improve the uncertainty estimates?" The answer to this question lies in identifying which sources of uncertainty are significant contributors to the uncertainty in the RUL prediction. For example, if the quality of the sensors can be improved, then it may be possible to obtain a better state estimate (with lesser uncertainty) during Kalman filtering, which may in turn lead to a less uncertain RUL prediction. Another aspect of uncertainty management deals with how uncertainty-related information can be used in the decision-making process.

Most of the research in the PHM community pertains to the topics of uncertainty quantification and propagation; few articles have directly addressed the topic of uncertainty management. Even within the realm of uncertainty quantification and propagation, the estimates of uncertainty have sometimes been misinterpreted. For example, when statistical principles are used to estimate a parameter, there is an emphasis on calculating the estimate with the minimum variance. When this principle is applied to RUL estimation, it is important not to arbitrarily reduce the variance of RUL itself. Celaya et al. (Celaya, Saxena, & Goebel, 2012) explored this idea and explained that the variance of RUL needs to be carefully calculated by accounting for the different sources of uncertainty.

2.2. Sources of Uncertainty

In many practical applications, it may even be challenging to identify and quantify the different sources of uncertainty that affect prognostics. Some researchers have classified the different sources of uncertainty into different categories in order facilitate uncertainty quantification and management. While it has been customary to classify the different sources of uncertainty into aleatory (physical variability) and epistemic (lack of knowledge), such a classification may not be suitable for condition-based monitoring purposes; this point will be explained in detail in the next section. A completely different approach for classification, particularly applicable to condition-based monitoring, is outlined below:

1. **Present uncertainty**: Prior to prognosis, it is important to be able to precisely estimate the condition/state of the component/system at the time at which RUL needs to be computed. This is related to state estimation commonly achieved using filtering. Output data (usually collected through sensors) is used to estimate the state and many filtering approaches are able to provide an estimate of the uncertainty in the state. Practically, it is possible to improve the estimate of the states and thereby reduce the uncertainty, by using better sensors and improved filtering approaches.
2. **Future uncertainty**: The most important source of uncertainty in the context of prognostics is due to the fact that the future is unknown, i.e. both the loading and operating conditions are not known precisely, and it is important to assess the uncertainty in loading and environmental conditions before performing prognostics. If these quantities were known precisely (without any uncertainty), then there would be no uncertainty regarding the *true* remaining useful life of the component/system. However, this true RUL needs to be estimated using a model; the usage of a model imparts additional uncertainty as explained below.
3. **Modeling uncertainty**: It is necessary to use a functional model in order to predict future state behavior.

Further, as mentioned before, the end-of-life is also defined using a Boolean threshold function which indicates end-of-life by checking whether failure has occurred or not. These two models are combinedly used to predict the RUL, and may either be physics-based or data-driven. It may be practically impossible develop models which accurately predict reality. Modeling uncertainty represents the difference between the predicted response and the true response (which can neither be known nor measured accurately), and comprises of several parts: model parameters, model form, and process noise. While it may be possible to quantify these terms until the time of prediction, it is practically challenging to know their values at future time instants.

4. **Prediction method uncertainty**: Even if all the above sources of uncertainty can be quantified accurately, it is necessary to quantify their combined effect on the RUL prediction, and thereby, quantify the overall uncertainty in the RUL prediction. It may not be possible to do this accurately in practice and leads to additional uncertainty. This topic will be revisited again, later in Section 5.

While the different sources of uncertainty and the various uncertainty-related activities have been explained in detail, it is important to understand how to interpret this uncertainty. This topic is detailed in the next section.

3. Interpreting Uncertainty

A probabilistic approach to uncertainty representation and quantification has been most commonly used in the prognostics and health management domain. Though probabilistic methods, mathematical axioms and theorems of probability have been well-established in the literature, there is considerable disagreement among researchers on the interpretation of probability. There are two major interpretations based on physical and subjective probabilities, respectively. It is essential to understand the difference between these two interpretations before attempting to interpret the uncertainty in RUL prediction.

3.1. Physical Probability

Physical probabilities (Szabó, 2007), also referred to objective or frequentist probabilities, are related to random physical experiments such as rolling dice, tossing coins, roulette wheels, etc. Each trial of the experiment leads to an event (which is a subset of the sample space), and in the long run of repeated trials, each event tends to occur at a persistent rate, and this rate is referred to as the relative frequency. These relative frequencies are expressed and explained in terms of physical probabilities. Thus, physical probabilities are defined only in the context of random experiments. The theory of classical statistics is based on physical probabilities. Within the realm of physical probabilities, there are two types of interpretations: von Mises' frequentist (Von Mises, 1981) and Popper's propensity (Popper, 1959); the former is more easily understood and widely used.

In the context of physical probabilities, randomness arises only due to the presence of physical probabilities. If the true value of any particular quantity is deterministic, then it is not possible to associate physical probabilities to that quantity. In other words, when a quantity is not random but unknown, then tools of probability cannot be used to represent this type of uncertainty. For example, the mean of a random variable, sometimes referred to as the population mean, is deterministic. It is meaningless to talk about its probability distribution. In fact, for any type of parameter estimation, the underlying parameter is assumed to be deterministic and only an estimate of this parameter is obtained. The uncertainty in the parameter estimate is addressed through confidence intervals. The interpretation of confidence intervals, as explained in the forthcoming subsection, is sometimes confusing and misleading. Further, the uncertainty in the parameter estimate cannot be used for further uncertainty quantification. For example, if the model parameters of a battery model are estimated under a particular loading condition, then this uncertainty cannot be used for quantifying the battery-response for a similar loading condition. This is a serious limitation, since it is not possible to propagate uncertainty after parameter estimation, which is often necessary in system-level uncertainty quantification (Sankararaman, 2012).

Clearly, there are two limitations of the frequentist interpretation of probability. First, a truly deterministic but unknown quantity cannot be assigned a probability distribution. Second, uncertainty represented using confidence intervals cannot be used for further uncertainty propagation. The second interpretation of probability, i.e. the subjective interpretation, overcomes these limitations.

3.2. Subjective Probability

Subjective probabilities (de Finetti, 1977) can be assigned to any "statement". It is not necessary that the concerned statement is in regard to an event which is a possible outcome of a random experiment. In fact, subjective probabilities can be assigned even in the absence of random experiments. The Bayesian methodology is based on subjective probabilities, which are simply considered to be degrees of belief and quantify the extent to which the "statement" is supported by existing knowledge and available evidence. In recent times, the terms "subjectivist" and "Bayesian" have become synonymous with one another. Calvetti and Somersalo (Calvetti & Somersalo, 2007) explain that "randomness" in the context of physical probabilities is equivalent to "lack of information" in the context of subjective probabilities. In this approach, even deterministic quantities can be represented using probability distributions which reflect the subjective degree of the analysts belief regarding such quantities. As a result, probability distributions can be assigned to parameters that need to be estimated, and therefore, this interpretation facilitates uncertainty propagation after parameter estimation. Interestingly, subjective probabilities can also be applied in situations where physical probabilities are involved (Sankararaman, 2012).

The concept of likelihood and its use in Bayes' theorem are key to the theory of subjectvive probability. The numerical implementation of Bayes' theorem may be complicated in some practical cases, and several sampling techniques have been developed by researchers to address this issue. Today, Bayesian methods are used to solve a variety of problems in engineering. Filtering techinques such as particle filtering, Kalman filtering, etc. are also primarily based on the use of Bayes theorem, and sequential sampling.

3.3. Summary

Both the frequentist and subjectivist approaches have been well-established in the literature, in order to aid uncertainty quantification. In fact, both the approaches may yield similar results (but different interpretations) for a few standard problems involving Gaussian variables. Sometimes, both approaches may be suitable for a given problem at hand; for example, Kalman filtering has a purely frequentist interpretation based on least squares minimization as well as a purely Bayesian interpretation which relies on continuously updating the uncertainty in the state estimates using Bayes theorem. It is acceptable to interpret uncertainty using the frequentist approach or the Bayesian approach, provided the interpretation is suitable for the problem at hand. The following section further explores this idea in the context of PHM and RUL estimation.

4. Understanding Uncertainty in RUL

Consider the problem of estimating the remaining useful life prediction, in the context of prognostics and health management. Researchers have pursued two different classes of methods for this purpose; while the first method is based on reliability-testing, the second method is based on condition-monitoring and future behavior prediction. There is a significant difference in the interpretation of uncertainty, when RUL is estimated using these two different approaches. Understanding this difference is important for prognostics and decision-making, and this is focus of the present section.

4.1. Testing-Based Prognostics

Consider a simple numerical example where the remaining useful life needs to be calculated at a given time instant. Assume that a set of run to failure experiments have been performed with high level of control, ensuring same usage and operating conditions. The time to failure for all the n samples ($r_i; i = 1$ to n) are measured. It is important to understand that *different* RUL values are obtained due to inherent variability across the n different specimens, thereby confirming the presence of physical probabilities. Assume that these random samples belong to an underlying probability density function (PDF) $f_R(r)$, with expected value $E(R) = \mu$ and variance $Var(R) = \sigma^2$. The goal of uncertainty quantification is to characterize this probability density function based on the available n data. Theoretically, an infinite amount of data is necessary to accurately estimate this PDF; however, due to the presence limited data, the estimated PDF is not accurate. As a result, both frequentists and subjectivists express uncertainty regarding the estimate itself. However, frequentists and subjectivists quantify and express this uncertainty in completely different ways.

For the sake of illustration, assume that the entire PDF can be equivalently represented using its mean and variance; in other words, assume that the random variable R follows a two-parameter distribution. Therefore, estimating the parameters μ and σ is equivalent to estimating the PDF. In the context of physical probabilities (frequentist approach), the "true" underlying parameters μ and σ are referred to as "population mean" and "population standard deviation" respectively. Let θ and s denote the mean and the standard deviation of the available n data. As stated earlier, due to the presence of limited data, the sample parameters (θ and s) will not be equal to the corresponding population parameters (μ and σ). The fundamental assumption in this approach is that, since there are true but unknown population parameters, it is meaningless to talk about the probability distribution of any population parameter. Instead, the sample parameters are treated as random variables, i.e., if another set of n data were available, then another realization of θ and s would have been obtained. Using the sample parameters (μ and σ) and the number of data available (n), frequentists construct confidence intervals on the population parameters.

Confidence intervals can be constructed for both μ and σ (Haldar & Mahadevan, 2000). It is important that these intervals be interpreted correctly. As stated earlier, the interpretation of confidence intervals may be confusing and misleading. A 95% confidence interval on μ *does not* imply that "the probability that μ lies in the interval is equal to 95%"; such a statement is wrong because μ is purely deterministic and physical probabilities cannot be associated with it. The random variable here is in fact θ, and the interval calculated using θ. Therefore, the correct implication is that "the probability that the estimated confidence interval contains the true population mean is equal to 95%".

Alternatively, it is also possible to address the problem of computing $f_R(r)$ purely from a subjective (Bayesian) point of view. One important difference now is that the Bayesian approach does not clearly differentiate between "sample parameters" and "population parameters". The probability distribution of μ is directly computed using the available data (recall that this was impossible in the frequentist approach since μ is the true parameter and precise but unknown), and this uncertainty is referred to as the analyst's degree of belief for the underlying true parameter μ. Similarly, the probability distribution of σ can also be computed. Recall that one realization of the parameters (μ and σ) uniquely define the PDF $f_R(r)$. However, since the parameters are themselves uncertain, R is now represented by a family of distributions (Sankararaman & Mahadevan, 2011), reflective of the fact that there is limited data. This family of distributions will shrink to the true underlying PDF as the number of available data increases.

4.2. Condition-Based Prognostics

Most of the discussion pertaining to testing-based prognostics is not applicable to condition-based monitoring and prognostics. The distinctive feature of condition-based monitoring is that each component/subsystem/system is considered by itself, and therefore, "variability across specimens" is non-existent. Any such "variability" is spurious and must not be considered. At any generic time instant t_P at which prognostics needs to be performed, the component/subsystem/system is at a specific state. The actual state of the system is purely deterministic, i.e., the true value is completely precise, however unknown. Therefore, if a probability distribution is assigned for this state, then this distribution is simply reflective of the analyst's knowledge regarding this state and cannot be interpreted from a frequentist point of view. Thus, by virtue of definition of condition-based monitoring, physical probabilities are not present here, and a subjective (Bayesian) approach is only suitable for uncertainty quantification.

The goal in condition-based prognostics is, at any generic time instant t_P, to predict the remaining useful life of the component/subsystem/system as condition-based estimate of the usage time left until failure. First, measurements until time t_P are used to estimate the state at time t_P. Then, using a forecasting method (which may be model-based or data-driven), future state values (corresponding to time instants greater than t_P) are computed. In order to forecast future state values, it is also necessary to assume future loading conditions (and operating conditions) which is a major challenge in condition-based prognostics. Typically, the analyst subjectively assumes statistics for future loading conditions based on past experience and existing knowledge; thus, the subjective interpretation of uncertainty is clearly consistent across the entire condition-based monitoring procedure, and therefore, inferences made out of condition-based monitoring also need to be interpreted subjectively. This forecasting is stopped when failure is reached, as indicated by the aforementioned boolean threshold function. This indicates the end-of-life (EOL) and the EOL can be directly used to compute the remaining useful life (RUL) prediction. Note that it is important to interpret the uncertainty in EOL and RUL subjectively.

4.3. Why is the RUL Prediction Uncertain?

In light of the above discussion, it is necessary to revisit the question "Why is the RUL uncertain?" from a new perspective. While Section 2 explained that RUL is uncertain because there are several sources of uncertainty which influence RUL estimation, now it is clear that the uncertainty in RUL could arise due to variability across multiple specimens (testing-based prognostics scenario) or simply due to subjective uncertainty regarding a single specimen (condition-based prognostics). The following section discusses the computation of RUL in detail by presenting a detailed framework for uncertainty quantification in prognostics, and explains how to calculate the uncertainty in remaining useful life prediction.

5. Uncertainty Quantification in RUL

First, a general computational framework for uncertainty quantification in prognostics and remaining useful life prediction in presented. Second, it is illustrated as to how the problem of computing uncertainty in the remaining useful life prediction can be viewed as an uncertainty propagation problem. Third, the need of rigorous mathematical algorithms for uncertainty quantification in RUL is demonstrated using certain numerical examples. Finally, the challenges involved in computing RUL uncertainty are discussed in detail.

5.1. Computational Framework for Prognostics

Suppose that it is desired to perform prognostics and predict the RUL at a generic time-instant t_P. Daigle and Goebel (Daigle & Goebel, 2011) explain that it is important to develop an architecture for model-based prognostics for practical engineering purposes. This paper considers the architecture in Fig. 1, where the whole problem of prognostics can be considered to consist of the following three sub-problems:

1. Present state estimation
2. Future state prediction
3. RUL computation

5.1.1. State Estimation

The first step of estimating the state at t_P serves as the precursor to prognosis and RUL computation. Consider the state space model which is used to continuously predict the state of the system, as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{v}(t)) \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\theta}(t) \in \mathbb{R}^{n_\theta}$ is the parameter vector, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(t) \in \mathbb{R}^{n_v}$ is the process noise vector, and $\mathbf{f}$ is the state equation.

The state vector at time t_P, i.e., $\mathbf{x}(t)$ (and the parameters $\boldsymbol{\theta}(t)$,

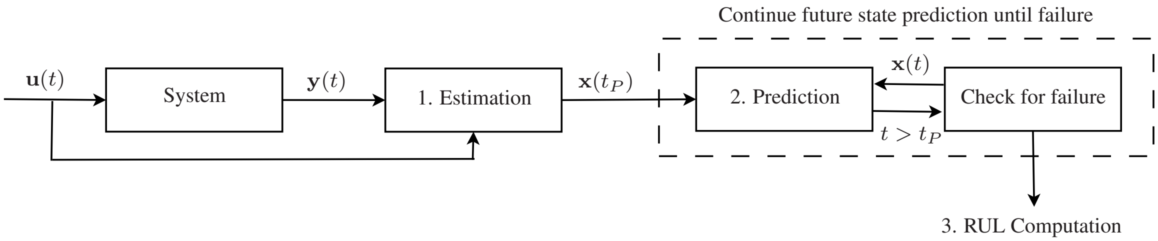


Figure 1. Model-Based Prognostics Architecture

if they are unknown) is (are) estimated using output data collected until t_P. Let $\mathbf{y}(t) \in \mathbb{R}^{n_y}$, $\mathbf{n}(t) \in \mathbb{R}^{n_n}$, and $\mathbf{h}$ denote the output vector, measurement noise vector, and output equation respectively. Then,

$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{n}(t)) \tag{2}$$

Typically, filtering approaches such as Kalman filtering, particle filtering, etc. may be used for such state estimation. It must be recalled that these filtering methods are collectively known as Bayesian tracking methods, not only because they use Bayes theorem for state estimation but also are based on the subjective interpretation of uncertainty. In other words, any time instant, there is nothing uncertain regarding the true states. However, the true states are not known precisely, and therefore, the probability distributions of these state variables are estimated through filtering. The estimated probability distributions are simply reflective of the subjective knowledge regarding those state variables.

5.1.2. State Prediction

Having estimated the state at time t_P, Eq. 1 is used to predict the future states of the component/system. This differential equation can be discretized and used to predict $\mathbf{x}(t+1)$ as a function of $\mathbf{x}(t)$. Therefore, using this recursive relation, the state at any future time instant $t > t_P$ can be calculated.

5.1.3. RUL Computation

RUL computation is concerned with the performance of the component that lies outside a given region of acceptable behavior. The desired performance is expressed through a set of n_c constraints, $C_{EOL} = \{c_i\}_{i=1}^{n_c}$, where $c_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_u} \to \mathbb{B}$ maps a given point in the joint state-parameter space given the current inputs, $(\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t))$, to the Boolean domain $\mathbb{B} \triangleq [0, 1]$, where $c_i(\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t)) = 1$ if the constraint is satisfied, and 0 otherwise (Daigle & Goebel, 2013).

These individual constraints may be combined into a single *threshold function* $T_{EOL} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_u} \to \mathbb{B}$, defined as:

$$T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t)) = \begin{cases} 1, & 0 \in \{c_i(\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t))\}_{i=1}^{n_c} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

T_{EOL} is equal to 1 when any of the constraints are violated. Then, the End of Life (EOL, denoted by E) at any time instant t_P is then defined as the earliest time point at which the value of T_{EOL} becomes equal to one. Therefore,

$$E(t_P) \triangleq \inf\{t \in \mathbb{R} : t \geq t_P \wedge T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t)) = 1\}. \tag{4}$$

The Remaining Useful Life (RUL, denoted by R) at time instant t_P is expressed as:

$$R(t_P) \triangleq E(t_P) - t_P. \tag{5}$$

Note that the output equation (Eq. 2) or output data ($\mathbf{y}(t)$) is not used in the prediction stage, and EOL and RUL are dependent only on the state estimates at time t_P; though these state estimates are obtained using the output data, the output data is not used for EOL/RUL calculation after state estimation.

For the purpose of implentation, $\mathbf{f}$ in Eq. 1 is transformed into the corresponding discrete-time version.

5.2. RUL Estimation through Uncertainty Propagation

Thus, it is clear that RUL predicted at time t_P, i.e., $R(t_P)$ depends on

1. Present state estimate ($\mathbf{x}(t_P)$); using the present state estimate and the state space equations in Eq. 1, the future states ($\mathbf{x}(t_P)$, $\mathbf{x}(t_P+1)$, $\mathbf{x}(t_P+2)$, ..., $\mathbf{x}(t_P+R(t_P))$) can be calculated.
2. Future loading ($\mathbf{u}(t_P)$, $\mathbf{u}(t_P+1)$, $\mathbf{u}(t_P+2)$, ..., $\mathbf{u}(t_P+R(t_P))$); these values are needed to calculate the future state values using the state space equations.
3. Parameter values from time t_P until time $t_P + R(t_P)$ (denoted by $\boldsymbol{\theta}(t_P)$, $\boldsymbol{\theta}(t_P+1)$, ..., $\boldsymbol{\theta}(t_P+R(t_P))$).
4. Process noise ($\mathbf{v}(t_P)$, $\mathbf{v}(t_P+1)$, $\mathbf{v}(t_P+2)$, ..., $\mathbf{v}(t_P+R(t_P))$).

For the purpose of RUL prediction, all of the above quantities are independent quantities and hence, RUL becomes a dependent quantity. Let $\boldsymbol{X} = \{X_1, X_2, ...X_i, ...X_n\}$ denote the vector of all the above dependent quantities, where n is the length of the vector $\boldsymbol{X}$, and therefore the number of uncertain quantities that influence the RUL prediction. Then the calculation of RUL (denoted by R) can be expressed in terms of a function, as:

$$R = G(\boldsymbol{X}) \tag{6}$$

The above functional relation in Eq. 6 can be graphically explained, as shown in Fig. 2.

For example, consider the case where the component/system is subjected to uniform loading (characterized by one variable, the amplitude which remains constant with time), modeled using one parameter (which is time-invariant), and characterized using two states (the state estimates at time t_P and Eq. 1 can be used to predict the state values at any future time instant). Then, excluding the effect of process noise, there are $n = 4$ quantities that affect the RUL prediction. Note that there are $R(t_P) + 1$ process noise terms for each state; therefore, the inclusion of process noise increases the value of n, and therefore the dimensionality of the problem. This raises a practical concern and has been addressed in an earlier publication by replacing the time-variant process noise using an equivalent time-invariant process noise (Sankararaman & Goebel, 2013). In the rest of the paper, a generalized framework is presented without using this equivalent time-invariant process noise concept.

Knowing the values of $\boldsymbol{X}$, it is possible to compute the value of R, using Fig. 2 that is equivalently represented by Eq. 6. The quantities contained in $\boldsymbol{X}$ are uncertain, and the focus in prognostics to compute their combined effect on the RUL prediction, and thereby compute the probability distribution of R. The problem of estimating the uncertainty in R is equivalent to propagating the uncertainty in $\boldsymbol{X}$ through G, and it is necessary to use computational methods for this purpose.

5.3. Need for Computational Approaches

The problem of estimating the uncertainty in R using uncertainty propagation techniques is a non-trivial problem, and needs rigorous computational approaches. This involves estimating the probability density function of R (PDF, denoted by $f_R(r)$) or equivalently the cumulative distribution function of R (CDF, denoted by $F_R(r)$). In some rare cases, it is possible to analytically obtain the distribution of R. Some of such special cases are listed below:

1. Each and every quantity contained in $\boldsymbol{X}$ follows a normal (Gaussian) distribution, and the function G can be expressed as a weighted linear combination of the quantities in $\boldsymbol{X}$. In this case, R also follows a normal distribution, and its statistics can be calculated analytically.

2. Each and every quantity contained in $\boldsymbol{X}$ follows a lognormal distribution, and if the *logarithm* of the function G can be expressed as a weighted combination of the quantities in $\boldsymbol{X}$, then $log(R)$ follows a normal distribution whose statistics can be estimated analytically. In other words, R also follows a lognormal distribution.

While Gaussian distributions and linear state space models (linear $\mathbf{f}$ in Eq. 1) may be commonly used in the prognostics and health management domain, it is important to understand that using linear state space models is not equivalent to G being linear. In other words, the use of the threshold function along with the linear state models automatically renders G non-linear.

In order to illustrate this important point, and to emphasize the importance of using rigorous computational methods, consider a simple example where the state state equation is given by:

$$x(t+1) = a.x(t) + b. \tag{7}$$

Assume that a suitable time-discretization has been chosen for the purpose of implementation. It is desired to predict future behavior and compute RUL at $t_P = 0$, and state value at this time is denoted by $x(0)$ which is a Gaussian random variable. Further a and b are constants (i.e., not random) which are used to predict future states. It can be easily demonstrated that the state value at any future time instant can be expressed as a function of $x(0)$.

$$x(n) = a^n.x(0) + \sum_{j=0}^{j=n-1} a^j b \tag{8}$$

It is clear from Eq. 8 that the state value at any future time instant is a linear function of $x(0)$, and therefore is also Gaussian. In order to compute the remaining useful life, it is necessary to chose a threshold function. Depending on the choices of a and b, $x(n)$ may either be an increasing function or a decreasing function. If $x(n)$ is a decreasing function, then the threshold function will indicate that failure occurs when the state value x becomes smaller than a critical lower bound (l). Alternatively, if $x(n)$ is an increasing function, then the threshold function will indicate that failure occurs when the state value x becomes greater than a critical upper bound u. Without loss of generality, any of the two cases may be chosen for illustrative purposes. For example, consider that $x(n)$ is decreasing and failure happens when $x < l$. Therefore, the remaining useful life (r, an instance of the random variable R) is equal to the smallest n such that $x(n) < l$. Therefore RUL can be calculated as

$$r = \inf\{n : a^n.x(0) + \sum_{j=0}^{j=n-1} a^j b < l\}, \tag{9}$$

Assuming that the chosen time-discretization level is infinitesimally small, it is possible to directly estimate the RUL by solving the equation:

$$a^r.x(0) + \sum_{j=0}^{j=r-1} a^j.b = l. \tag{10}$$

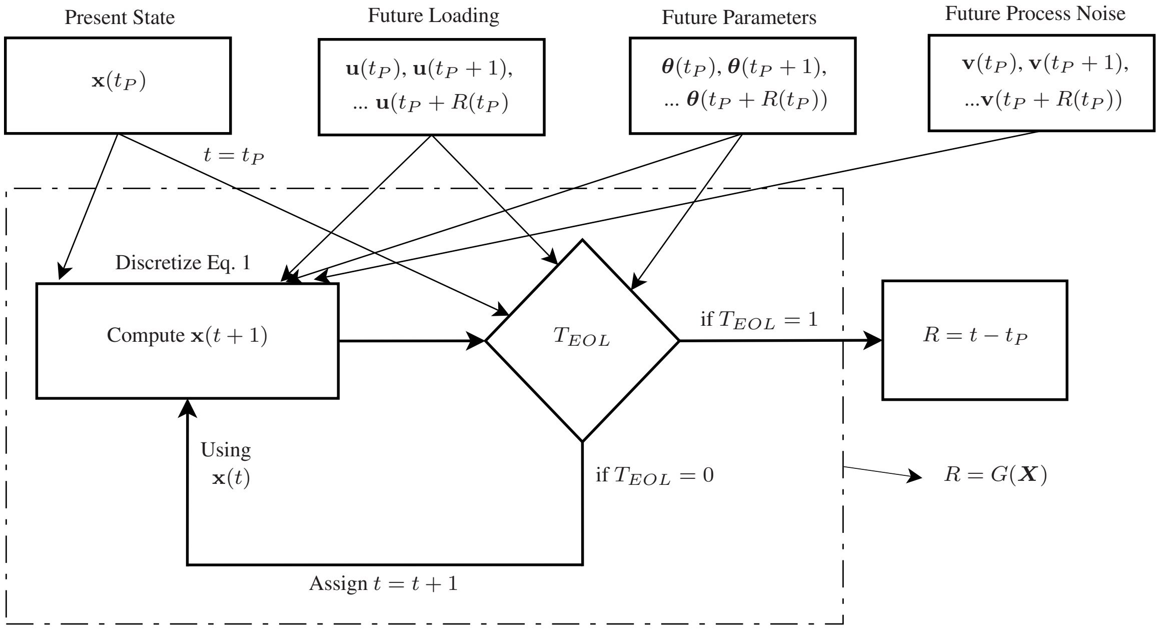

Present State
$\mathbf{x}(t_P)$
Future Loading
$\mathbf{u}(t_P), \mathbf{u}(t_P+1)$, ... $\mathbf{u}(t_P+R(t_P)$
Future Parameters
$\boldsymbol{\theta}(t_P), \boldsymbol{\theta}(t_P+1)$, ... $\boldsymbol{\theta}(t_P+R(t_P))$
Future Process Noise
$\mathbf{v}(t_P), \mathbf{v}(t_P+1)$, ...$\mathbf{v}(t_P+R(t_P))$
$t=t_P$
Discretize Eq. 1
Compute $\mathbf{x}(t+1)$
T_{EOL}
if $T_{EOL}=1$
$R=t-t_P$
Using $\mathbf{x}(t)$
if $T_{EOL}=0$
$R=G(\boldsymbol{X})$
Assign $t=t+1$

Figure 2. Definition of G

The above equation calculates the RUL (r) as a function of the initial state ($x(0)$). Hence, the above equation is similar to G defined earlier in Fig. 2. The difference now is that the only considered source of uncertainty is the state estimate $x(0)$; model uncertainty, future loading uncertainty, etc. are not considered here. The RUL R follows a Gaussian distribution if and only if it is linearly dependent on $x(0)$. In other words, R follows a Gaussian distribution if and only if Eq. 10 can be rewritten as:

$$\alpha.r+\beta.x(0)+\gamma=0 \tag{11}$$

for some arbitrary values of α, β, and γ. If it were possible to estimate such values for α, β, and γ, the distribution of RUL can be obtained analytically.

In order to examine if this is possible, rewrite Eq. 10 as:

$$x(0)=\frac{1}{a^r}(l-\sum_{j=0}^{j=r-1} a^j.b) \tag{12}$$

While $x(0)$ is completely on the left hand side of this equation, r appears not only as an exponent in the denominator but is also indicative of the number of terms in the summation on the right hand side of the above equation. Therefore, it is clear that the relationship between r and $x(0)$ is not linear. Therefore, even if the initial state ($x(0)$, a realization of $X(0)$) follows a Gaussian distribution, the RUL (r, a realization of R) does not follow a Gaussian distribution. Thus, it is clear that even for a simple problem consisting of linear state models, an extremely simple threshold function, and only one uncertain variable that is Gaussian, the calculation of the probability distribution of R is neither trivial nor straightforward.

Practical problems in the prognostics and health management domain may consist of:

1. Several non-Gaussian random variables which affect the RUL prediction,
2. A non-linear multi-dimensional state space model,
3. Uncertain future loading conditions,
4. A complicated threshold function which may be defined in multi-dimensional space.

The fact that the distribution of RUL simply depends on the quantities indicated in Fig. 2 implies that it is technically inaccurate to artifically assign the probability distribution type (or any statistic such as the mean or variance) to RUL.

5.4. Illustrations

Sometimes, the probability distribution of RUL may be extremely skewed; for example, the RUL of a lithium-ion battery used to power an unmanned aerial vehicle was predicted by Sankararaman et al. (Sankararaman et al., 2013) and it was observed that the probability distribution was particularly skewed near failure. The PDFs of the End-of-Discharge (EOD) prediction at various time instants ($T=0$ seconds through $T=4000$) are shown in Fig. 3 and then the PDF of End-of-Discharge predicted at $T=5000$ seconds (which corresponds to near-failure) is indicated in Fig. 4. The RUL can be calculated by simply subtracting the prediction time-instant from the EOD prediction. Note the significant change in the shape of the PDF near failure. It is extremely important to be able to accurately predict the RUL particularly as failure is approaching, and it is clear from Fig. 4 that assuming a normal distribution or an arbitrary standard deviation would not be able to achieve this goal; only a theoretically accurate uncertainty quantification method can reproduce this probability distribution, whose mode almost coincides with its lower bound (left-hand-side tail).

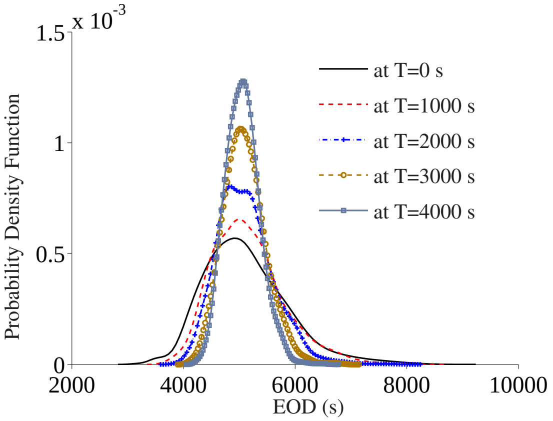


Figure 3. EOD Prediction at Multiple Time Instants

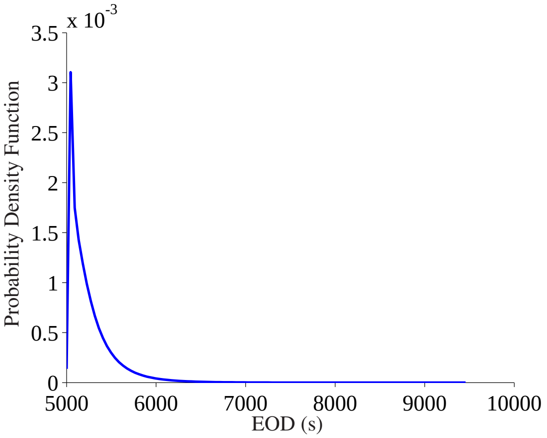


Figure 4. EOD Prediction at $T = 5000$ seconds (near failure)

Sometimes, depending on the chosen statistics of future loading conditions, the distribution of EOD may even be multimodal. For example, Saha et al. (Saha & Goebel, 2008) calculated future loading statistics that lead to a multi-modal PDF for the EOD, as shown in Fig. 5.

It is important to capture such characteristics of the RUL (which is equivalent to the end-of-discharge in Fig 3-5) probability distribution, and this can be accomplished only by using accurate uncertainty quantification methodologies without making critical assumptions regarding the shape of the PDF of the RUL, its mean, median, mode, standard deviation, etc. Therefore, the goal must be to accurately calcuate the probability distribution of R by propagating the different sources of uncertainty through G as indicated in Fig. 2.

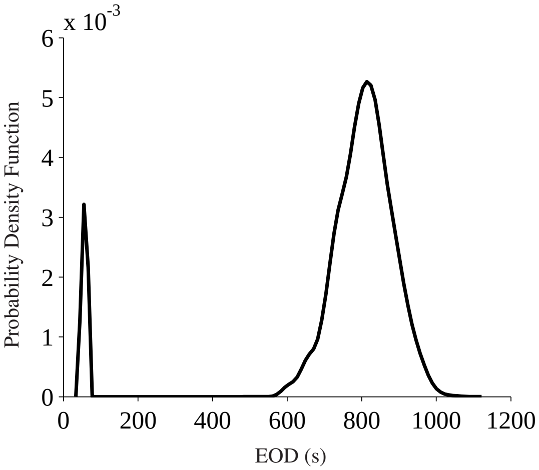


Figure 5. A Multi-Modal PDF

5.5. Uncertainty Propagation Methods

In order to answer the obvious question: “How to calculate the uncertainty in R and estimate the PDF of R?”, it is necessary to resort to rigorous computational methodologies which have been developed by statisticians and researchers in the field of uncertainty quantification in order to solve a typical uncertainty propagation problem. There are different types of sampling methods such as Monte Carlo sampling (Caflisch, 1998), Latin hypercube sampling (Loh, 1996), adaptive sampling (Bucher, 1988), importance sampling (Glynn & Iglehart, 1989), unscented transform sampling (Van Zandt, 2001), etc. Alternatively, there are analytical methods such as the first-order second moment method (Dolinski, 1983), first-order reliability method (Hohenbichler & Rackwitz, 1983), second-order reliability method (Der Kiureghian, Lin, & Hwang, 1987), etc. In addition, there are also methods such as the efficient global reliability analysis (Bichon, Eldred, Swiler, Mahadevan, & McFarland, 2008) method which involve both sampling and the use of analytical techniques. All of these methods empirically calculate the probability distribution of RUL; while some of these methods calculate the PDF ($f_R(r)$) of RUL, some other methods calculate the CDF ($F_R(r)$), and some other methods directly generate samples from the target probability density function ($f_R(r)$). Due to some limitations of each of these methods, it may not be possible to accurately calculate the actual probability distribution of R. Accurate calculation is possible only by using infinite samples for Monte Carlo sampling. Any other method (for example, the use of a limited, finite number of samples) will lead to uncertainty in the estimated probability distribution, and this additional uncertainty is referred to as prediction-method uncertainty. It is possible to decrease (and maybe eventually eliminate) this type of uncertainty either by using advanced probability techniques or powerful computing power.

It is necessary to further investigate the aforementioned uncertainty propagation methods, and identify whether they can be applied to prognostics health monitoring. Some earlier publications have investigated the use of certain methods such as Monte Carlo sampling, unscented transform sampling, first-order reliability methods, etc. in this regard.

5.6. Challenges

There are several challenges in using different uncertainty quantification methods for prognostics, health management and decision-making. It is not only important to understand these challenges but also necessary to understand the requirements of PHM systems in order to integrate efficient uncertainty quantification along with prognostics and aid risk-informed decision-making. Some of the issues involved in such integration are outlined below:

1. An uncertainty quantification methodology for prognostics needs to be computationally feasible for implementation in online health monitoring. This requires quick calculations, while uncertainty quantification methods have been traditionally known to be time-consuming and computationally intensive.
2. Sometimes, the probability distribution of RUL may be multi-modal and the uncertainty quantification methodology needs to be able to accurately capture such distributions.
3. Existing verification, validation, and certification protocols require algorithms to produce deterministic, i.e., repeatable calculations. Several uncertainty quantification methods are non-deterministic, i.e. produce different (albeit, only slightly if implemented well) results on repetition.
4. The uncertainty quantification method needs to be accurate, i.e., the entire probability distribution of $\boldsymbol{X}$ needs to be correctly accounted for, and the functional relationship defined by G in Fig. 2. Some methods use only a few statistics (usually, mean and variance) of $\boldsymbol{X}$ and some methods make approximations (say for example, linear) of G. Finally, it is important to correctly propagate the uncertainty to compute the entire probability distribution of RUL.
5. While it is important to be able to calculate the entire probability distribution of RUL, it is also important to be able to quickly obtain bounds on RUL which can be useful for online decision-making.

Each uncertainty quantification method may address one or more of the above issues, and therefore, it may even be necessary to resort to different methods to achieve different goals. Future research needs to continue this investigation, analyze different types of uncertainty quantification methods and study their applicability to prognostics before these methods can be applied in practice.

6. Conclusion

This paper discussed the significance and interpretation of uncertainty the context of prognostics and health management. The prediction of remaining useful life in engineering systems is affected by several sources of uncertainty, and it is important to correctly interpret this uncertainty in order to facilitate meaningful decision-making. Uncertainty can be interpreted in two ways, either in terms of physical probabilities from a frequentist point of view or in terms of subjective probabilities from a Bayesian point of view. While a frequentist interpretation may be suitable for testing-based prognostics, there are no physical probabilities in the context of condition-based prognostics. Therefore, uncertainty in the context of condition-based monitoring needs to be interpreted subjectively, and hence, a Bayesian approach is more suitable for this purpose. It was also explained that Bayesian tracking methods for state estimation are so-called not only because they use Bayes theorem but are also based on the principle of subjective probability.

This paper also emphasized the importance of accurately computing the uncertainty in the remaining useful life prediction. It was illustrated that it may not be analytically possible to calculate the uncertainty in the remaining useful life prediction even for certain simple problems involving Gaussian random variables and linear state-prediction models. Therefore, it is necessary to resort to computational methodologies for such uncertainty quantification and compute the probability distribution of remaining useful life prediction. In this process, it is important not to make assumptions regarding the shape of the probability distribution of the remaining useful life prediction or any of its statistics such as the mean, median, standard deviation, etc.

Finally, it was explained that the problem of estimating the probability distribution of remaining useful life can be viewed as an uncertainty propagation problem which can be solved using different types of computational approaches. Several sampling-based methods, analytical methods and hybrid methods have been developed by researchers in the field of uncertainty quantification and it is necessary to investigate the applicability of these methods to prognostics and health management. Further, several challenges involved in integrating uncertainty quantification techniques into prognostics and health management were outlined. It is clear that further research is necessary to address these challenges and develop a comprehensive framework for uncertainty quantification in prognostics and health management.

Acknowledgment

The work reported herein was in part funded by the NASA System-wide Safety Assurance Technologies (SSAT) project under the Aviation Safety (AvSafe) Program of the Aeronautics Research Mission Directorate (ARMD), and by the

NASA Automated Cryogenic Loading Operations (ACLO) project under the Office of the Chief Technologist (OCT) of Advanced Exploration Systems (AES). The authors would like to acknowledge valuable discussions with Dr. Abhinav Saxena and Dr. José Celaya, since these discussions raised several important questions regarding uncertainty in prognostics, and ultimately, led to the ideas explored in this paper.

REFERENCES

Bichon, B., Eldred, M., Swiler, L., Mahadevan, S., & McFarland, J. (2008). Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA journal*, *46*(10), 2459–2468.

Bucher, C. G. (1988). Adaptive samplingan iterative fast monte carlo procedure. *Structural Safety*, *5*(2), 119–126.

Caflisch, R. E. (1998). Monte carlo and quasi-monte carlo methods. *Acta numerica*, *1998*, 1–49.

Calvetti, D., & Somersalo, E. (2007). *Introduction to bayesian scientific computing: ten lectures on subjective computing* (Vol. 2). Springer New York.

Celaya, J., Saxena, A., & Goebel, K. (2012). Uncertainty representation and interpretation in model-based prognostics algorithms based on kalman filter estimation. In *Proceedings of the Annual Conference of the PHM Society* (pp. 23–27).

Celaya, J., Saxena, A., Kulkarni, C., Saha, S., & Goebel, K. (2012). Prognostics approach for power MOSFET under thermal-stress aging. In *Reliability and Maintainability Symposium (RAMS), 2012 Proceedings-Annual* (pp. 1–6).

Coppe, A., Haftka, R. T., Kim, N. H., & Yuan, F.-G. (2010). Uncertainty reduction of damage growth properties using structural health monitoring. *Journal of Aircraft*, *47*(6), 2030–2038.

Daigle, M., & Goebel, K. (2010). Model-based prognostics under limited sensing. In *Aerospace Conference, 2010 IEEE* (pp. 1–12).

Daigle, M., & Goebel, K. (2011). A model-based prognostics approach applied to pneumatic valves. *International Journal of Prognostics and Health Management*, *2*(2).

Daigle, M., & Goebel, K. (2013). Model-based prognostics with concurrent damage progression processes. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, *43*(3), 535-546. doi: 10.1109/TSMCA.2012.2207109

Daigle, M., Saxena, A., & Goebel, K. (2012). An efficient deterministic approach to model-based prediction uncertainty estimation. In *Annual conference of the prognostics and health management society* (pp. 326–335).

de Finetti, B. (1977). Theory of probability, Volumes I and II. *Bull. Amer. Math. Soc. 83 (1977), 94-97*, 0002–9904.

deNeufville, R. (2004). Uncertainty management for engineering systems planning and design. In *Engineering systems symposium mit.* Cambridge, MA..

Der Kiureghian, A., Lin, H.-Z., & Hwang, S.-J. (1987). Second-order reliability approximations. *Journal of Engineering Mechanics*, *113*(8), 1208–1225.

Dolinski, K. (1983). First-order second-moment approximation in reliability of structural systems: critical review and alternative approach. *Structural Safety*, *1*(3), 211–231.

Engel, S. J., Gilmartin, B. J., Bongort, K., & Hess, A. (2000). Prognostics, the real issues involved with predicting life remaining. In *Aerospace Conference Proceedings, 2000 IEEE* (Vol. 6, pp. 457–469).

Farrar, C. R., & Lieven, N. A. (2007). Damage prognosis: the future of structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *365*(1851), 623–632.

Glynn, P. W., & Iglehart, D. L. (1989). Importance sampling for stochastic simulations. *Management Science*, *35*(11), 1367–1392.

Gu, J., Barker, D., & Pecht, M. (2007). Uncertainty assessment of prognostics of electronics subject to random vibration. In *AAAI fall symposium on artificial intelligence for prognostics* (pp. 50–57).

Haldar, A., & Mahadevan, S. (2000). *Probability, reliability, and statistical methods in engineering design*. John Wiley & Sons, Incorporated.

Hastings, D. and McManus, H. (2004). A framework for understanding uncertainty and its mitigation and exploitation in complex systems. In *Engineering Systems Symposium MIT* (p. 19). Cambridge MA..

Hohenbichler, M., & Rackwitz, R. (1983). First-order concepts in system reliability. *Structural safety*, *1*(3), 177–188.

Liao, H., Zhao, W., & Guo, H. (2006). Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model. In *Reliability and Maintainability Symposium, 2006. RAMS'06. Annual* (pp. 127–132).

Loh, W.-L. (1996). On latin hypercube sampling. *The annals of statistics*, *24*(5), 2058–2080.

Ng, K.-C., & Abramson, B. (1990). Uncertainty management in expert systems. *IEEE Expert Systems*, 20.

Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008, oct.). Advances in uncertainty representation and management for particle filtering applied to prognostics. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on* (p. 1 -6). doi: 10.1109/PHM.2008.4711433

Popper, K. (1959). The propensity interpretation of probability. *The British journal for the philosophy of science*, *10*(37), 25–42.

Saha, B., & Goebel, K. (2008). Uncertainty management for diagnostics and prognostics of batteries using Bayesian techniques. In *Aerospace Conference, 2008 IEEE* (pp. 1–8).

Sankararaman, S. (2012). *Uncertainty quantification and integration in engineering systems* (Ph.D. Dissertation). Vanderbilt University.

Sankararaman, S., Daigle, M., Saxena, A., & Goebel, K. (2013). Analytical algorithms to quantify the uncertainty in remaining useful life prediction. In *Aerospace Conference, 2013 IEEE* (pp. 1–11).

Sankararaman, S., & Goebel, K. (2013, Apr). Uncertainty quantification in remaining useful Life of aerospace components using state space models and inverse FORM. In *Proceedings of the 15th Non-Deterministic Approaches Conference.*

Sankararaman, S., Ling, Y., & Mahadevan, S. (2011). Uncertainty quantification and model validation of fatigue crack growth prediction. *Engineering Fracture Mechanics*, *78*(7), 1487–1504.

Sankararaman, S., Ling, Y., Shantz, C., & Mahadevan, S. (2011). Uncertainty quantification in fatigue crack growth prognosis. *International Journal of Prognostics and Health Management*, *2*(1).

Sankararaman, S., & Mahadevan, S. (2011). Likelihood-based representation of epistemic uncertainty due to sparse point data and/or interval data. *Reliability Engineering & System Safety*, *96*(7), 814–824.

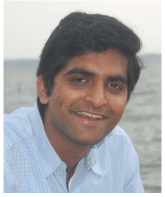
Szabó, L. (2007). Objective probability-like things with and without objective indeterminism. *Studies In History and Philosophy of Science Part B: Studies In History and Philosophy of Modern Physics*, *38*(3), 626–634.

Tang, L., Kacprzynski, G., Goebel, K., & Vachtsevanos, G. (2009, march). Methodologies for uncertainty management in prognostics. In *Aerospace conference, 2009 IEEE* (p. 1 -12). doi: 10.1109/AERO.2009.4839668

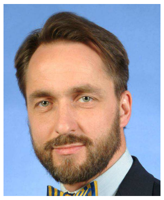
Van Zandt, J. R. (2001). A more robust unscented transform. In *International symposium on optical science and technology* (pp. 371–380).

Von Mises, R. (1981). *Probability, statistics and truth*. Dover Publications.

Biographies



Shankar Sankararaman received his B.S. degree in Civil Engineering from the Indian Institute of Technology, Madras in India in 2007 and later, obtained his Ph.D. in Civil Engineering from Vanderbilt University, Nashville, Tennessee, U.S.A. in 2012. His research focuses on the various aspects of uncertainty quantification, integration, and management in different types of aerospace, mechanical, and civil engineering systems. His research interests include probabilistic methods, risk and reliability analysis, Bayesian networks, system health monitoring, diagnosis and prognosis, decision-making under uncertainty, treatment of epistemic uncertainty, and multidisciplinary analysis. He is a member of the Non-Deterministic Approaches (NDA) technical committee at the American Institute of Aeronautics, the Probabilistic Methods Technical Committee (PMC) at the American Society of Civil Engineers (ASCE), and the Prognostics and Health Management (PHM) Society. Currently, Shankar is a researcher at NASA Ames Research Center, Moffett Field, CA, where he develops algorithms for uncertainty assessment and management in the context of system health monitoring, prognostics, and decision-making.



Kai Goebel is the Deputy Area Lead for Discovery and Systems Health at NASA Ames where he also directs the Prognostics Center of Excellence. After receiving the Ph.D. from the University of California at Berkeley in 1996, Dr. Goebel worked at General Electrics Corporate Research Center in Niskayuna, NY from 1997 to 2006 as a senior research scientist before joining NASA. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion and his interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He holds 17 patents and has published more than 250 papers in the area of systems health management.

A Novel Computational Methodology for Uncertainty Quantification in Prognostics Using The Most Probable Point Concept

Shankar Sankararaman[1] and Kai Goebel[2]

[1] *SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA*
shankar.sankararaman@nasa.gov

[2] *NASA Ames Research Center, Moffett Field, CA 94035, USA*
kai.goebel@nasa.gov

Abstract

This paper develops a novel computational approach to quantify the uncertainty in prognostics in the context of condition-based monitoring. Prognostics consists of two major steps; first, it is necessary to estimate the state of health at any time instant, and then, it is required to predict the remaining useful life of the engineering component/system of interest. While the topic of estimation has been addressed through different types of Bayesian tracking techniques, this paper primarily focuses on the second aspect of future prediction and remaining useful life computation, which is influenced by several sources of uncertainty. Therefore, it is important to identify these sources of uncertainty, and quantify their combined effect on the remaining useful life prediction. The computation of uncertainty in remaining useful life can be treated as an uncertainty propagation problem which can be solved using probabilistic techniques. This paper investigates the use of the Most Probable Point approach (which was originally developed to estimate the failure probability of structural systems) for calculating the probability distribution of the remaining useful life prediction. The proposed methodology is illustrated using a battery which is used to power an unmanned aerial vehicle.

1. Introduction

Research in the past few years has been advocating the use of an onboard health management system in engineering systems used for time-critical, safety-critical, and cost-critical missions. An accurate health management system constantly monitors the performance of the engineering system, performs diagnosis (fault detection, isolation, and estimation), prognosis (predict possible failures in the future and estimate remaining useful life) and aid online decision-making (fault mitigation, fault recovery, mission replanning, etc.). The prediction of remaining useful life is an important aspect of prognostics, and is directly useful in different types of decision-making. This paper focuses on the calculation of remaining useful life in the context of model-based prognostics and condition-based monitoring.

In practical applications, there are several sources of uncertainty which affect the performance of both the engineering system and the health management system. For example, the loading conditions and operating conditions of the engineering system may be random in nature. The sensors, which are part of the health management system, may not be accurate due to measurement errors, and this may prevent accurate estimation of the system state. The system models which are used by the health management system for estimation and prediction may have certain errors. As a result of the presence of such uncertianties, it is important to rigorously account for the sources of uncertainty during diagnosis, prognosis, and decision-making. While the topic of uncertainty quantification in diagnosis has gained attention in literature (Sankararaman & Mahadevan, 2011b, 2013), the importance of uncertainty significantly increases in the context of prognosis, since the focus is on predicting future behavior, which is far more challenging and uncertain than fault diagnosis. The primary objective of this paper is to develop a computational methodology which can quantify the combined effect of the various sources of uncertainty on prognostics, and estimate the overall uncertainty in the remaining useful life (RUL) prediction.

In the past, several researchers have used different types of methods to quantify uncertainty in prognostics. Tang et al. (Tang, Kacprzynski, Goebel, & Vachtsevanos, 2009) discuss the use of Bayesian tracking algorithms for uncertainty quantification and management in prognostics for In-

Shankar Sankararaman et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

tegrated Vehicle Health Management (IVHM) systems. The "Damage Prognosis Project" at Los Alamos National Laboratory (C. R. Farrar, Lieven, & Bement, 2005; Inman, Farrar, Junior, & Junior, 2005) exclusively dealt with prognosis and uncertainty quantification applied to structual composites. Several researchers worked on this project and published articles that deal with model development, verification, validation, prediction, and uncertainty quantification; the conclusions of this project have been documented by Farrar et al (C. Farrar & Lieven, 2007). Sankararaman et al. (Sankararaman, Ling, Shantz, & Mahadevan, 2009, 2011) quantified the uncertainty in fatigue crack growth prognosis in metals, by using finite element models (for structural analysis), crack growth models (to predict future crack growth), and Monte Carlo simulation (for uncertainty quantification). Gu et al. (Gu, Barker, & Pecht, 2007) also used Monte Carlo simulation in order to compute the uncertainty in damage in electronics subjected to random vibration. In some practical applications, Monte Carlo simulation using exhaustive sampling may be computationally expensive, and this challenge has inspired the development of intelligent sampling-based algorithms (DeCastro, 2009; Orchard, Kacprzynski, Goebel, Saha, & Vachtsevanos, 2008; Daigle, Saxena, & Goebel, 2012) and mathematical techniques such as relevance vector machines (Saha & Goebel, 2008) and principle component analysis (Usynin & Hines, 2007), that can reasonably approximate the uncertainty in the prognostic predictions. Further, Bayesian and maximum relative entropy methods (Guan et al., 2011) have also been used for estimating uncertainty in prognostics.

The above described methods for uncertainty quantification are mainly based on sampling techniques, and may require several thousands of samples to accurately calculate the uncertainty in RUL prediction. This may be computationally expensive for online monitoring, and therefore, Sankararaman et al. (Sankararaman, Daigle, Saxena, & Goebel, 2013) discussed analytical approaches for predicting the uncertainty quantification. These analytical methods are based on first-order Taylor's series expansion, and in particular, one method known as the Inverse First-order Reliability Method was implemented for calculating the uncertianty in RUL prediction. This method was originally developed by structural engineers for calculating the failure probability of structural engineering applications. This method is based on the concept of Most Probable Point (MPP) Estimation, and this paper further investigates the applicability of this approach to different types of loading conditions and RUL calculation. Note that the term "reliability method" is avoided in the rest of the present paper in order to avoid confusion with "reliability testing" methods for prognostics, since the proposed methodology is primarily applicable to condition-based online monitoring.

The rest of the paper is organized as follows. Section 2 discusses a framework for uncertainty quantification in prognostics, and explains that the problem of estimating the uncertainty in the RUL prediction can be viewed as an uncertainty propagation problem. Section 3 discusses the importance of future loading conditions, and describes three different types of future loading scenarios for prognostics. Section 4 explains the proposed computational methodology for quantifying prognostics uncertainty and estimating the uncertainty in the remaining useful life prediction; this methodology is illustrated using a numerical example in Section 5. Finally, conclusions are drawn in Section 6.

2. Uncertainty Quantification in Condition-Based Monitoring and Prognostics

This section discusses the need for uncertainty quantification in prognostics and health monitoring. First, a model-based framework for prognostics is presented, and then, the various sources of uncertainty are discussed with reference to this framework. Finally, it is illustrated that uncertainty quantification in prognostics can be viewed as an uncertainty propagation problem.

2.1. Model-based Framework for Prognostics

The goal of prognostics is to predict the future behavior of a component/system at any generic instant of prediction, denoted by t_P. This is accomplished by estimating the states of the systems at all time instants $t > t_P$. The inputs ($\mathbf{u}(t)$) and outputs ($\mathbf{y}(t)$) to the system are known until the prediction time $t = t_P$, and in order to perform prediction, the future inputs i.e., $\mathbf{u}(t)\ \forall\ t > t_P$ also need to be available. A generalized prognostics architecture is showed in Fig. 1.

The first step in prognostics is to estimate the state at time t_P. Using the estimated state, the second step is to predict future states until failure; thereby, the end-of-life (EOL) and the remaining useful life (RUL) can be predicted.

State-space models are used for both estimation and prediction. Consider a generic state space model which is used to continuously predict the state of the system, as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{v}(t)) \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\theta}(t) \in \mathbb{R}^{n_\theta}$ is the parameter vector, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(t) \in \mathbb{R}^{n_v}$ is the process noise vector, and $\mathbf{f}$ is the state equation. This state equation can be constructed using physics-based principles, or using data-driven techniques.

While Eq. 1 is used for state prediction, actual sensor measurements (which are available until time $t = t_P$) are used for state estimation. The sensor measurements are modeled using a generic output equation, such as:

$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{n}(t)) \tag{2}$$

where $\mathbf{y}(t) \in \mathbb{R}^{n_y}$, $\mathbf{n}(t) \in \mathbb{R}^{n_n}$, and $\mathbf{h}$ denote the output vec-

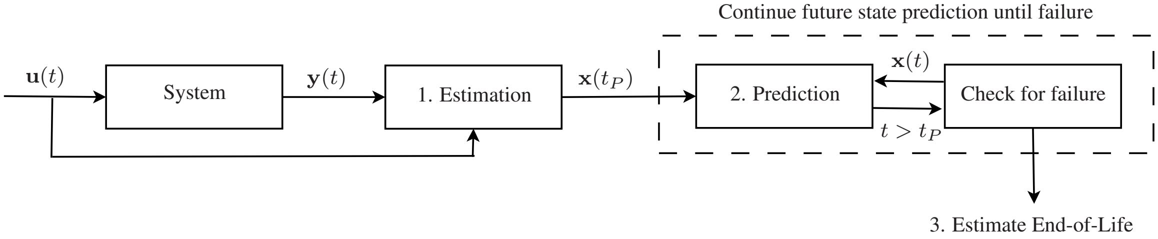


Figure 1. Model-Based Prognostics Architecture

tor, measurement noise vector, and output equation respectively. Note that output measurements are available only until time $t = t_P$. Therefore, the output equation is used only in the estimation stage, and not in the prediction stage.

(Note: While the above framework for state-space representation is general, some practical systems are time-invariant. Therefore, $\mathbf{f}$ and $\mathbf{h}$ may not depend on t. Sometimes, the output equation $\mathbf{h}$ may depend only on the states $\mathbf{x}(t)$ and not the inputs $\mathbf{u}(t)$. While the proposed methodology is presented using the general framework, simplifications and assumptions may be included depending on the physical system under consideration.)

2.2. State Estimation

Bayesian tracking approaches as Kalman filtering, particle filtering, etc. can be used for state estimation. These methods use Bayes theorem to update the uncertainty in the states continuously as a function of time, as and when new measurements are available. While particle filtering is the most general method that can account for different distribution types and account for non-linearity, Kalman filtering can be used only when Eq. 1 is linear and all the uncertain quantities are Gaussian. When the uncertain quantities are Gaussian, the extended Kalman filter can be applied by linearizing Eq. 1. Similarly, the unscented Kalman filter (Daigle et al., 2012) can be used to approximate the mean and variance of response quantities (future states and remanining useful life prediction), even for non-linear models.

2.3. Prognostics: Future Prediction

Prognostics consists of predicting future behavior of engineering systems, identifying possible failure modes, and estimating the remaining useful life. Estimating system state is an essential precursor to prognostics, because damage and/or system faults can be represented using state variables or a collection of state variables. Therefore, predicting how the damage will progress or how the fault will grow is equivalent to estimating future states of the systems, based on the information available at the prediction time instant t_P. In order predict future behavior and thereby perform prognosis until end-of-life (denoted by $E(t_P) = t + R(t_P)$, where $R(t_P)$ is the remaining useful life), the following pieces of information are necessary:

1. State prediction model, as in Eq. 1.
2. Present state estimate ($\mathbf{x}(t_P)$); using the present state estimate and the state space equations in Eq. 1, the future states ($\mathbf{x}(t_P)$, $\mathbf{x}(t_P+1)$, $\mathbf{x}(t_P+2)$, ..., $\mathbf{x}(t_P+R(t_P))$) can be calculated.
3. Future loading ($\mathbf{u}(t_P)$, $\mathbf{u}(t_P+1)$, $\mathbf{u}(t_P+2)$, ..., $\mathbf{u}(t_P+R(t_P))$); these values are needed to calculate the future state values using the state space equations.
4. Parameter values from time t_P until time $t_P + R(t_P)$ (denoted by $\boldsymbol{\theta}(t_P)$, $\boldsymbol{\theta}(t_P+1)$, ..., $\boldsymbol{\theta}(t_P+R(t_P))$).
5. Process noise ($\mathbf{v}(t_P)$, $\mathbf{v}(t_P+1)$, $\mathbf{v}(t_P+2)$, ..., $\mathbf{v}(t_P+R(t_P))$).

While writing "$\boldsymbol{\theta}(t_P)$, $\boldsymbol{\theta}(t_P+1)$, ..., $\boldsymbol{\theta}(t_P+R(t_P))$", note that unit time discretization has been used for the sake of illustration. During implementation, appropriate time-discretization values need to be selected.

During prognosis, all the future states can be predicted as a function of the above quantities, and in a practical engineering system, all of them may be potentially uncertain. First, the state-prediction model is prone to have modeling errors. Second, the state estimate at time t_P is uncertain as a result of the Bayesian tracking method used for estimation. Third, future loading cannot be precisely known in many applications, and therefore, uncertainty regarding future loading needs to be considered. Fourth, model parameters are usually estimated using filtering; but it is challenging to know future parameter values. In this paper, model parameter values are assumed to be constant over time and precisely known (without uncertainty). Fifth, process noise is conventionally assumed to follow a probability distribution (usually, Gaussian with zero mean and a specified variance), and needs to be accounted for in prognostics.

Since prognostics needs to be performed until failure, a boolean function (Sankararaman et al., 2013) of the states is necessary to define end-of-life of the engineering system. Such a boolean function can be defined as:

$$T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t)) = \begin{cases} 1, & c(\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t)) \leq 0 \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $c(\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t)) \leq 0$ represents failure criterion. There may be multiple failure criteria too (Daigle et al., 2012; Sankararaman et al., 2013) in some cases, and therefore, the definition of T_{EOL} needs to account for all such failure criteria.

In fact, it can be easily shown that T_{EOL} is a function of the above list of quantities, and this functional relationship can be expressed through a graphical flowchart, as shown in Fig. 2.

As seen in Fig. 2, both the End-of-life (EOL, denoted by $E(t_P)$) and the remaining useful life (RUL, denoted by $R(t_P)$) can be calculated as a function of the above list of quantities. Let the function, which predicts the RUL, be denoted as:

$$R = G(\boldsymbol{X}), \tag{4}$$

where $\boldsymbol{X}$ represents the concatenated vector of quantities contained in (1) present state estimates ($\mathbf{x}(t_P)$); (2) future loading values ($\mathbf{u}(t_P)$, $\mathbf{u}(t_P + 1)$, $\mathbf{u}(t_P + 2)$, ..., $\mathbf{u}(t_P + R(t_P))$); and (3) future process noise values. Again, note that unit time discretization has been used for illustration. Since these quantities are uncertain, the problem of estimating the uncertainty in prognostics, and thereby computing the uncertainty in EOL and RUL can be viewed as propagating the uncertainty in $\boldsymbol{X}$ through G (Sankararaman & Goebel, 2013a).

3. Future Loading Conditions

In order to perform efficient prognosis, it is necessary to know what loading conditions the system will experience in the future. However, in many practical applications, it is challenging to be able to precisely predict future loading. Therefore, it is essential to estimate the uncertainty in future loading conditions and incorporate this information in prognostics. Time-series analysis techniques and signal processing methods can be used to represent future loading conditions, continuously as a function of time. Further, different types of engineering application may require different types of techniques for loading characterization and uncertainty representation. Therefore, a good prediction methodology should be able to account for different types of representation.

Three different types of future loading conditions - constant amplitude loading, white noise loading, variable amplitude loading - are discussed in this paper. Uncertainty representation for each of the above types of loading conditions are explained in the remainder of this section. Sample loading trajectories are graphically explained through appropriate illustrations.

3.1. Type I: Constant Amplitude Loading

This is the simplest form of loading, where the value of $\mathbf{u}$ is constant at all time instants. However, the constant value is assumed to be random, and one random variable is sufficient to represent uncertainty in this type of loading condition. Multiple realizations of constant amplitude loading conditions are shown in Fig. 3.

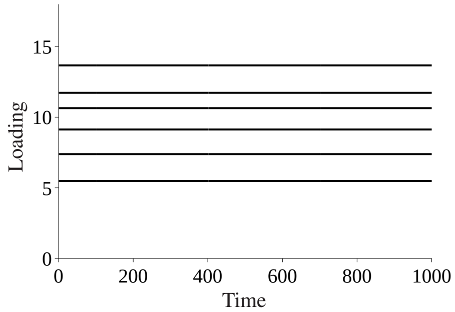


Figure 3. Type-I Loading: Multiple Realizations

3.2. Type II: White Noise Loading

Now, the probability distribution of $\mathbf{u}(t)$ is assumed to be known. For the sake of simplicity, this probability distribution is assumed to be constant at all time instants. Therefore, at any time instant, the loading value is selected by sampling from this probability distribution. Let $f_{\mathbf{U}(t)}(\mathbf{u}(t))$ denote the probability density function (PDF) of this distribution. Loading values at multiple time instants are independently sampled from this distribution. A typical realization of this type of loading condition is shown in Fig. 4.

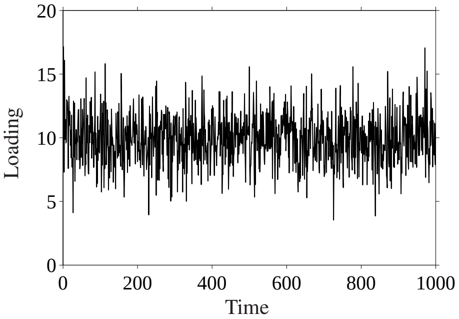


Figure 4. Type-II Loading: One Realization

The number of random variables necessary to represent such a type of loading condition depends on the number of time steps from prediction time t_P until end-of-life $E(t_P)$, which in turn depends on the chosen time-discretization level. Therefore, the number of variables may range from a few tens to several thousands, and this poses a computational challenge for uncertainty propagation. Therefore, a new methodology is proposed in this paper to overcome this challenge.

Note that the value of loading varies from time to time, since it is sampled individually at every time instant. Suppose that the time-variant process is replaced with a time-invariant constant value denoted by $\mathbf{u}^E$. In other words,

$$\mathbf{u}(t) = \mathbf{u}^E \quad \forall \quad t \in [t_P, t_P + R(t_P)] \tag{5}$$

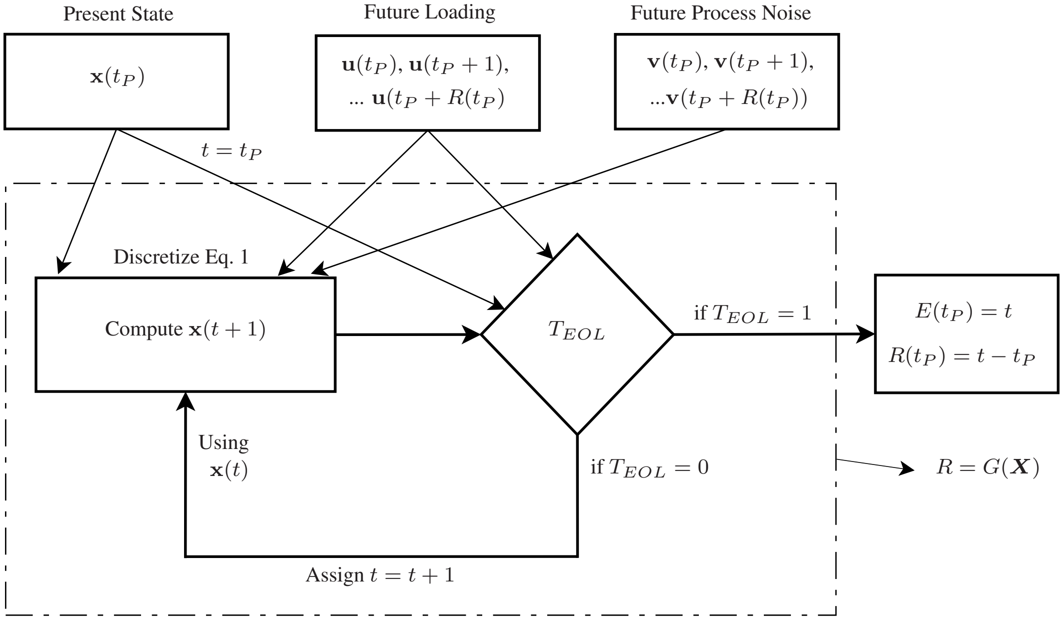


Figure 2. Uncertainty in Prognostics: An Uncertainty Propagation Problem

The above equation means that the same realization of loading will be used for prediction at every future time instant. In order for this to be valid, it is important to choose a suitable probability distribution for $\mathbf{u}^E$, so that the effect of propagating this distribution through G is equivalent to propagating the original distribution of $\mathbf{u}(t)$ through G. This can be accomplished by computing the likelihood of $\mathbf{u}^E$ such that Eq. 5 is satisfied. In other words, any value of $\mathbf{u}^E$ has an associated probability with which Eq. 5 is satisfied; the likelihood of $\mathbf{u}^E$ is proportional to this probability. The probability distribution of the true loading distribution can be used to calculate this likelihood, as:

$$L(\mathbf{u}^E | R(t_P)) \propto \prod_{t=t_P}^{t=t_P+R(t_P)} f_{\mathbf{U}(t)}(\mathbf{u}(t) = \mathbf{u}^E) \qquad (6)$$

where $f_{\mathbf{U}(t)}(\mathbf{v}(t))$ is the probability density function of the true loading $\mathbf{u}(t)$. Also note that the likelihood function is conditioned on the RUL and written as $R(t_P)$. Further, the above equation assumes that the loading values at two different times are independent of each other. If any statistical dependence is unknown, then it can be easily included in the above equation by conditioning appropriately. Having calculated the likelihood, the PDF of $\mathbf{u}^E$ can be calculated as (Sankararaman & Mahadevan, 2011a):

$$f_{\mathbf{U}^E}(\mathbf{u}^E | R(t_P)) = \frac{L(\mathbf{u}^E | R(t_P))}{\int_D L(\mathbf{u}^E | R(t_P)) d\mathbf{u}^E} \qquad (7)$$

In Eq. 7, the domain of integration D is chosen such that $\mathbf{u}^E \in D$ if and only if $L(\mathbf{u}^E) \neq 0$. Now, propagating the uncertainty in $\mathbf{u}^E$ through G is equivalent to propagating the uncertainty in $\mathbf{u}(t)$ through G. Therefore, $\mathbf{u}^E$ is referred to as equivalent time-invariant loading and its distribution is referred to as the equivalent time-invariant loading distribution.

The use of the equivalent time-invariant loading variable reduces the number of variables to the same number as the constant loading case, and therefore facilitates computation for uncertainty propagation. Now, the time-invariant equivalent variable $\mathbf{u}^E$ is used in $\boldsymbol{X}$ in Eq. 4, instead of the true loading values. The corresponding probability distribution (which is actually a function of $R(t_P)$) will be used in uncertainty propagation to calculate uncertainty in RUL, as explained later in Section 4.

Note that the above equivalent time-invariant concept can be used to address process noise (Sankararaman & Goebel, 2013b), since the process noise is also treated as white noise (Gaussian, in several models) in many practical applications.

3.3. Type III: Variable Amplitude Loading

In this type loading, multiple time-windows of varying lengths are considered, and within each time-window, the loading is assumed to be a constant. The number of time-windows is assumed to be known; the length of each time window is assumed to be a random variable, and the constant amplitude for each time window is also considered to be a random variable. Therefore, if there are five time-windows, then ten random variables are needed to represent the entire loading trajectory. Each realization of the set of random variables will correspond to one particular loading trajectory. Multiple realizations of a variable amplitude loading scenario are depicted in Fig. 5.

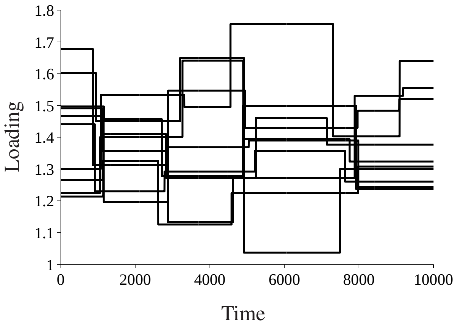


Figure 5. Type-III Loading: Multiple Realizations

Since the number of random variables is small (linear function of number of windows), the proposed uncertainty quantification methodology can be directly applied for this type of loading condition.

4. Most Probable Point Approach for Uncertainty Quantification

This section develops the proposed methodology for quantifying prognostics uncertainty and estimating the uncertainty in the remaining useful life prediction. A generic methodology is presented using the functional relationship $R = G(\boldsymbol{X})$, explained earlier in Eq. 4 in Section 2. It is also explained as to how the methodology can be adapted for the different types of loading conditions discussed in the previous section.

4.1. Most Probable Point Concept

The Most Probable Point (MPP) concept was originally developed by structural engineers to predict the failure probability of structural engineering applications. In this paper, this concept is used for uncertainty quantification in prognostics and remaining useful life prediction.

Consider the functional relationship $R = G(\boldsymbol{X})$. The goal of uncertainty propagation is to compute the uncertainty in R, given the uncertainty in $\boldsymbol{X}$. In order words, the goal is to compute the probability density function (PDF) or cumulative distribution function (CDF) of R, based on the probability distribution of $\boldsymbol{X}$. Let $f_{\boldsymbol{X}}(\boldsymbol{x})$ and $F_{\boldsymbol{X}}(\boldsymbol{x})$ denote the PDF and CDF of $\boldsymbol{X}$ respectively. Note that an upper case letter is used to represent the random variable and a corresponding lower case letter to represent a generic realization of that random variable.

If all the variables $\boldsymbol{X}$ are Gaussian (i.e., normal) and if G is linear, then it can be easily proved that R is also Gaussian, and the statistics of R can be calculated analytically. In the context of prognostics, even if the state-space models and the EOL threshold function are linear, their combination renders G non-linear. Therefore, it is necessary to estimate the distribution of R by considering non-linear functions and non-normal variables since the uncertain quantities may not necessarily follow Gaussian distributions. This is accomplished through a two-step procedure; first, all the uncertain quantities are transformed into Gaussian variables using well-known standard normal transformations, and then, the non-linear function is linearized using first-order Taylor's series expansion, as explained below.

1. **Standard normal transformation**: First, it is necessary to transform all the variables in $\boldsymbol{X}$ to equivalent normal distributions. For the sake of uniformity, all the variables are transformed to the standard normal distribution. There are several two-parameter and three-parameter transformations discussed in the literature (Haldar & Mahadevan, 2000). This paper uses a simple one-parameter transformation. Consider a single random variable X_i (instead of the vector denoted by $\boldsymbol{X}$) with PDF $f_{X_i}(x_i)$ and CDF $F_{X_i}(x_i)$. Then, based on the probability integral transform concept, every x_i can be transformed into a corresponding u_i using the equation:

$$u_i = \Phi^{-1}(F_{X_i}(X_i = x_i)) \tag{8}$$

   where $\Phi^{-1}(.)$ refers to the inverse of the standard normal distribution function (Haldar & Mahadevan, 2000). Now, u_i is function of x_i, and for any chosen distribution for X_i, the corresponding U_i follows the standard normal distribution N(0, 1). Eq. 8 first calculates the CDF which is equivalent to transforming the original variable to the standard *uniform* distribution (upper and lower bounds of 0 and 1 respectively), and then calculates the inverse CDF of the standard normal distribution, thereby transforming to the standard normal distribution. Note that the above transformation is performed for each variable X_i individually, and hence it is not directly applicable if the variables $\boldsymbol{X}$ are statistically dependent or correlated. In such cases, it is necessary to transform $\boldsymbol{X}$ into an uncorrelated standard normal space. Such a transformation can be generically represented as $\boldsymbol{U} = T(\boldsymbol{X})$, and the corresponding inverse transformation can be written as $\boldsymbol{X} = T^{-1}(\boldsymbol{U})$. Several mathematical transforms have been discussed in the literature for this purpose; for example, refer to Liu and Der Kiureghian (Liu & Der Kiureghian, 1986) for details regarding the Morgenstern transform (Morgenstern, 1956) and the Nataf transform (Nataf, 1962).

2. **Linearize "G" using Taylor's Series**: Now that all the variables $\boldsymbol{X}$ have been transformed into equivalent standard normal variables $\boldsymbol{U}$, the next task is to linearize $G(\boldsymbol{X})$ using Taylor's series methodology. Obviously, the point or location of linearization will affect the estimate of the statistics of R, and hence needs to be chosen carefully. The Most Probable Point (MPP) concept guides in choosing the point of linearization, as explained below.

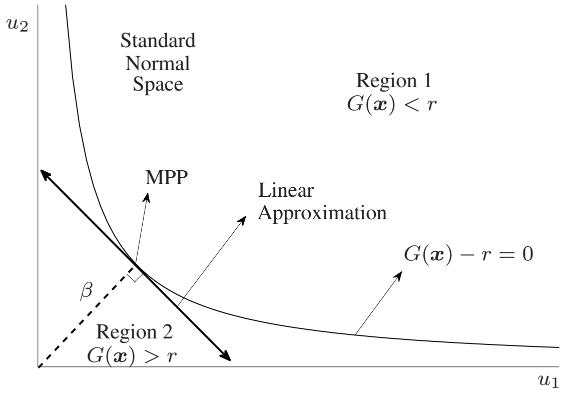

u_2
Standard Normal Space
Region 1 $G(\boldsymbol{x}) < r$
MPP
Linear Approximation
$G(\boldsymbol{x}) - r = 0$
β
Region 2 $G(\boldsymbol{x}) > r$
u_1

Figure 6. Most Probable Point Estimation

Instead of trying to estimate the complete statistics of R by choosing one "global" linearization point, the concept of MPP advocates to solve a "local" problem by focusing on the CDF value at a *particular* realization of R, i.e., $P(R \leq r) = F_R(r) = \eta$. Then, the method linearizes the curve represented by the equation $G(\boldsymbol{x}) - r = 0$, by choosing an appropriate point of linearization. It is evident that an arbitrary location (say, mean of $\boldsymbol{X}$) cannot be chosen as the location of linearization, since it may not even satisfy the above equation for a generic value of r. Therefore, the first condition the point of localization must satisfy is the equation $G(\boldsymbol{x}) - r = 0$. Of the several points that lie on the curve represented by the equation $G(\boldsymbol{x}) - r = 0$, the point which has the highest likelihood of occurrence is chosen and therefore, the point of linearization is called as the Most Probable Point (MPP). The likelihood of occurrence is proportional to the PDF value and the joint PDF needs to be maximized; in practice, this maximization can be easily performed in the standard normal space. In the standard normal space $\boldsymbol{U}$, the closer any point is to the origin, the higher is its likelihood of occurrence. Therefore, the MPP is estimated through an optimization problem by estimating the point on the curve $G(\boldsymbol{x}) - r = 0$ which is closest the origin in the standard normal space, as shown in Fig. 6.

4.2. Calculating the CDF of RUL

The MPP can also be described as the minimum distance (measured from the origin in the standard normal space) point on the curve represented by the equation $G(\boldsymbol{x})-r = 0$. Let β denote this minimum distance, i.e., the distance of MPP from the origin in the standard normal space. Then, it can be easily proved (Haldar & Mahadevan, 2000) that:

$$F_R(r) = P(R \leq r) = \eta = \Phi(-\beta) \tag{9}$$

where $\Phi(.)$ represents the standard normal distribution function. Thus, estimation of MPP directly leads to the calculation of the CDF $F_R(r)$, only locally at $R = r$. In a practical problem, the goal is to compute the uncertainty in R, and therefore, it may not be possible to choose a suitable value for r. Inversely, given the value of η, it is also possible to calculate the value of r which satisfies Eq. 9, using optimization. An iterative, numerical procedure for such an optimization is outlined below:

1. Given a value of η, compute β such that $\eta = \Phi(-\beta)$.
2. Initialize counter $j = 0$ and start with an initial guess for the MPP, i.e., $\boldsymbol{x^j} = \{x_1^j, x_2^j, ... \, x_i^j, ...x_n^j\}$, and a corresponding initial guess for r is obtained.
3. Transform into uncorrelated standard normal space $\boldsymbol{u} = T(\boldsymbol{x})$ and calculate $\boldsymbol{u^j} = \{u_1^j, u_2^j, ... \, u_i^j, ...u_n^j\}$. During this transformation, the original distributions of the variables are used. In the case of process noise, the equivalent time-invariant process noise distribution is used. If Gaussian white process loading is considered, then, the equivalent time-invariant loading distribution is used. Recall that the time-invariant distribution is a function of r, which was calculated in the previous step.
4. Compute the gradient vector in the standard normal space, i.e., $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ...\alpha_n\}$, a column vector where
$$\alpha_i = \frac{\partial G}{\partial u_i} = \frac{\partial G}{\partial x_i} \times \frac{\partial x_i}{\partial u_i} \tag{10}$$
The above derivatives depends both on G, and the chosen transformation T.
5. In the iterative procedure, the next point u^{j+1} is calculated as:
$$\boldsymbol{u^{j+1}} = -\frac{\boldsymbol{\alpha}}{|\boldsymbol{\alpha}|}\beta \tag{11}$$
6. Transform back into the original space using $\boldsymbol{X} = T^{-1}(\boldsymbol{U})$. In other words, compute $\boldsymbol{x^{j+1}}$, and continue starting from Step 3 until the iterative procedure converges. Using tolerance limits δ_1 and δ_2, convergence can be verified if the following two criteria are satisfied: (i) the point lies on the curve of demarcation, i.e., $|G(\boldsymbol{x^j}) - r| \leq \delta_1$; and (2) the solution does not change between two iterations, i.e., $|\boldsymbol{x^{j+1}} - \boldsymbol{x^j}| \leq \delta_2$.

Note that the above iterative algorithm relies on the existence of a unique minimum distance point. If G is non-convex or if there are multiple local minima, then the above algorithm may not be able to identify the optimal MPP. This may happen when the probability distribution of RUL is multimodal. The applicability of the inverse-FORM approach to such cases needs to be investigated in future research. In this paper, it is assumed that the MPP can be identified using the above algorithm, usually within four or five iterations.

Hence, given a value of η, the value of $F_R(r)$ can be calculated using the above algorithm. Note that η denotes a probability level, and by choosing multiple values of η such as

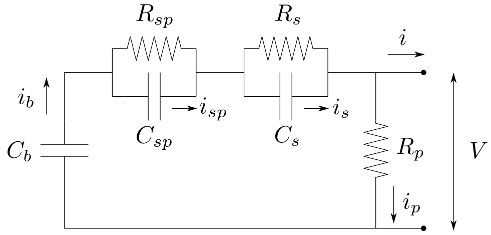


Figure 7. Battery equivalent circuit

0.01, 0.1, 0.2, ... 0.9, 0.99, it is possible to estimate the entire CDF of R. In the context of condition-based monitoring, this procedure is repeated at every time instant prognosis needs to be performed. Note that the proposed methodology is an analytical procedure and can produce repeatable (deterministic) calculations as against Monte Carlo sampling-based approaches. This is an important criteria for existing verification, validation, and certification protocols in the aerospace domain. Further, this methodology requires a few tens of prognostic evaluations (in contrast with several hundreds of evaluations required by sampling-based methods), and therefore, directly aids in real-time, online prognosis.

5. Case Study: Battery Prognostics

The proposed methods are illustrated using a lithium-ion battery that powers an unmanned aerial vehicle (Saha, Quach, & Goebel, 2012) at NASA Langley Research Center. This unmanned aerial vehicle is being used as a test-bed for prognostics and decision-making at NASA Ames Research Center and NASA Langley Research Center.

5.1. Description of the Model

The battery model, extended from that used by Daigle et al. (Daigle et al., 2012) for prognosis, is similar to the models presented by Chen and Rincon-Mora (Chen & Rincon-Mora, 2006). The model is based on an electrical circuit equivalent as shown in Fig. 7, where the large capacitance C_b holds the charge q_b of the battery. The nonlinear C_b captures the open-circuit potential and concentration overpotential. The R_{sp}-C_{sp} pair captures the major nonlinear voltage drop due to surface overpotential, R_s captures the so-called Ohmic drop, and R_p models the parasitic resistance that accounts for self-discharge. This empirical battery model is sufficient to capture the major dynamics of the battery, but ignores temperature effects and other minor battery processes. The governing equations for the battery model are presented in continuous time below. The implementation of the proposed methodology considers a discrete-time version with a discrete time-step of 1 second.

The state-of-charge, SOC, is computed as

$$SOC = 1 - \frac{q_{max} - q_b}{C_{max}}, \tag{12}$$

where q_b is the current charge in the battery (related to C_b), q_{max} is the maximum possible charge, and C_{max} is the maximum possible capacity. The resistance related to surface overpotential is a nonlinear function of SOC:

$$R_{sp} = R_{sp_0} + R_{sp_1} \exp\left(R_{sp_2}(1 - SOC)\right), \tag{13}$$

where R_{sp_0}, R_{sp_1}, and R_{sp_2} are empirical parameters. The resistance, and, hence, the voltage drop, increases exponentially as SOC decreases.

Voltage drops across the individual circuit elements are given by

$$V_b = \frac{q_b}{C_b}, \tag{14}$$

$$V_{sp} = \frac{q_{sp}}{C_{sp}}, \tag{15}$$

$$V_s = \frac{q_s}{C_s}, \tag{16}$$

$$V_p = V_b - V_{sp} - V_s, \tag{17}$$

where q_{sp} is the charge associated with the capacitance C_{sp}, and q_s is the charge associated with C_s. The voltage V_b is also the open-circuit voltage of the battery, which is a nonlinear function of SOC (Chen & Rincon-Mora, 2006). This is captured by expressing C_b as a third-order polynomial function of SOC:

$$C_b = C_{b_0} + C_{b_1} SOC + C_{b_2} SOC^2 + C_{b_3} SOC^3 \tag{18}$$

The terminal voltage of the battery is

$$V = V_b - V_{sp} - V_s. \tag{19}$$

Currents associated with the individual circuit elements are given by

$$i_p = \frac{V_p}{R_p}, \tag{20}$$

$$i_b = i_p + i, \tag{21}$$

$$i_{sp} = i_b - \frac{V_{sp}}{R_{sp}}, \tag{22}$$

$$i_s = i_b - \frac{V_s}{R_s}. \tag{23}$$

The charges are then governed by

$$\dot{q}_b = -i_b, \tag{24}$$

$$\dot{q}_{sp} = i_{sp}, \tag{25}$$

$$\dot{q}_s = i_s. \tag{26}$$

It is of interest to predict the end-of-discharge as defined by a voltage threshold V_{EOD} (16 V, in this example). So, C_{EOL} consists of only one constraint:

$$c_1 : V > V_{EOD}. \tag{27}$$

The parameters of the battery model are assumed to be deterministic and are shown in Table 1. If the parameters are uncertain, and described through probability distributions, then parameter uncertainty can also be easily included, as indicated in Fig. 2.

In Table 1, all voltages are measured in Volts, resistances are measured in Ohms, charges are measured in Coulombs, and capacitances are measured in Coulombs per Volt (or Farads). Note that C_{b_0}, C_{b_1}, C_{b_2}, and C_{b_3} are simply fitting parameters in Eq. 18 and do not have physical meaning.

Table 1. Battery Model Parameters

Parameter	Value	Unit
C_{b_0}	19.80	Farad (F)
C_{b_1}	1745.00	Farad (F)
C_{b_2}	-1.50	Farad (F)
C_{b_3}	-200.20	Farad (F)
R_s	0.0067	Ohm (Ω)
C_s	115.28	Farad (F)
R_p	1×10^4	Ohm (Ω)
C_{sp}	316.69	Farad (F)
R_{sp_0}	0.0272	Ohm (Ω)
R_{sp_1}	1.087×10^{-16}	Ohm (Ω)
R_{sp_2}	34.64	(No unit)
q_{max}	3.11×10^4	Coulomb (C)
C_{max}	30807	Coulomb (C)

The following sections deal with the different sources of uncertainty that affect the RUL prediction, and implement the proposed MPP-based methodology.

5.2. Future Loading Uncertainty

As explained earlier in Section 3, three types of future loading uncertainty are discussed in this paper, and uncertainty quantification results are presented for each type. In this numerical example, loading refers to the current drawn by the battery.

1. **Type-I**: The first type of future loading condition is constant amplitude loading condition. The constant amplitude (in Amps) is considered to be normally distributed ($N(35, 5)$), and this distribution is truncated at a specified lower bound (5.0) and upper bound (80).
2. **Type-II**: The second type of future loading condition is white noise, i.e., at every future time instant the loading value is drawn from a particular distribution. In this paper, the distribution is chosen to be truncated normal ($N(35, 5)$) with a specified lower bound (5.0) and upper bound (80), and the loading values at multiple time instants are considered to be independent of each other. Note that the statistics are identical to that of Type-I loading scenario, but the actual loading trajectory is completely different.
3. **Type-III**: The third type of future loading condition is chosen to be variable amplitude loading with 6 different segments. The time-length (T) of each segment is chosen at random and within each time-segment, the amplitude is considered to be constant; further, the constant amplitude (I) is also chosen randomly. Therefore, there are 12 random variables each of which is assumed to follow a truncated normal distribution. This truncated normal distribution is represented using the mean (μ), standard deviation (σ), lower bound (l) and upper bound (u) of random variable, and the statistics of the 12 random variables are tabulated in Table 2. The six segments in Table 2 correspond to multiple flight segments such as take-off, climb, cruise, landing, etc. Note that the statistics for the amplitude are identical to that of Type-I and Type-II loading, but the actual loading trajectory is completely different.

Such statistics are chosen in order to enable comparison between the three types of loading condition; since the actual amplitudes are similar, it is expected that that overall uncertainty in RUL prediction for each loading scenario should not be wholly different from another.

Table 2. Variable Amplitude Loading: Statistics

Segment	I_μ	I_σ	I_l	I_u	T_μ	T_σ	T_l	T_u
I	35	5	5	80	60	10	50	75
II	35	5	5	80	120	10	90	140
III	35	5	5	80	90	10	70	115
IV	35	5	5	80	120	10	100	145
V	35	5	5	80	90	10	75	120
VI	35	5	5	80	60	10	40	80

5.3. Process Noise Uncertainty

At any time instant, there are three states, and hence three process noise terms. Typically, the statistics of process noise are calculated as a result of the parameter estimationg procedure. In this study, all the three process noise terms were determined to have zero mean and variances equal to 1, 1×10^{-4}, and 1×10^{-6} respectively. For the sake of illustration, it is assumed that the three process noise terms are statistically independent, and further, these process noise values at two different time instants are also statistically independent of each other. In this case, it can be shown that, if the true distribution of the process noise follows a normal distribution with mean 0 and standard deviation σ, then the equivalent time-invariant process noise follows a normal distribution with mean 0 and standard deviation $\frac{\sigma}{\sqrt{R}}$, where R is the remaining useful life prediction calculated using G.

5.4. State Uncertainty

Typically, the state estimation, which is an inverse problem, is addressed using a filtering technique that can continuously estimate the uncertainty in the state when measurements are continuously available as a function of time. In this paper, the focus is on prognostic uncertainty, and the state estimation is not explicitly carried out. The state values are assumed to be available, and the uncertainty in the states are predetermined based on the authors' past experiences with the use of filtering techniques for the above described problem. There are three state variables ((1) charge in C_b; (2) charge in C_{sp}; and (3) charge in C_s) in this example and at any time instant, they are assumed to be normally distributed with a specified mean. First, the mean of the initial states are chosen to be 3.1×10^4, 0, and 0 respectively, and the mean values of the states at other time instants are provided in Fig. 8—10. The standard deviation of the states is chosen to 0.1 times the mean of the states; if the mean is zero, then the standard deviation is chosen to be equal to 0.1.

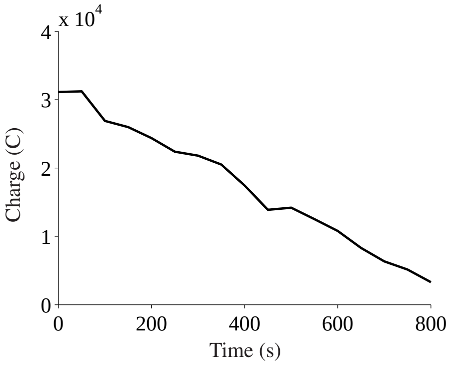


Figure 8. State No. 1: Charge in C_b

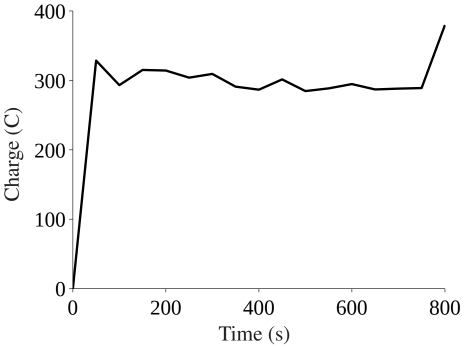


Figure 9. State No. 2: Charge in C_{sp}

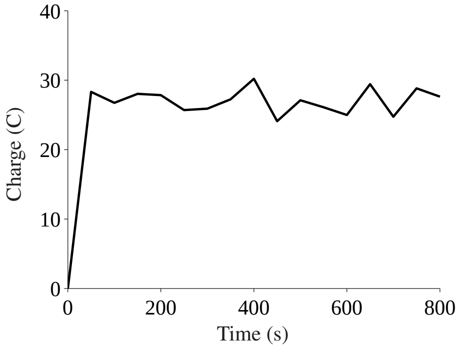


Figure 10. State No. 3: Charge in C_s

5.5. Prognostics Uncertainty Quantification

The results of prognostics uncertainty quantification are discussed in this section. As explained through Fig. 2, the aforementioned sources of uncertainty affect future state prediction and therefore, the remaining useful life prediction. The proposed MPP-based methodology is used to calculate the uncertainty in RUL, and the 90% probability bounds (estimated by repeating the iterative algorithm for $\eta = 0.05$ and $\eta = 0.95$) of RUL corresponding to the three different loading types are plotted in Fig. 11—13.

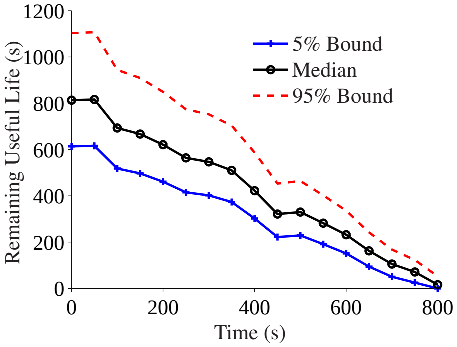


Figure 11. Type-I Loading: 90% Bounds of RUL

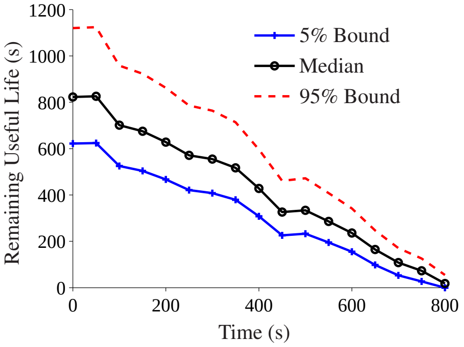


Figure 12. Type-II Loading: 90% Bounds of RUL

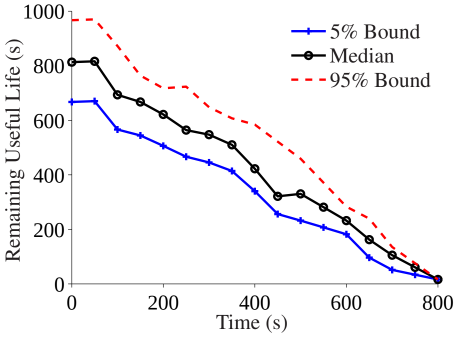


Figure 13. Type-III Loading: 90% Bounds of RUL

Note that the uncertainty is high at initial time instants, and then gradually decreases near the end-of-life of the battery.

Initially, the uncertainty in RUL is high because it is necessary to predict at a farther time instant; future loading and the associated uncertainty need to be considered for a longer period of time. However, at a latter time instant, future loading needs to be assumed for a reduced period of time and hence, the uncertainty in the RUL decreases. In fact, any good prognostic algorithm should depict this behavior, i.e., the prediction of RUL at a later time instant must have lower uncertainty than the prediction at an earlier time instant. From Fig. 11 - 13, it can be seen that the uncertainty in the RUL prediction is similar for three loading cases; particularly, the uncertainty in the case of Type-III loading was observed to be significantly smaller than the other two loading scenarios. For example, the 90% RUL bounds for Type-I, Type-II, and Type-III loading scenarios at the initial time instant ($t_P = 0$ seconds) are [586 1137], [593 1154], and [667 967] respectively. Further, in this example, the RUL prediction was found to be the most sensitive to the first state variable, i.e., charge in C_b. That is why the RUL prediction in Fig. 11 - 13 is similar to the state evolution in Fig. 8.

For the sake of verification, the computation of RUL was also performed using Monte Carlo sampling and the solutions in Fig. 11—13 were compared. While Fig. 11—13 show the RUL values corresponding to $\eta = 0.05$, $\eta = 0.50$, and $\eta = 0.95$, all computations were actually peformed for 13 different η values (0.01, 0.05, 0.1, 0.2 ... 0.9, 0.95, 0.99) in order to construct the entire CDF of RUL. This CDF was compared with the Monte Carlo estimate and the solution from the proposed method was in good agreement with the Monte Carlo estimate. For example, the comparison between Monte Carlo sampling and MPP-based method in the case of the three loading scenarios (at selected time instants) is shown in Fig. 14—16. As seen from these figures, the error in the proposed methodology is very small, with respect to the Monte Carlo solution. Further, while the Monte Carlo solution required thousand evaluations of G, the proposed MPP-based method required much fewer evaluations. The precise number of evaluations for the MPP-based method depends on the selected number of η values and dimension of $\boldsymbol{X}$; typically, favourable results are obtained by using about one-tenth of the number of samples required for Monte Carlo sampling. Considering that the proposed method requires much less computational power than Monte Carlo, and that it may be computationally infeasible to perform Monte Carlo sampling at every time instant, it is evident that the MPP-based procedure provides a good alternative for uncertainty quantification in prognostics.

6. Conclusion

This paper presented a new computational methodology for quantifying uncertainty in prognostics, in the context of condition-based monitoring. First, a model-based computational framework for prognostics was discussed, and the different sources of uncertainty were analyzed in the context of this framework. It was demonstrated that the problem of quantifying uncertainty in prognostics and predicting the end-of-life can be posed as an uncertainty propagation problem. An analytical methodology, based on the Most Probable Point (MPP) concept, was proposed to estimate the uncertainty in end-of-life prediction and thereby the remaining useful life prediction. The Most Probable Point concept was originally developed by structural engineers to compute the failure probability of structural engineering applications, and in this paper, this approach has been extended to quantify prognostics uncertainty.

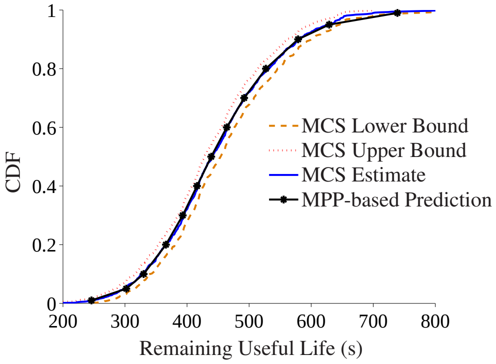


Figure 14. Comparison at Time = 400 s: Type-I Loading

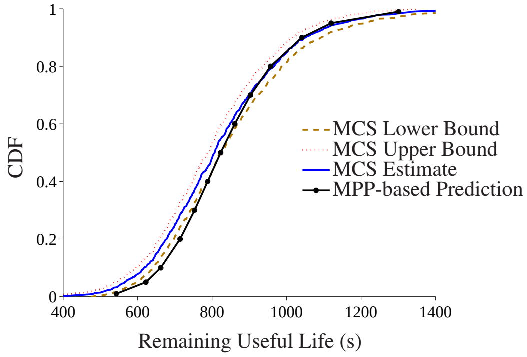


Figure 15. Comparison at Initial Time: Type-II Loading

Further, different types of future loading conditions were discussed for prognostics, and it was explained that the overall prediction methodology may need to be adapted to suit each type of loading condition. In this paper, uncertainty quantification methodology was developed for three types of loading conditions: constant amplitude loading, Gaussian white noise loading, and variable amplitude loading, and demonstrated using a numerical example of a battery used to power an unmanned aerial vehicle. In this paper, it was assumed that the information regarding future loading uncertainty was available. Future work may address the characterization of future loading uncertainty based on possible maneuvers and trajectories, and characterization of model uncertainty. Fur-

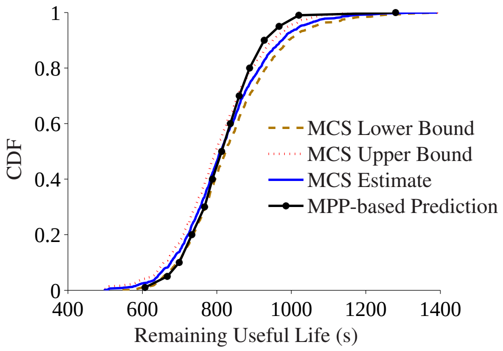


Figure 16. Comparison at Initial Time: Type-III Loading

ther, the applicability of the proposed methodology may also be investigated to practical situations where the probability distribution of the remaining useful life prediction may be multi-modal in nature.

Acknowledgment

The work reported herein was in part funded by the NASA System-wide Satefy Assurance Technologies (SSAT) project under the Aviation Safety (AvSafe) Program of the Aeronautics Research Mission Directorate (ARMD), and by the NASA Automated Cryogenic Loading Operations (ACLO) project under the Office of the Chief Technologist (OCT) of Advanced Exploration Systems (AES). The authors would also like to thank Dr. Matthew Daigle at NASA Ames Research Center for providing the model for the lithium-ion battery.

References

Chen, M., & Rincon-Mora, G. A. (2006). Accurate electrical battery model capable of predicting runtime and I-V performance. *IEEE Transactions on Energy Conversion*, *21*(2), 504 - 511.

Daigle, M., Saxena, A., & Goebel, K. (2012). An efficient deterministic approach to model-based prediction uncertainty estimation. In *Annual Conference of the Prognostics and Health Management Society.*

DeCastro, J. A. (2009). Exact nonlinear filtering and prediction in process model-based prognostics. In *Annual Conference of the Prognostics and Health Management Society.* San Diego, CA..

Farrar, C., & Lieven, N. (2007). Damage prognosis: the future of structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *365*(1851), 623–632.

Farrar, C. R., Lieven, N. A., & Bement, M. T. (2005). An introduction to damage prognosis. *Damage Prognosis: For Aerospace, Civil and Mechanical Systems*, 1–12.

Gu, J., Barker, D., & Pecht, M. (2007). Uncertainty assessment of prognostics of electronics subject to random vibration. In *Aaai fall symposium on artificial intelligence for prognostics* (pp. 50–57).

Guan, X., Liu, Y., Jha, R., Saxena, A., Celaya, J., & Geobel, K. (2011). Comparison of two probabilistic fatigue damage assessment approaches using prognostic performance metrics. *International Journal of Prognostics and Health Management*, *1*, 005.

Haldar, A., & Mahadevan, S. (2000). *Probability, reliability, and statistical methods in engineering design.* John Wiley & Sons, Inc.

Inman, D. J., Farrar, C. R., Junior, V. L., & Junior, V. S. (2005). *Damage prognosis for aerospace, civil and mechanical systems.* Wiley.

Liu, P.-L., & Der Kiureghian, A. (1986). Multivariate distribution models with prescribed marginals and covariances. *Probabilistic Engineering Mechanics*, *1*(2), 105–112.

Morgenstern, D. (1956). Einfache beispiele zweidimensionaler verteilungen. *Mitt. Math. Statist*, *8*(1), 234–235.

Nataf, A. (1962). Détermination des distributions de probabilités dont les marges sont données. *Comptes rendus de lacademie des sciences*, *225*, 42–43.

Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008). Advances in uncertainty representation and management for particle filtering applied to prognostics. In *Prognostics and health management, international conference on* (p. 1 -6).

Saha, B., & Goebel, K. (2008). Uncertainty management for diagnostics and prognostics of batteries using bayesian techniques. In *Aerospace Conference, 2008 IEEE* (p. 1 -8).

Saha, B., Quach, C. C., & Goebel, K. (2012). Optimizing battery life for electric uavs using a bayesian framework. In *Aerospace Conference, 2012 IEEE.*

Sankararaman, S., Daigle, M., Saxena, A., & Goebel, K. (2013). Analytical algorithms to quantify the uncertainty in remaining useful life prediction. In *Aerospace Conference, 2013 IEEE* (pp. 1–11).

Sankararaman, S., & Goebel, K. (2013a). Remaining useful life estimation in prognosis: An uncertainty propagation problem. In *2013 AIAA Infotech@Aerospace Conference.*

Sankararaman, S., & Goebel, K. (2013b). Uncertainty quantification in remaining useful life of aerospace components using state space models and inverse form. In *Proceedings of the 15th Non-Deterministic Approaches Conference.*

Sankararaman, S., Ling, Y., Shantz, C., & Mahadevan, S. (2009). Uncertainty quantification in fatigue damage prognosis. In *Annual Conference of the Prognostics and Health Management Society.*

Sankararaman, S., Ling, Y., Shantz, C., & Mahadevan, S.

(2011). Uncertainty quantification in fatigue crack growth prognosis. *International Journal of Prognostics and Health Management*, *2*(1).

Sankararaman, S., & Mahadevan, S. (2011a). Likelihood-based representation of epistemic uncertainty due to sparse point data and/or interval data. *Reliability Engineering & System Safety*, *96*(7), 814 - 824.

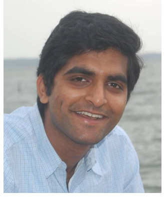
Sankararaman, S., & Mahadevan, S. (2011b). Uncertainty quantification in structural damage diagnosis. *Structural Control and Health Monitoring*, *18*(8), 807–824.

Sankararaman, S., & Mahadevan, S. (2013). Bayesian methodology for diagnosis uncertainty quantification and health monitoring. *Structural Control and Health Monitoring*, *20*(1), 88–106.

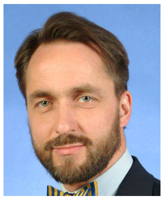
Tang, L., Kacprzynski, G. J., Goebel, K., & Vachtsevanos, G. (2009). Methodologies for uncertainty management in prognostics. In *Aerospace Conference, 2009 IEEE* (pp. 1–12).

Usynin, A., & Hines, J. W. (2007). Uncertainty management in shock models applied to prognostic problems. In *Artificial Intelligence For Prognostics: Papers From The AAAI Fall Symposium.*

BIOGRAPHIES



Shankar Sankararaman received his B.S. degree in Civil Engineering from the Indian Institute of Technology, Madras in India in 2007 and later, obtained his Ph.D. in Civil Engineering from Vanderbilt University, Nashville, Tennessee, U.S.A. in 2012. His research focuses on the various aspects of uncertainty quantification, integration, and management in different types of aerospace, mechanical, and civil engineering systems. His research interests include probabilistic methods, risk and reliability analysis, Bayesian networks, system health monitoring, diagnosis and prognosis, decision-making under uncertainty, treatment of epistemic uncertainty, and multidisciplinary analysis. He is a member of the Non-Deterministic Approaches (NDA) technical committee at the American Institute of Aeronautics, the Probabilistic Methods Technical Committee (PMC) at the American Society of Civil Engineers (ASCE), and the Prognostics and Health Management (PHM) Society. Currently, Shankar is a researcher at NASA Ames Research Center, Moffett Field, CA, where he develops algorithms for uncertainty assessment and management in the context of system health monitoring, prognostics, and decision-making.



Kai Goebel is the Deputy Area Lead for Discovery and Systems Health at NASA Ames where he also directs the Prognostics Center of Excellence. After receiving the Ph.D. from the University of California at Berkeley in 1996, Dr. Goebel worked at General Electrics Corporate Research Center in Niskayuna, NY from 1997 to 2006 as a senior research scientist before joining NASA. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion and his interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He holds 17 patents and has published more than 250 papers in the area of systems health management.

An Energy-Based Prognostic Framework to Predict Fatigue Damage Evolution in Composites

Juan Chiachío[1], Manuel Chiachío[2], Abhinav Saxena[3], Guillermo Rus[4], Kai Goebel[5]

[1,2,4] *University of Granada, Department of Structural Mechanics. Granada, 18071, Spain*
{jchiachio, mchiachio, grus}@ugr.es

[1,2,3] *SGT Inc., NASA Ames Research Center, Intelligent Systems Division, Moffett Field, CA, 94035-1000, USA*
abhinav.saxena@nasa.gov

[5] *NASA Ames Research Center, Intelligent Systems Division, Moffett Field, CA, 94035-1000, USA*
kai.goebel@nasa.gov

ABSTRACT

In this work, a prognostics framework to predict the evolution of damage in fiber-reinforced composites materials under fatigue loads is proposed. The assessment of internal damage thresholds is a challenge for fatigue prognostics in composites due to inherent uncertainties, existence of multiple damage modes, and their complex interactions. Our framework, considers predicting the balance of release strain energies from competing damage modes to establish a reference threshold for prognostics. The approach is demonstrated on data collected from a run-to-failure tension-tension fatigue experiment measuring the evolution of fatigue damage in carbon-fiber-reinforced polymer (CFRP) cross-ply laminates. Results are presented for the prediction of expected degradation by micro-cracks for a given panel with the associated uncertainty estimates.

1. INTRODUCTION

Composites are high-performance materials with a wide range of engineering applications like aerospace, automotive, and construction because of their high specific stiffness and strength in relation to their weight. Most of these applications involve components subject to cyclic loadings that make them susceptible to fatigue degradation. Unlike metals, fatigue damage degradation in composites is a complex multi-scale process driven by several internal fracture mechanisms such as matrix-cracks, local and global delaminations, fiber breakage, etc. The progression of these damage modes and their interactions are significantly influenced by the material properties, stacking sequence, stress level and other loading conditions (Reifsnider & Talug, 1980; R. D. Jamison, Schulte, Reifsnider, & Stinchcomb, 1984; Beaumont, Dimant, & Shercliff, 2006). Among them, transverse matrix cracking[1] holds a central position as a precursor of other damage modes in adjacent plies, such us local delaminations[2] and fiber breakage (Lee, Allen, & Harris, 1989; Beaumont et al., 2006). It is generally accepted that the matrix crack density in off-axis plies tends asymptotically to an upper bounded value corresponding to a spacing of aspect ratio unity[3], termed as Characteristic Damage State (CDS) (Reifsnider & Talug, 1980). This state is usually concomitant with more severe damage scenarios that may lead to a subsequent catastrophic failure.

In addition to the CDS, damage progression may exceed other "subcritical damage states" before ultimate failure, corresponding for instance with the onset of local delaminations or global delaminations respectively. These damage states define tolerance limits that can be used as thresholds for prognostics. However, establishing a deterministic damage progression path to these subcritical states is not an easy task because of the uncertainties in the growth and interactions of internal fracture modes. It is exactly in this context, where the benefits of the proposed prognostic framework can be fully exploited to deal with the uncertainty of the damage accumulation process during fatigue conditions. This framework al-

Juan Chiachío et.al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] The terms matrix micro-cracks, transverse cracks or intralaminar cracks can be invariably used to refer to the cracks growing along fiber directions in off-axis plies.

[2] Local delaminations are small inter-laminar fractures growing from the tips of matrix cracks.

[3] Ratio between average crack spacing $(2l)$ and ply thickness, t.

lows us to sequentially update the information about the existing damage modes by fusing probabilistic information from sensors and models and, ultimately, redefining the path to the subsequent subcritical damage states.

Prognostics is concerned with determining the health of system components and making end of life (EOL) predictions based on meeting specific critical thresholds, given an evolutionary damage model. As with diagnostics, prognostics methods are typically categorized as either model-based or data-driven, depending on whether the damage evolution model is based on physical first principles, or, alternatively uses run-to-failure data to capture trends of degradation. In the recent decades, a large number of articles have been reported to cover data-driven approaches for damage evolution under the topics of stochastic modeling of fatigue damage growth for several applications (Wu & Ni, 2004). First attempts involve modeling the damage evolution as a Markov process (Lin & Yang, 1983). Rowatt and Spanos (1998); Ganesan (2000); Wei, Johnson, and Haj-Ali (2010) proposed Markov chain models to describe the progression of fatigue degradation in composites from laminate compliance measurements, as an extension of the pioneering work of Bogdanoff and Kozin (1985). More recently Chiachio, Chiachio, and Rus (2011) proposed an enhancement of (Rowatt & Spanos, 1998) by a model parameterization that allows accounting for the non-stationarity of the damage process.

On the other hand, model-based approaches provide EOL estimates that are more accurate and precise than data-driven approaches, if good models are available (M. Daigle & Goebel, 2010). Specifically, these models have the ability to adapt to different systems (specimen, materials, conditions, etc.) without much training, they are transparent to human understanding, and furthermore, they can incorporate monitoring data in a Structural Health Monitoring (SHM) context. Particularly in composites, where multiple fracture modes may co-exist, a model-based prognostics framework allows dynamically assessing the dominant damage mode and establishing the thresholds of each of the competing damage modes, by means of a comparison of the energy spent by each single mode (Nairn & Hu, 1992).

In this work, a model-based prognostic framework is formulated to predict in real time the accumulation of damage in composites under fatigue conditions. A tension-tension fatigue experiment in a cross-ply CFRP laminate is used for case study, measuring the evolution of matrix-cracks density. Damage thresholds for remaining useful life (RUL) prediction are sequentially updated by means of a model parameterization of the energy release rate due to microcracking. To this end, a particle filter algorithm (Arulampalam, Maskell, Gordon, & Clapp, 2002; Cappé, Godsill, & Moulines, 2007), is used for the joint state-parameter sequential estimation.

The rest of the paper is organized as follows. Section 2 describes the model of damage propagation and introduces the basis for establishing damage thresholds based on energy release. The transformation of the physics-based evolutionary model into a probabilistic state space model is covered in Section 3. Section 4 formally defines the prognostics problem and describes the algorithmic architecture. Section 5 presents the demonstration of the approach on real data of fatigue considering a cross-ply CFRP laminate. Finally, some concluding remarks are presented in Section 6.

2. Damage propagation models

Behind the versatility of the Markov chain models and other stochastic models for fatigue damage evolution, their lack of physical meaning has been the main source of criticism. Instead, the Paris' law[4] (Paris, Gomez, & Anderson, 1961), which relates the crack growth rate to the range in applied stress intensity factor, has been widely used for being better associated with the physics of the damage growth process. However in contrast to metals, there is no observation of a single crack growth in composites under fatigue loadings but a densification of multiple interlaminar and intralaminar microcracks. The crack tip stress intensity factor is therefore meaningless, instead, the range in microcracking energy release rate (ΔG) can be incorporated within the "traditional" power law formulation, leading to the modified Paris' Law (Nairn & Hu, 1992):

$$\frac{\mathrm{d}\rho}{\mathrm{d}n} = A(\Delta G)^{\alpha} \tag{1}$$

In the equation above, ρ is the crack density which increases with the fatigue cycles n, and A and α are fitting parameters. The energy release rate ΔG represents the energy released due to the formation of a new crack between two existing cracks for a specific stress amplitude: $\Delta G = G_{|\sigma_{max}} - G_{|\sigma_{min}}$. This energy is intimately connected with the effective laminate stiffness, i.e. the stiffness due to the current damage state, which requires a number of theoretical assumptions to be made for composite materials and needs the hard to get information about the geometrical crack pattern per ply, damage extension, etc (Talreja & Singh, 2012). See the next section for further information. It follows that analytical expressions for ΔG are quiet complex, hence a closed-form solution for the modified Paris' Law is difficult to obtain. To overcome this drawback, the resulting differential equation is solved by approximating the derivative by finite differences as:

$$\frac{\Delta\rho}{\Delta n} = \frac{\rho_n - \rho_{n-1}}{1} = A\left(\Delta G(\rho_{n-1})\right)^{\alpha} \tag{2}$$

hence

$$\rho_n = \rho_{n-1} + A\left(\Delta G(\rho_{n-1})\right)^{\alpha} \tag{3}$$

[4] $\frac{\mathrm{d}l}{\mathrm{d}n} = A(\Delta K)^{\alpha}$, where l: crack length, A and α: fitting parameters, and ΔK: stress intensity factor.

2.1. Energy release rate

Several families of models can be found in the composites literature addressing the relation between the (macro-scale) effective stiffness E, and the (micro-scale) matrix-crack density ρ, which essentially forms the basis for the underlying physics behind the energy release rate, ΔG. Therefore, there is a closed connection between micro-scale damage propagation models (i.e., the modified Paris' Law) and macro-scale stiffness models. These models are generally classified as 1) computational methods, 2) semi-analytical methods and 3) analytical methods. The recent work of Talreja and Singh (2012) provides a thorough review on this topic. The computational models are mostly based on the finite element method (FEM), the finite difference method (FDM), and the boundary element method (BEM). In all these models, a homogenization approach is needed to incorporate the microstructure of damage within the formulation of the numerical element, which requires several assumptions to be adopted. The main drawback of this kind of models from the prognostic point of view, is that they are very demanding computationally. Surrogate models may alleviate that problem by adopting data-driven techniques but are beyond the scope of this work. The aim of semi-analytical methods relies on improving the numerical predictions of the former by means of adding experimentally-fitted parameters, or by using numerical expansion of some of the model coefficients (Aboudi, Lee, & Herakovich, 1988; Lee & Hong, 1993). However, these approaches are mostly extensions of the previously developed ideas (Talreja & Singh, 2012), and also, the computational cost is still high. The details of these methods are not relevant to the discussions in this paper and the readers are referred to the cited articles and in particular to the critical review provided in Talreja and Singh (2012) and Talreja (2008).

The analytical models are expressed through closed-form relations grounded in the physics of the stress-strain relationships in the presence of damage. Some assumptions about damage distribution are used such as crack pattern, interaction between plies and the presence of local delamination among others. Depending on the level of assumptions, the body of literature classifies them into shear-lag models (1D), variational models (2D), and crack opening displacement models (3D). Among them the variational models are presumably the most referred ones in the literature, because of their efficiency in accounting for the complexity of damage in relation to the assumptions adopted, compared to the rest of cited analytical models. Additionally, this class of models allows considering the interaction between different damage modes, such as matrix cracks, local and global delaminations. Therefore, they are adopted in this work to study the energy release rate term in the Paris' Law, ΔG.

Originally proposed by Hashin (1985), these models establish the relationship between stresses and strains considering a two-dimensional damage pattern of matrix cracks, local delamination, and global delamination (Nairn & Hu, 1992). The released energy is further obtained by integrating the stresses and strains fields over the laminate volume. Equations 4a and 4b provide analytical expressions for the energy release rate for (a) the formation of a new microcrack between two cracks separated an average dimensionless distance of $\bar{l}$, and (b) the growth of local delamination with a dimensionless length δ between two existing microcracks separated by a distance $\bar{l}$.

$$\Delta G_{TC} = C_3 t_{90} \left(\frac{E_x^{(90)}}{E_0} \Delta\sigma_0 \right)^2 \left(2\chi\left(\bar{l}/2\right) - \chi\left(\bar{l}\right) \right) \tag{4a}$$

$$\Delta G_{LD} = C_3 t_{90} \left(\frac{E_x^{(90)}}{E_0} \Delta\sigma_0 \right)^2 \left(\frac{\chi'(0) - \chi'(\bar{l} - \delta)}{2} \right) \tag{4b}$$

For the case of global delamination, the model for the energy release rate is adopted from Nairn and Hu (1992) and (Hosoi, Takamura, Sato, & Kawada, 2011):

$$\Delta G_{GD} = h \left(\frac{\Delta\sigma_0}{E_{eff}} \right)^2 \left(E_x^{(\text{center})} - E_x^{(\text{edge})} \right) \tag{5}$$

See a detailed description of all the terms in these equations in the Nomenclature section. The functions χ, χ' and C_3 are defined in the Appendix. Notice that ΔG_{LD} depends on the magnitude $(\bar{l} - \delta)$, which expresses the separation between the tips of two growing delaminations starting from the tips of the matrix micro-cracks. Without lack of generality, thermal stresses are not considered in Eq. 4 and 5, given that the data used in this paper were collected in a temperature controlled environment.

2.2. Competing damage modes

Due to the dependency upon the laminate stacking sequence, ply properties and experimental conditions for the damage progression, an energy-based framework is used to establish thresholds of damage without much previous experimental evidences. Matrix microcracks are the most plausible damage mode in the first stages of fatigue damage for angle-ply, quasi-isotropic and cross-ply laminates (R. Jamison, 1985). Hence, we focus on predicting the saturation of matrix microcracks and the onset of the subsequent damage mode, which is a priori unknown.

Based on a balance of energies between different plausible damage modes by Eq. 4 and 5, one can address the question of whether the next increment in damage will be through another transverse crack or a different damage mode (i.e., local delamination, global delamination, etc) induced by the existing microcracks (Nairn & Hu, 1992). Figure 1 illustrates this concept through a case study for a cross-ply laminate. See the

Table 1. Ply properties used in the calculations.

Long. Modulus E_x [GPa]	Trans. Modulus E_y [GPa]	In-plane Poisson ν_{xy}	Out-of-plane Poisson ν_{yz}	Shear modulus G_{xy} [GPa]	Out-of-plane-Shear modulus G_{yz} [GPa]	Thickness t [mm]
137.5	8.4	0.309	0.5	6.2	3.092	0.132

material parameters used for this exercise in Table 1. Observe that initially, the energy release for transverse cracks is larger than the rest of damage modes. Therefore, matrix microcracks are expected to accumulate at a faster rate at earlier cycles. Results also show that until the final stage of the process, the local delamination mode of damage releases more strain energy than the global delamination mode. Therefore, local delaminations are expected much earlier than global delaminations along the fatigue damage process. These conclusions agree with the experimental evidence obtained for cross-ply laminates (R. Jamison, 1985). Observe also that the point where TC and LD curves intersect defines a critical value for the matrix-crack density. At this point local delaminations are more likely to appear than another matrix crack. This point can be computed from Equation 4b using $\delta = 0$. Another important reference-point is the intersection of the LD and GD curves, where local delaminations are supposed to cease starting a global delamination process. Notice that global delamination is unlikely to appear since the energy release rate associated to its growth is too small. Therefore, the required damage for global delamination onset is so severe, that the specimen usually fail before that point.

Based on this reasoning, prognostic thresholds can be establish by predicting the position of these reference points as the fatigue process continues. It is important to remark that the energy term ΔG depends on model parameters which are sequentially updated as new data arrives. Hence, the intersection points defining the thresholds dynamically shift their position until a convergence stage. All these aspects are covered in the next sections.

3. Stochastic embedding

Let assume a physical system that may be idealized by an I/O model as:

$$\mathbf{x}_n = g(\mathbf{x}_{n-1}, \mathbf{u}_n, \boldsymbol{\theta}_n) + \mathbf{v}_n \tag{6a}$$

$$\mathbf{y}_n = h(\mathbf{x}_n) + \mathbf{w}_n \tag{6b}$$

where g and h are the state equation and observation equation, respectively; $\mathbf{x}_n \in \mathbb{R}^{n_x}$ is the state vector, $\mathbf{u}_n \in \mathbb{R}^{n_u}$ is the input vector and $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ is the model parameter vector. $\mathbf{v}_n$ and $\mathbf{w}_n$ are the process noise vector and measurement noise vector, which can be defined[5] as zero mean Gaussians, $\mathcal{N}(0, \sigma_{v_n})$ and $\mathcal{N}(0, \sigma_{w_n})$ respectively.

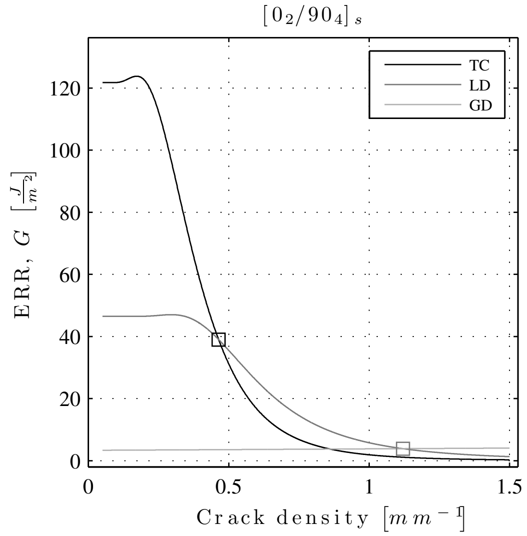


Figure 1. Energy release rate term as a function of the matrix crack density. TC, LD and GD are referring to transverse cracks, local delamination and global delamination, respectively. The two markers correspond to the points where a change in the dominant fracture mode is expected.

The model g is dependent upon a set $\boldsymbol{\theta}$ of model parameters that can be updated to sequentially improve the model predictability as more experimental data are available. In this context, standard Sequential Monte Carlo (SMC) methods (Doucet, De Freitas, & Gordon, 2001) fail and it is necessary to rely on more sophisticated algorithms. Although this problem is still open in the specific literature (Liu & West, 2001; Storvik, 2002; Kantas, Doucet, Singh, & Maciejowski, 2009; Patwardhan, Narasimhan, Jagadeesan, Gopaluni, & Shah, 2012), here we choose the “artificial dynamics” approach (Liu & West, 2001) due to its pragmatism and simplicity, by which model parameters performs a random walk by introducing a small (and decreasing with n) artificial noise term, $\boldsymbol{\xi}_n$, as:

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \boldsymbol{\xi}_n \tag{7}$$

To sequentially reduce the variance of this artificial error sequence $\boldsymbol{\xi}_n$, there are many alternative methods (Kantas et al., 2009), however the recent method proposed by (M. Daigle & Goebel, 2010; M. J. Daigle & Goebel, 2013) is chosen by its efficiency. In our problem, the damage state variable is defined as: $\mathbf{x}_n = \rho_n$, where ρ_n is the matrix crack density at

[5] A rational way to define a probability model for the error term could be to select it such that it produces the most uncertainty (largest Shannon entropy). The maximum-entropy PDF for an unbounded variable given its mean and variance is a Gaussian distribution.

cycle n, i.e, $\rho_n = {}^{1}\!/_{2l_n}$, being l_n the half-separation between cracks at cycle n, expressed in millimeters. Hence Eq. 6 can be rewritten as:

$$\rho_n = g(\rho_{n-1}, \mathbf{u}_n, \boldsymbol{\theta}_n) + v_n \tag{8a}$$

$$\hat{\rho}_n = \rho_n + w_n \tag{8b}$$

where $g(\rho_{n-1}, \mathbf{u}_n, \boldsymbol{\theta}_n)$ is the modified Paris' law. Given that for a particular cycle n the error term v_n is a Gaussian PDF, the probability model for the state transition equation (Eq. 8a) will be also a Gaussian:

$$p(\rho_n|\rho_{n-1}, \mathbf{u}_n, \boldsymbol{\theta}_n) = \mathcal{N}(g(\rho_{n-1}, \mathbf{u}_n, \boldsymbol{\theta}_n), \sigma_{v_n}) \tag{9a}$$

Similarly, the expression for measurement equation (8b) and the artificial evolution of parameters are obtained as follows:

$$p(\hat{\rho}_n|\rho_n) = \mathcal{N}(\rho_n, \sigma_{w_n}) \tag{10a}$$

$$p(\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1}) = \mathcal{N}(\boldsymbol{\theta}_{n-1}, \boldsymbol{\xi}_n) \tag{10b}$$

Finally, to select the set of most sensitive model parameters $\boldsymbol{\theta}$ among the set of parameters that defines the modified Paris' law, a global sensitivity analysis was done following the methodology proposed by (Saltelli et al., 2008). By means of this study, the ply properties $\{E_x, E_y, t\}$ together with the Paris' law fitting parameters $\{A, \alpha\}$ emerged as the key parameters in terms of model output uncertainty. Then the set of updatable parameters was defined by adding the standard deviation of the model error and measurement error to the last choice, i.e., $\boldsymbol{\theta} = \{A, \alpha, E_x, E_y, t, \sigma_v, \sigma_w\}$. The rest mechanical and geometrical parameters act as static non-updatable input parameters, i.e., $\mathbf{u} = \{\sigma_0, B, G_{xy}, G_{yz}, \nu_{xy}, \nu_{yz}\}$.

4. Damage prognostics

For predicting remaining useful life (RUL) of a composite structure, we are interested in predicting the time when the damage grows beyond a predefined acceptable threshold (Saxena, Celaya, Saha, Saha, & Goebel, 2010). The time at which it occurs is known as the expected end of life (EOL). Using the most current knowledge of the system state at cycle n estimated by particle filters (Eq. 11), the goal now is to estimate the EOL$_n$, as probability: $p(EOL_n|\mathbf{y}_n)$. The particle filter computes the joint state-parameter distribution $p(\rho_n, \boldsymbol{\theta}_n|\mathbf{y}_n, \mathbf{u}_n)$, which can be approximated by a set of N discrete weighted *particles*, $\{(\rho_n^i, \boldsymbol{\theta}_n^i), \omega_n^i\}_{i=1}^N$, as

$$p(\rho_n, \boldsymbol{\theta}_n|\mathbf{y}_n) \approx \sum_{i=1}^{N} \omega_n^i \delta(\rho_n - \rho_n^i)\delta(\boldsymbol{\theta}_n - \boldsymbol{\theta}_n^i) \tag{11}$$

In our problem, we consider a sequence of measurements, $\mathbf{y}_n = \hat{\boldsymbol{\rho}}_{0:n}$, where $\hat{\boldsymbol{\rho}}_{0:n} = \{\hat{\rho}_0, \hat{\rho}_1, \ldots, \hat{\rho}_n\}$. Hence, Eq. 11 can be rewritten as:

$$p(\rho_n, \boldsymbol{\theta}_n|\hat{\boldsymbol{\rho}}_{0:n}) \approx \sum_{i=1}^{N} \omega_n^i \delta(\rho_n - \rho_n^i)\delta(\boldsymbol{\theta}_n - \boldsymbol{\theta}_n^i) \tag{12}$$

For simpler notation the conditioning on the model input $\mathbf{u}_n$ is dropped from Eq. 11. Applying Bayes' Theorem, the importance weights ω_n^i can be updated as:

$$\omega_n^i \propto p(\hat{\rho}_n|\rho_n)\omega_{n-1}^i \tag{13}$$

Here we assume that the system model is Markovian of order one and that the observations are conditionally independents of the state. To this end, we use the sampling importance resampling (SIR) particle filter, using systematic resampling (M. Daigle & Goebel, 2010).

4.1. Prognostics threshold

The damage space itself may be defined by means of a set of thresholds $\mathbf{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_c\}$ on more than one critical parameters. In such cases, these thresholds can be combined into a *threshold function* $T_{EOL} = T_{EOL}(\rho, \boldsymbol{\theta})$, that maps a given point in the joint state-parameter space to the Boolean domain $\{0, 1\}$ (M. Daigle & Goebel, 2011). For instance, when a given particle i starting from cycle n performs a random walk and hits any of the thresholds in $\mathbf{C}$, then $T_{EOL}^i = 1$, otherwise $T_{EOL}^i = 0$. The time $n' \geqslant n$ at which that happens defines the EOL$_n$ for that particle. Mathematically:

$$EOL_n^i = \inf\{n' \in \mathbb{N} : n' \geqslant n \wedge T_{EOL}^i(x_{n'}^i, \boldsymbol{\theta}_{n'}^i) = 1\} \tag{14}$$

Using the updated weights at the starting time n, a probabilistic estimation of the EOL is given as:

$$p(EOL_n|\hat{\boldsymbol{\rho}}_{0:n}) \approx \sum_{i=1}^{N} \omega_n^i \delta(EOL_n - EOL_n^i) \tag{15}$$

Once EOL$_n$ is estimated, the remaining useful life can be simply obtained as $RUL_n = EOL_n - n$. An algorithmic description of the prognostic procedure is provided as Algorithm 1.

Algorithm 1 RUL prediction

```
**Requires:** $\{(\rho_n^i, \boldsymbol{\theta}_n^i), \omega_n^i\}_{i=1}^N$, $\mathbf{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_c\}$
**Output:** $\{EOL_n^i, \omega_n^i\}_{i=1}^N$
**for** $i = 1 \to N$ **do**
    Calculate: $T_{EOL}^i(\rho_n^i, \boldsymbol{\theta}_n^i)$
    **while** $T_{EOL}^i = 0$ **do**
        Simulate: $\boldsymbol{\theta}_{n+1}^i \sim p(\cdot|\boldsymbol{\theta}_n^i)$; $\rho_{n+1}^i \sim p(.|\rho_n^i, \boldsymbol{\theta}_{n+1}^i)$
        $n \leftarrow n + 1$
        $(\rho_n^i, \boldsymbol{\theta}_n^i) \leftarrow (\rho_{n+1}^i, \boldsymbol{\theta}_{n+1}^i)$
    **end while**
    $EOL_n^i \leftarrow n$
    $RUL_n^i = EOL_n^i - n$
**end for**
```

5. Results

The proposed framework is applied to fatigue cycling data for cross-ply graphite-epoxy laminates. Torayca T700G unidirectional carbon-prepreg material was used for 15.24 [cm] x 25.4 [cm] coupons with dogbone geometry. See the laminate ply properties in Table 1.

The tests, as reported in (Saxena et al., 2011), were conducted under load-controlled tension-tension fatigue loading with a frequency of $f = 5$ [Hz], a maximum stress of 80% of their ultimate stress, and a stress ratio $R = 0.14$. Lamb wave signals were periodically recorded using a PZT sensor network to estimate internal micro-crack density. A mapping between PZT raw data and micro-crack density was developed following the methodology proposed in Larrosa and Chang (2012).

Results for sequential damage state estimation and prognostics are presented in Figure 2. Based on the reasoning provided in Section 2.1, we initially expect for this laminate that the matrix cracks should saturate around the value of 0.427 $[mm]^{-1}$ (see Figure 1). Observe in Figure 2a that the threshold of matrix cracks density converges from its initial value to 0.419 $[mm]^{-1}$, as model parameters $\boldsymbol{\theta}$ are sequentially updated with new data.

Figures 2b shows the filtered-estimation of matrix micro-cracks together with the sequence of collected data. Every time new data arrive, a damage magnitude is estimated and the updated model is further used to propagate the damage into the future to compute the RUL, calculated as: $RUL_n = EOL_n - n$, using the methodology described in Section 4. These predictions are plotted against time in Figure 2c. Observe in Figure 2c that the RUL prediction is appreciably inaccurate within the first stage of fatigue before the threshold reaches its convergence value. In view of Figure 2a, this stage corresponds to the interval of cycles required for data to train model parameters. From this period, the prediction precision clearly improves with time. We use the two shaded cones of accuracy at 10% and 20% of true RUL, denoted as RUL* to help evaluating the prediction accuracy and precision. Notice also in Figure 2a that accuracy seems to depart from true RUL at the final stage, which indicates that the model and its variance structure do not fully capture the damage dynamics towards the end. Such behavior have been previously reported in (Saxena et al., 2010) and may be related with the asymptotic behavior of the micro-crack evolution, which requires more efficient algorithms for prognostics in such cases.

6. Conclusions

A model-based prognostics framework to predict the fatigue damage evolution in composites is proposed. We consider a stochastically embedded modified Paris' law, as a phenomenological model of damage evolution due to the benefits of the physical meaning of ΔG for estimating the RUL. We demonstrate the validity of this framework on data collected from a tension-tension fatigue experiment using CFRP cross-ply laminate. The following general concluding remarks are extracted:

- By means of balance of energies between different damage modes computed by ΔG, the more plausible damage mode can be elucidate together with the threshold for this mode.

- Due to the model parameterization, the threshold of damage dynamically changes as new data arrives. Hence this approach requires an initial period of cycles before the predictions reach an acceptable accuracy.

- More research effort is need to achieve more efficient prognostic algorithms to improve the accuracy at the final stage of the process, where most of the damage modes in composites typically accumulate in an asymptotic manner.

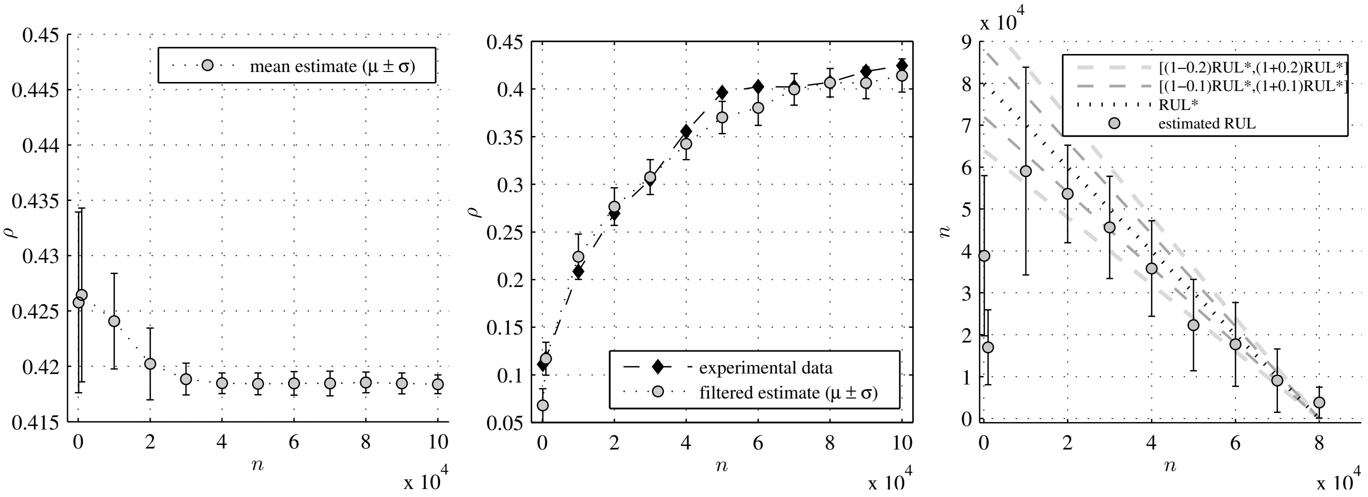
Appendix

The function $\chi(\bar{l})$ in the equations 4a and 4b accounts the change in the effective x-direction modulus caused by the formation of a new micro-crack midway between two existing micro-cracks. Expressions for $\chi(\bar{l})$ and its first derivate are given by:

$$\begin{aligned} \chi(\bar{l}) &= 2\alpha_1\alpha_2(\alpha_1^2+\alpha_2^2)\frac{\cosh(2\alpha_1\bar{l})-\cos(2\alpha_2\bar{l})}{\alpha_2\sinh(2\alpha_1\bar{l})-\sin(2\alpha_2\bar{l})} \\ \chi'(\bar{l}) &= 4\alpha_1\alpha_2(\alpha_1^2+\alpha_2^2)^2\frac{\sinh(2\alpha_1\bar{l})\sin(2\alpha_2\bar{l})}{\alpha_2\sinh(2\alpha_1\bar{l})+\alpha_1\sin(2\alpha_2\bar{l})} \end{aligned} \tag{16a}$$

$$\begin{aligned} \chi(\bar{l}) &= \alpha_1\alpha_2(\alpha_1^2-\alpha_2^2)\frac{\tanh(\alpha_2\bar{l})\tanh(\alpha_1\bar{l})}{\alpha_2\tanh(\alpha_2\bar{l})-\alpha_1\tanh(\alpha_1\bar{l})} \\ \chi'(\bar{l}) &= \alpha_1^2\alpha_2^2(\alpha_1^2-\alpha_2^2)\frac{\frac{\tanh(\alpha_2\bar{l})}{\cosh^2(\alpha_1\bar{l})}-\frac{\tanh(\alpha_1\bar{l})}{\cosh^2(\alpha_2\bar{l})}}{\big(\alpha_2\tanh(\alpha_2\bar{l})-\alpha_1\tanh(\alpha_1\bar{l})\big)^2} \end{aligned} \tag{17a}$$

where $\alpha_1 = \sqrt{\frac{-p}{2}+\sqrt{\frac{p^2}{4}-q}}$ and $\alpha_2 = \sqrt{\frac{-p}{2}-\sqrt{\frac{p^2}{4}-q}}$. The Eq. 16a applies for the case of $\frac{4q}{p^2} > 1$. Otherwise, Eq. 17a should be consider. p and q are relations of the ply properties and the stacking sequence defined by $p = {}^{(C_2 - C_4)}/{C_3}$, $q = {}^{C_1}/{C_3}$. The terms $C_i,\ i : \{1, \ldots, 4\}$,



(a)

(b)

(c)

Figure 2. (a): Trace of the threshold value for crack density saturation over cycles. (b): Sequential estate estimation of microcrack density by the particle filter. At each cycle n, the estimation is calculated using the data available up to that cycle. (c): Remaining useful life (RUL) prediction.

are known functions of the laminate properties defined as:

$$C_1 = \frac{1}{E_x} + \frac{1}{\lambda E_y} \tag{18a}$$

$$C_2 = \left(\lambda + \frac{2}{3}\right)\frac{\nu_{yz}}{E_y} - \frac{\lambda \nu_{xy}}{3E_x} \tag{18b}$$

$$C_3 = (1+\lambda)\left(3\lambda^2 + 12\lambda + 8\right)\frac{1}{60E_y} \tag{18c}$$

$$C_4 = \frac{1}{3}\left(\frac{1}{G_{yz}} + \frac{\lambda}{G_{xy}}\right) \tag{18d}$$

The reader is referred to the Nomenclature section for information of laminate constants involved in the last equations.

ACKNOWLEDGMENT

The two first authors would like to thank the Ministry of Education of Spain for the FPU grants AP2009-4641, AP2009-2390, to the European Union for the “Programa Operativo FEDER de Andalucia 2007-2013” for project GGI3000IDIB and to the Prognostics Center of Excellence at NASA Ames, which kindly hosted them during the course of this work. Authors would also like to thank Structures and Composites lab at Stanford University for experimental data and NASA ARMD/AvSafe project SSAT, which provided partial support for this work.

NOMENCLATURE

h	Laminate half-thickness
t_S	$[S]$ sublaminate thickness
t_{90}	$[90_n]$ sublaminate half-thickness
λ	Ply thickness ratio $\lambda = {}^{t_0}/_{t_{90}}$
$\bar{l}$	average dimensionless half spacing of cracks, $\bar{l} = \frac{l}{t_{90}}$
B	Laminate half-width
δ	Average local delamination length, $\delta = \frac{d}{t_{90}}$
E_x^{90}	Undamaged x-direction $[90_n]$ sublaminate modulus
E_0	Undamaged laminate Young’s modulus
E_{eff}	Damaged x-direction laminate Young’s modulus
$E_x^{(\text{center})}$	Damaged Young modulus of sublaminate from the center side.
$E_x^{(\text{edge})}$	Damaged Young modulus of sublaminate from the edge side.
σ_0	Applied stress

REFERENCES

Aboudi, J., Lee, S., & Herakovich, C. (1988). Three-dimensional analysis of laminates with cross cracks. *ASME, Transactions, Journal of Applied Mechanics*, *55*, 389–397.

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, *50*(2), 174–188.

Beaumont, P., Dimant, R., & Shercliff, H. (2006). Failure processes in composite materials: getting physical. *Journal of Materials Science*, *41*(20), 6526–6546.

Bogdanoff, J., & Kozin, F. (1985). *Probabilistic models of comulative damage.* John Wiley & Sons.

Cappé, O., Godsill, S. J., & Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, *95*(5), 899–924.

Chiachio, J., Chiachio, M., & Rus, G. (2011). An inverse-problem based stochastic approach to model the cumulative damage evolution of composites. *Procedia Engineering*, *14*(0), 1557 - 1563.

Daigle, M., & Goebel, K. (2010). A model-based prognostics approach applied to pneumatic valves. *International Journal of the PHM Society*, *2*(8), 16.

Daigle, M., & Goebel, K. (2011). Multiple damage progression paths in model-based prognostics. In *Aerospace conference, 2011* IEEE (pp. 1–10).

Daigle, M. J., & Goebel, K. (2013). Model-based prognostics with concurrent damage progression processes. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, *43*(3), 535-546.

Doucet, A., De Freitas, N., & Gordon, N. (2001). *Sequential* M*onte* C*arlo methods in practice*. Springer Verlag.

Ganesan, R. (2000). A data-driven stochastic approach to model and analyze test data on fatigue response. *Computers & Structures*, *76*(4), 517–531.

Hashin, Z. (1985). Analysis of cracked laminates: a variational approach. *Mechanics of Materials*, *4*(2), 121–136.

Hosoi, A., Takamura, K., Sato, N., & Kawada, H. (2011). Quantitative evaluation of fatigue damage growth in cfrp laminates that changes due to applied stress level. *International Journal of Fatigue*, *33*(6), 781–787.

Jamison, R. (1985). The role of microdamage in tensile failure of graphite/epoxy laminates. *Composites Science and Technology*, *24*(2), 83–99.

Jamison, R. D., Schulte, K., Reifsnider, K. L., & Stinchcomb, W. W. (1984). Characterization and analysis of damage mechanisms in tension-tension fatigue of graphite/epoxy laminates. *Effects of defects in composite materials, ASTM STP*, *836*, 21–55.

Kantas, N., Doucet, A., Singh, S. S., & Maciejowski, J. M. (2009). An overview of sequential monte carlo methods for parameter estimation in general state-space models. In *15th* IFAC S*ymposium on* S*ystem* I*dentification* (Vol. 15).

Larrosa, C., & Chang, F. (2012). Real time in-situ damage classification, quantification and diagnosis for composite structures. In *Proceedings of the 19th* I*nternational* C*ongress on* S*ound and* V*ibration* (Vol. 15).

Lee, J., Allen, D., & Harris, C. (1989). Internal state variable approach for predicting stiffness reductions in fibrous laminated composites with matrix cracks. *Journal of Composite Materials*, *23*(12), 1273–1291.

Lee, J., & Hong, C. (1993). Refined two-dimensional analysis of cross-ply laminates with transverse cracks based on the assumed crack opening deformation. *Composites Science and Technology*, *46*(2), 157–166.

Lin, Y., & Yang, J. (1983). On statistical moments of fatigue crack propagation. *Engineering Fracture Mechanics*, *18*(2), 243–256.

Liu, J., & West, M. (2001). Combined parameter and state estimation in simulation-based filtering. In A. Doucet, N. Freitas, & N. Gordon (Eds.), *Sequential monte carlo methods in practice* (p. 197-223). Springer New York.

Nairn, J., & Hu, S. (1992). The initiation and growth of delaminations induced by matrix microcracks in laminated composites. *International Journal of Fracture*, *57*(1), 1–24.

Paris, P., Gomez, M., & Anderson, W. (1961). A rational analytic theory of fatigue. *The Trend in Engineering*, *13*, 9–14.

Patwardhan, S. C., Narasimhan, S., Jagadeesan, P., Gopaluni, B., & Shah, S. L. (2012). Nonlinear bayesian state estimation: A review of recent developments. *Control Engineering Practice*, *20*(10), 933 - 953.

Reifsnider, K., & Talug, A. (1980). Analysis of fatigue damage in composite laminates. *International Journal of Fatigue*, *2*(1), 3 - 11.

Rowatt, J., & Spanos, P. (1998). Markov chain models for life prediction of composite laminates. *Structural Safety*, *20*, 117–135.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., et al. (2008). *Global sensitivity analysis: the primer*. Wiley-Interscience.

Saxena, A., Celaya, J., Saha, B., Saha, B., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of the PHM Society*, *1*(1), 20.

Saxena, A., Goebel, K., Larrosa, C., Janapati, V., Roy, S., & Chang, F. (2011). Accelerated aging experiments for prognostics of damage growth in composites materials. In *The 8th* I*nternational* W*orkshop on* S*tructural* H*ealth* M*onitoring,* F.-K. C*hang, editor.* (Vol. 15).

Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *Signal Processing, IEEE Transactions on*, *50*(2), 281–289.

Talreja, R. (2008). Damage and fatigue in composites – A personal account. *Composites Science and Technology*, *68*, 2585–2591.

Talreja, R., & Singh, C. V. (2012). *Damage and failure of composite materials*. Cambridge University Press.

Wei, B.-S., Johnson, S., & Haj-Ali, R. (2010). A stochastic fatigue damage method for composite materials based on Markov chains and infrared thermography. *International Journal of Fatigue*, *32*(2), 350–360.

Wu, W., & Ni, C. (2004). Probabilistic models of fatigue crack propagation and their experimental verification. *Probabilistic Engineering Mechanics*, *19*(3), 247–257.

BIOGRAPHIES

Juan Chiachío is a PhD Fellow at the Department of Structural Mechanics, University of Granada, Spain. His research focus on uncertainty quantification of fatigue damage in composites using Bayesian methods. He receives his Bachelor Degree and Masters Degree in Civil Engineering in 2007. He also receives a Masters Degree in Structural Mechanics in 2011. Prior to joining the University of Granada in 2011, Juan worked as a structural engineer in ”Sener Ingenieria y Sistemas, S.A”, since 2007. His research has been awarded by the National Council of Education through one of the FPU annual fellowships, by the Andalusian Society of promotion of the Talent and by the European Council of Civil Engineers (ECCM) with the Silver Medal prize in the 1st European Contest of Structural Design (2008) with a new concept of all-composite CFRP self-stressed bridge. During the course of his PhD work, Juan has worked as guest scientist at Technical University of Hamburg (Germany), California Institute of Technology (Caltech) and NASA Ames Research Center (USA).

Manuel Chiachío is a PhD Fellow at the Department of Structural Mechanics, University of Granada, Spain. His research focus on uncertainty quantification of fatigue damage in composites using Bayesian methods. He receives his Bachelor Degree and Masters Degree in Civil Engineering in 2007. He also receives a Masters Degree in Structural Mechanics in 2011. Prior to joining the University of Granada in 2011, Manuel worked as a structural engineer in ”Sener Ingenieria y Sistemas, S.A”, since 2007. His research has been awarded by the National Council of Education through one of the FPU annual fellowships, by the Andalusian Society of promotion of the Talent and by the European Council of Civil Engineers (ECCM) with the Silver Medal prize in the 1st European Contest of Structural Design (2008) with a new concept of all-composite CFRP self-stressed bridge. During the course of his PhD work, Manuel has worked as a guest scientist at Technical University of Hamburg (Germany), California Institute of Technology (Caltech) and NASA Ames Research Center (USA).

Guillermo Rus started his research on computational mechanics at the University of Granada (UGR, 1995), where he disputed the PhD thesis on Numerical Methods for Nondestructive Identification of Defects (2001). He applied these experimentally at the NDE Lab at MIT (USA) as a Fulbright Postdoctoral Fellow, rendering novel robust quantitative approaches to ultrasonics monitoring. He started up the NDE Lab at the UGR (`www.ugr.es/ endlab`) as assistant professor in 2003, focusing on bioengineering applications in collaboration with University College London, Universit Paris VI and the Nanomaterials Technology Lab. (Spain), among others. He is also transferring this diagnosis technology to civil engineering for monitoring structural health of advanced materials, such as FRP damage state monitoring in collaboration with NASA, Caltech (USA), T. U. Hamburg (Germany). Rus tenured as associate professor in 2009 at UGR, is the author of 26 SCI papers, 9 books chapters, 3 patents and 18 invited seminars. His research career has been awarded by the Juan Carlos Simo prize for young researchers (Spain, 2007), the Honorary Fellowship of the Wessex Institute of Technology (UK, 2005), Fulbright Fellowship (USA, 2002) and the Excellence PhD award (Granada, 2001).

Abhinav Saxena is a Research Scientist with SGT Inc. at the Prognostics Center of Excellence of NASA Ames Research Center, Moffett Field CA. His research focus lies in developing and evaluating prognostic algorithms for engineering systems using soft computing techniques. He has co-authored more than seventy technical papers including several book chapters on topics related to PHM. He is also a member of the SAE's HM-1 committee on Integrated Vehicle Health Management Systems and IEEE working group for standards on prognostics. Abhinav is editor-in-chief of the International Journal of PHM and has led technical program committees in several PHM conferences. He is also a SGT technical fellow for prognostics. He has a PhD in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta. He earned his B.Tech in 2001 from Indian Institute of Technology (IIT) Delhi, and Masters Degree in 2003 from Georgia Tech. He has been a GM manufacturing scholar and is also a member of several professional societies including PHM Society, SAE, IEEE, AIAA, and ASME.

Kai Goebel is Deputy Area Lead of the Discovery and Systems Health Technology Area at NASA Ames Research Center. He also coordinates the Prognostics Center of Excellence. Prior to joining NASA in 2006, he was a senior research scientist at General Electric Corporate Research and Development center since 1997. Dr. Goebel received his Ph.D at the University of California at Berkeley in 1996. He has carried out applied research in the areas of real time monitoring, diagnostics, and prognostics and he has fielded numerous applications for aircraft engines, transportation systems, medical systems, and manufacturing systems. He holds 17 patents and has co-authored more than 250 technical papers in the field of IVHM. Dr. Goebel was an adjunct professor of the CS Department at Rensselaer Polytechnic Institute (RPI), Troy, NY, between 1998 and 2005 where he taught classes in Soft Computing and Applied Intelligent Reasoning Systems. He has been the co-advisor of 6 Ph.D. students. Dr. Goebel is a member of several professional societies, including ASME, AAAI, AIAA, IEEE, VDI, SAE, and ISO. He was the General Chair of the Annual Conference of the PHM Society, 2009, has given numerous invited and keynote talks and held many chair positions at the PHM conference and the AAAI Annual meetings series. He is currently member of the board of directors of the PHM Society and associate editor of the International Journal of PHM.

Integrated Sensor Arrays based on PiezoPaint^{TM} for SHM Applications

Karl Elkjaer[1], Konstantin Astafiev[2], Erling Ringgaard[3], and Tomasz Zawada[4]

[1,2,3,4]Meggitt A/S, Kvistgaard, 3490, Denmark

karl.elkjaer@meggitt.com
konstantin.astafiev@meggitt.com
erling.ringgaard@meggitt.com
tomasz.zawada@meggitt.com

ABSTRACT

Recent progress in development of new functional materials that are flexible and can be processed at very low temperatures (below 100 °C) opens a new opportunity for applications, such as non-destructive evaluation (NDE), or structural health monitoring (SHM) by applying active materials directly on the structures made out of a variety of materials, e.g. metals (aluminium), plastics, and polymers, including CFRP (Carbon Fibre Reinforced Polymer). This paper presents sensor arrays based on a flexible piezoelectric material – PiezoPaint™. The newly developed material exhibits relatively high sensitivity (d_{33} coefficient up to 45 pC/N), extremely low processing temperatures (< 120 °C), and high compliance in the cured state, enabling direct deposition of acoustic/vibration sensor arrays on structures to be monitored by means of screen- or pad-printing. The printed sensors have been applied for impact detection where four-element arrays and a fully integrated wiring system has been deposited directly on aluminium as well as CFRP plates. The presented results show very good performance in terms of sensitivity, flexibility of usage, and ultra-low weight, making PiezoPaint™ technology an attractive alternative for SHM particularly in aerospace applications.

1. INTRODUCTION

With the potential of replacing scheduled maintenance with as-needed maintenance, increasing vehicle lifetimes, discovering unpredicted damages, and increasing safety, the awareness on prognostics and health management has increased significantly in recent years (Raghavan & Cesnik 2007).

An essential part of prognostics and health management is structural health monitoring (SHM) which has been extensively studied in recent years (Raghavan & Cesnik 2007). The SHM field can be divided into two sub fields, passive and active SHM. Active SHM involves the use of actuators and sensors while passive SHM only applies sensors. Different sensing techniques are applied for SHM including fiber Bragg gratings, accelerometers and piezoelectric transducers (Liu & Nayak, 2012). In aerospace, the most useful SHM techniques are vibration-based approaches and guided wave based approaches (Liu & Nayak, 2012). The biggest challenges for SHM in aerospace in particular, are weight, wiring, and space availability.

Ultrasonic techniques are known to be very powerful and versatile for non-destructive testing of structural components and in the case of plate-like structures they generally rely on Lamb waves (Ringgaard, Zawada, Porchez, Bencheikh & Claeyssen, 2011). The full ultrasonic analysis needed to detect relevant defects is quite complex, since the various modes of the Lamb waves are dispersive and for example an extensive mode conversion may take place at interfaces between the various layers of a sandwich composite structure. In recent years the use of composite materials based on CFRP (carbon-fibre reinforced polymer) has increased dramatically in the aerospace industry. CFRP/honeycomb sandwich structures are characterised by a very high specific strength and stiffness compared to aluminium and other conventional structural materials. An additional advantage is that these materials are damage tolerant within certain limits (Hillger, Szewieczek, Schmidt, Sinapius, Aldave, Bosom, & Gonzalez, 2012). However, from the point of view of structural health monitoring, there are a number of issues complicating the matter. To begin with, such sandwich structures are sensitive to impacts of relatively low energy, and furthermore the resulting damage in the core is generally much more severe than the skin damage.

Typically, bulk piezoelectric materials are used for ultrasound-based NDT and SHM applications, however, such an approach has a number of limitations and drawbacks, such as high weight of the system, difficulties with integration of the sensor system into the structures, low compliance, and limited flexibility of application (Schäfer &

Elkjaer et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Janovsky, 2007). There has been an effort to use other classes of piezoelectric materials in order to address some of the above mentioned challenges, e.g. by using polyvinylidene fluoride (PVDF) family materials (Rao, Bhat, Murthy, Madhav, & Asokan, 2006). They are lightweight and can be applied to large areas. However, these polymer materials typically show low piezoelectric activity, are difficult to integrate with the structures, and pose several practical limitations in terms of wiring and processing, such as limited suitability for commercially available printing techniques. A noticeable degradation of dielectric- and piezoelectric properties of PVDF type films with the time, when the film is exposed to elevated temperatures (50-60 °C) has also been reported recently by Silva, Costa, Sencadas, Paleo, and Lanceros-Méndez (2011), which in turn limits the application of such films in aerospace industry.

In this work the functionality of a novel acoustic sensor array based on a flexible, printable piezoelectric material PiezoPaint™ is demonstrated. PiezoPaint™ technology alleviates or completely eliminates the drawbacks of bulk impact detection systems such as weight, complicated wiring and wave coupling, while offering higher piezoelectric response than PVDF. In addition it is compatible with commercially available printing techniques such as screen-printing and pad-printing. In essence the material is a 0-3 composite consisting of PZT particles and a polymer matrix.

In order to illustrate in practise the unique advantages of directly printed acoustic sensors the four-element sensor array based on PiezoPaint™ printed on an aluminium plate has been used for impact detection. A time difference of arrival (TDOA) algorithm has been applied for processing. Due to the direct integration of the sensors with the structure a very good sensitivity has been obtained, enabling accurate impact detection together with estimation of the impact energy. Kim, DeFrancisci, Chen, Rhymer, Funai, Delaney, Fung, Le, & White (2012) work with three sources of impacts on aircrafts. Impacts from ground service equipment which is usually high energy (10^2 – 10^3 J) high contact area impacts, high velocity ice e.g. hail impacts at in-flight speeds, and low velocity impacts from generic sources e.g. dropped equipment. In this study low velocity impacts were generated to test the demonstrators. Preliminary test results on a CFRP multilayer plate are presented as well.

2. Printable Piezoelectrics

In recent years, a number of attempts have been made to combine piezoelectric bulk materials and polymers, developing ceramic-polymer composite materials (Payo & Hale, 2010), however, in most cases such materials represent bulk composite materials that are not suitable for printing techniques, require high curing temperatures, are rigid in a cured state, and therefore cannot be applied to a variety of substrates, including CFRPs.

In order to keep the advantages of both bulk and polymer piezoelectric materials a new flexible piezoelectric material, PiezoPaint™, has been developed by Meggitt A/S[1]. It has been developed primarily with the aim of keeping high piezoelectric activity while making it compatible with different substrates and structures, including textiles, plastics, metals, composites, paper, etc., and the ability to be applied to large areas by using commercially available printing techniques, including but not limited to pad-, screen-, or stencil printing. An important goal that has been targeted when developing PiezoPaint™ material was also to keep as low a processing temperature as possible, limiting it to 120 °C, making the PiezoPaint™ compatible with more materials including CFRP.

PiezoPaint™ represents a composite material that consists of an organic vehicle (polymer matrix) and a piezoelectric powder, manufactured on the basis of commercially available piezoceramic. Typically, hard PZT materials (e.g. Meggitt's Ferroperm™ Piezoceramics Pz24 or Pz26 piezoceramics) are used for manufacturing PiezoPaint™, with a different volume content of the piezoelectric powder in the polymer matrix, depending on the final application. However, lead-free piezoelectric materials have also been demonstrated, showing reasonable performance of PiezoPaint™ material, manufactured on the basis of these materials (see Table 1).

Normally, PiezoPaint™ is prepared in the form of a paste, which could be applied to a number of different structures by using pad-, screen-, and stencil printing techniques. The viscosity and overall fluidity of the paste can be adjusted, depending on the deposition technique.

Typical properties of different PiezoPaint™ materials, in comparison with PVDF based co-polymer material is shown in Table 1.

Table 1 shows that in comparison with PVDF type co-polymers, PiezoPaint™ materials exhibit higher piezoelectric activity (more sensitive in the case of sensor application), have a higher dielectric constant that can be tuned (by using different ratio between polymer matrix and piezoelectric powder, or by utilizing different type of piezoelectric powder with either high or low dielectric constant), and can be utilized at higher temperatures.

As it has already been mentioned above, the properties of PiezoPaint™ materials can be tuned, depending on the final application, e.g. the dielectric constant or compliance can be adjusted to simplify the impedance matching or feasibility of integration with different structures, where the sensor on the basis of PiezoPaint™ material is to be applied.

[1] Patent pending

The microstructure of PiezoPaint™ material screen-printed on alumina substrate and silver bottom electrode is shown in Figure 1. It is shown that PiezoPaint™ material in a cured state has a relatively dense structure, with very low porosity and very fine grain structure (comparing with other similar composite structures, see e.g. work by Arlt K., and Wegener M., (2010)).

An example of PiezoPaint™ materials deposited onto different substrates (polymer, PCB, fabric) by using commercial screen printing technique is shown in Figure 2. These structures are relatively simple. The PiezoPaint™ is sandwiched between the top and the bottom electrodes. The electrodes are based on a commercially available silver paste that can be applied by using the same printing techniques. However, when the material is deposited onto very rough substrates such as textile, an additional interface layer is necessary.

Table 1. Properties of PiezoPaint^{TM} (PP) materials in comparison with piezoelectric polymer material (Omote et al, 1997)

Material	PVDF	PiezoPaint^{TM} PP-50B	PiezoPaint^{TM} PP-50LF
Type	Co-polymer	PZT Composite	Lead-Free Composite
ρ, g/cm3	1.8	5.2	< 4
T_{op}, °C	< 90	< 150	< 150
ε	10 – 12	125	250
tan δ, %	-	3.0	4.0
d_{33}, pC/N	-38	45	25

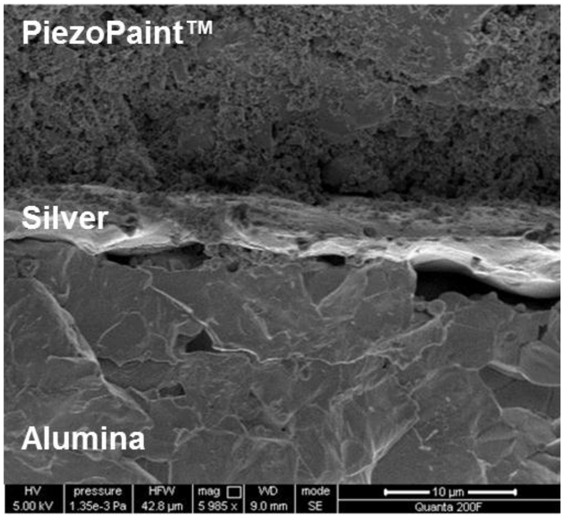


Figure 1. Microstructure (SEM image) of PiezoPaint™ material printed on alumina substrate.

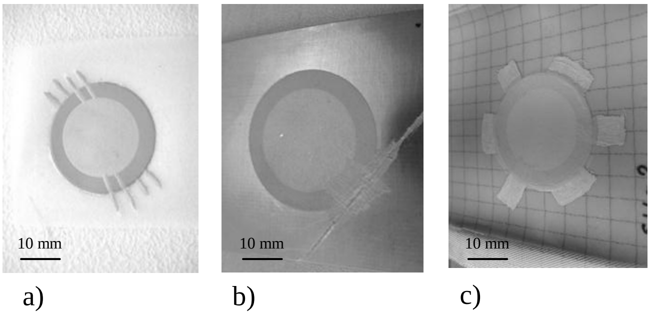


Figure 2. Examples of PiezoPaint™ materials deposited on different substrates: a) – polymer, b) –PCB, and c) – fabric.

3. Sensor Arrays

The test structure consists of circular sensors forming a square array on an aluminium plate. The sensors have been placed at the corners of the plate. The sensor positions can be optimized for impact estimation as in (Staszewski, Worden, Wardle, & Tomlinson, 2000). However, a simpler configuration of the sensors has been chosen as a proof of concept approach. By principle only 3 sensors are needed for simple triangulation but for this demonstration 4 sensors were used. This allows 4 location estimations to be carried out for each impact instead of only 1, making the system more reliable.

3.1. Screen Printing

Screen printing is a well-known pattern transfer technique where a viscous medium is pressed through a woven mesh with a squeegee to generate a sharp edged pattern. The technique is widely used in many branches of industry, e.g. fabric printing industry, but it has also found its way into the electronics industry (hybrid electronics). In this study screen printing was used to print the PiezoPaint^{TM} and the top electrodes, forming the structure of the sensor. The basic process is shown in Figure 3.

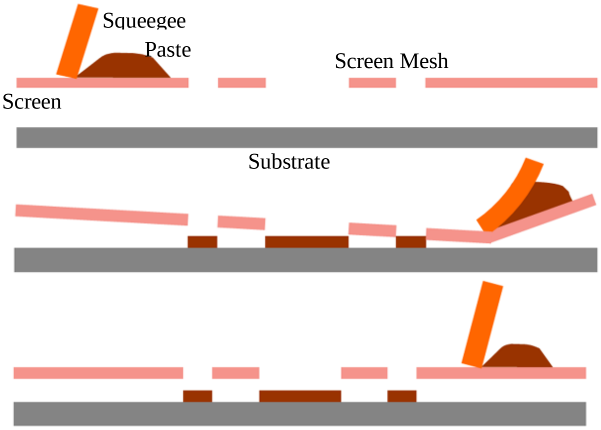


Figure 3. Basic screen printing process.

3.2. Fabrication of demonstrator on aluminium plate

The impact detection demonstrator has been fabricated by printing PiezoPaint^{TM} based sensors at the corners of an aluminium plate (20 cm x 30 cm x 0.5 cm), 2.5 cm from either side.

The sensors have been made as sandwiched structures with the aluminium plate serving as bottom electrode, and a commercial silver paste (Dupont Ag 5028) printed on top of an active layer serving as the top electrode. The diameters of the sensor patches were 12 mm while the top electrode diameters were 10 mm making the active transducer diameter 10 mm. A cross section of the deposited transducers is shown in Figure 4.

The following procedure has been used to fabricate the demonstrator.

- The PiezoPaint^{TM} has been deposited using a commercial screen printer.
- Dupont Ag 5028 has been deposited on top of the PiezoPaint^{TM} patches also using the screen printer.
- Poling has been carried out using a high electrical field (above 1.5. kV/mm) between the top and bottom electrode at an elevated temperature (above 60 °C).
- Wiring has been made at each location using an intermediate printed circuit board (PCB), and conductive epoxy.

The demonstrator with the PCB connectors is shown in Figure 5.

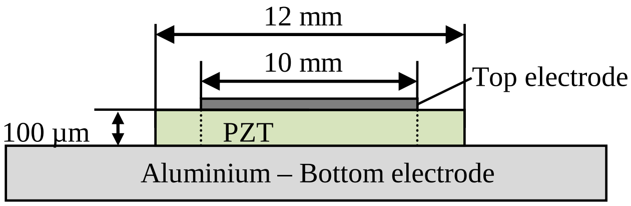


Figure 4. Cross section view of the PiezoPaint^{TM} sensor.

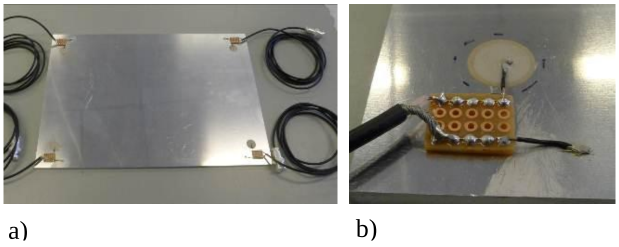


Figure 5. a) The aluminium impact detection demonstrator. b) The PCB used to connect the sensor to a coaxial cable.

4. EXPERIMENTAL PROCEDURE

To evaluate the functionality of the PiezoPaint^{TM} based sensors impact detection experiments have been performed. The experiments have been conducted by connecting the sensor output signal directly to a four-channel oscilloscope (Agilent Infiniium Oscilloscope DSO8064A). Time difference of arrival (TDOA) method has been used to estimate the impact locations.

4.1. Test setup

Impact detection tests were carried out using the setup shown in Figure 6 and Figure 7. Each sensor was connected to a port of the oscilloscope and the oscilloscope was set to take a single waveform acquisition with the trigger set up on channel 1. The impact detection location was estimated after data collection; however, a continuous impact monitoring is possible in real application. In order to control the location and the magnitude of the impacts the following setup was used. A tube with an array of small holes with 1 cm spacing was secured 1 cm above the plate. A pin was used to hold a steel ball in place inside the tube. When the pin was released the ball dropped at a chosen location from a certain height. This setup is shown in Figure 7. The impact energy is estimated by calculating the potential energy of the ball at a certain height and assuming that half of the energy is transferred into the plate at impact, causing deformation and propagating waves.

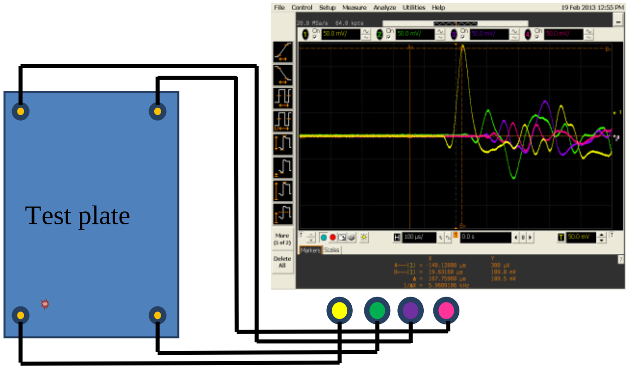


Figure 6. Measurement setup for impact detection experiments.

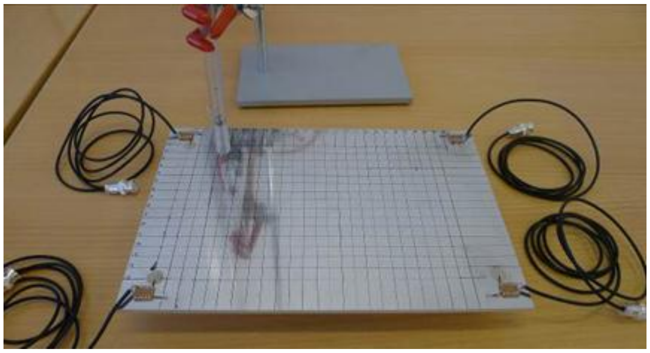


Figure 7. Setup for generating controlled impacts in known locations.

4.2. Time Difference of Arrival

Time difference of arrival (TDOA) is widely used in source localization and finds applications in GPS and cellular location (Patwari, Ash, Kyperountas, Hero III, Moses, & Correal, 2005). It employs three or more receivers to accurately compute the position of an emitter. In this study TDOA is used to estimate the location of an impact. An impact on any surface generates ultrasonic waves that propagate through the medium. The time difference of arrival of these waves at the sensor locations can be used to estimate the impact location. Knowing the speed of sound in the substrate the time differences can be used to calculate distance differences Eq. (1).

$$\Delta t_{21} \cdot v = |\Delta r_{21}| \qquad (1)$$

Where Δt_{21} is the time difference of arrival between sensor 1 and 2, v is the speed of sound in the substrate and Δr_{21} is the difference in distance between the two sensors and the impact.

Eq. (2) shows how the locations of sensor 1, sensor 2 and the impact relate to the distance difference.

$$|\Delta r_{21}| = |r_2| - |r_1| \Leftrightarrow$$

$$|\Delta r_{21}| = \sqrt{(x_2 - x_{im})^2 + (y_2 - y_{im})^2} - \sqrt{(x_1 - x_{im})^2 + (y_1 - y_{im})^2} \qquad (2)$$

Where $|r_1|$ and $|r_2|$ are the distances between the impact and sensors 1 and 2 respectively, and x_1, y_1, x_2, y_2, x_{im}, y_{im} are the x-y components of the sensors and the impact location respectively. Depending on the number of applied sensors a set of equations like Eq. (2) can be created. These equations represent non-linear hyperbolas; however the impact locations can be estimated by applying numerical methods such as the least-squares method (Yang, An, Xu (2008)).

Figure 8 shows the distance between the impact and the sensor and the hyperbolas represented by Eq. (2).

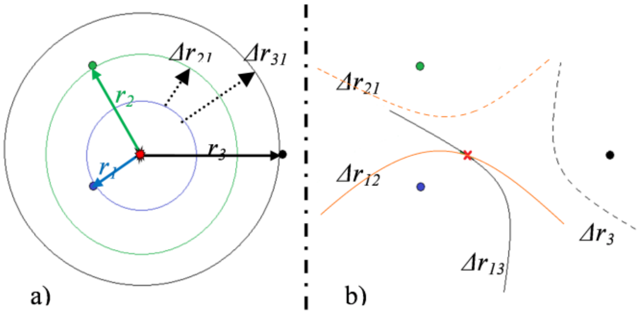


Figure 8. a) 3 receivers and emitter where r_1, r_2 and r_3 are the distances between the emission and the receivers. b) Hyperbolas intersect at the emission location marked by the red x.

4.3. Data Analysis

The TDOA algorithm has been implemented using *Octave*. The program first estimated the arrival times of the propagating waves generated by the impact, at the four sensor locations. To determine when a wave arrived at a sensor a threshold voltage was chosen. The threshold was set to 25 % of the maximum amplitude. This threshold value was chosen based on the best results from impact localization tests. This choice sets a lower limit on the signal to noise ratio required to achieve a useable signal.

$$SNR = \left(\frac{A_{signal}}{A_{noise}}\right)^2 = (4)^2 = 16 \text{ or}$$

$$SNR_{\mathrm{dB}} = 20\log_{10}(4) = 12.04 \text{ dB}$$

The amplitude of the noise was around 15 mV during the measurements which, with a 25 % threshold implied a minimum detectable voltage amplitude at the level of 60 mV. For comparison a 4 g steel ball dropping at the corner furthest away from sensor 1 lead to a maximum voltage amplitude of 100 mV as shown in Figure 9. This corresponds to a signal to noise ratio of 16.48 dB.

After estimating the time of arrival, the script applied the estimated time differences to solve the nonlinear equations numerically and plot the estimated locations with the actual locations.

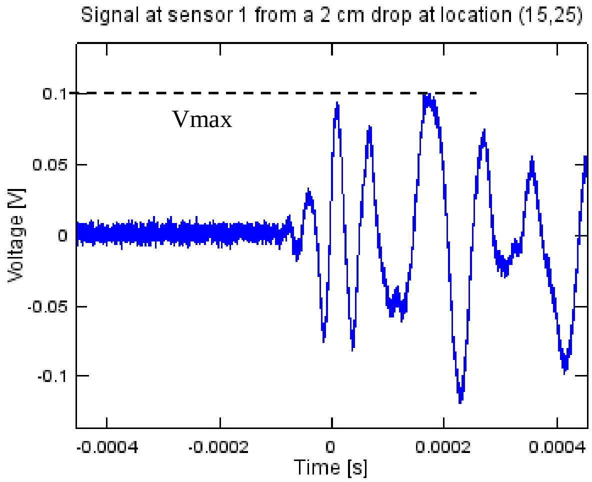


Figure 9. The voltage output at sensor one for a 2 cm drop. The maximum amplitude is 100 mV.

4.4. Experimental Results

4.4.1. Impact Energy

In order to assess the sensitivity of the printed sensors as well as the correlation of the output signal with the impact energy a steel ball with a mass of 4 g was dropped from fixed heights as close to sensor one as possible without landing on top of it. A voltage amplitude reading is shown in Figure 10 and the results from various drop heights are plotted in Figure 11.

Figure 11 shows that a linear fit can be made between the transferred momentum and the voltage amplitude for this specific case with R^2=0.97. The drop height can also be used to estimate the impact energy.

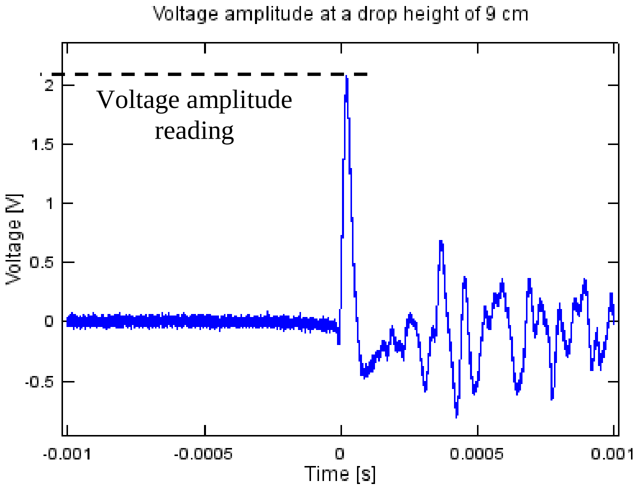


Figure 10. The measured voltage signal (9 cm drop).

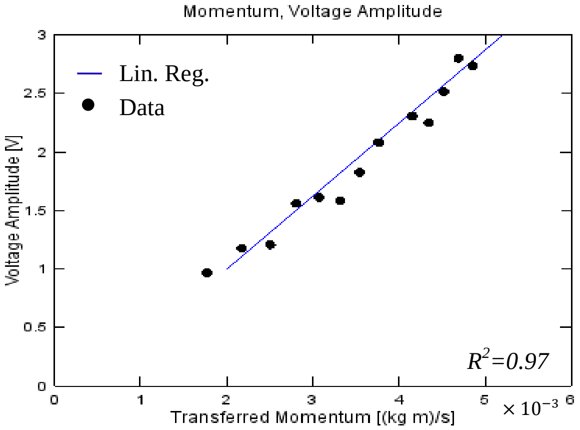


Figure 11. Output voltage amplitude as a function of transferred momentum.

4.4.2. Impact Localisation

In total, 153 impacts were captured across the plate. Results from the initial tests are shown in Figure 13. In Table 2 the average estimation error as a measure of accuracy is shown. More than 90 % of the impacts were located with an error smaller than 5 cm. For comparison Schäfer & Janovsky (2007) work with an impact location accuracy requirement of ~5 cm for hypervelocity impacts on spacecraft. Only 1 impact was discarded as it was completely off the plate.

Table 3 shows the error depending on impact positions and drop height. From Table 3 it is noted that the accuracy tends to depend on the position of the impact. Along the bottom edge of the plate (b in Figure 12) the location estimation is less accurate. Drop height does not seem to affect the accuracy in this specific case. However, the sample size is small and all impact velocities can be considered low (Kim, H. et al., (2012)).

The results bode well for future application, and with optimization of the impact estimation algorithms it is believed that the errors can be reduced significantly.

Table 2. Accuracy of impact location estimations.

	Error <5 cm	Error <7.5 cm	Error <10 cm	Error <12.5 cm	All
#	138	149	151	153	153
%	90.2	97.4	98.9	99.3	100

Table 3. Accuracy of impact location estimations depending on drop height and impact positions. Impact positions are illustrated in Figure 12. "Forced" means the plate was hit by hand with the steel ball.

Positions	Drop Height [cm]	Error < 5 cm [%]	Error < 7 cm [%]
Along line a	3	96 (23/24)	100 (24/24)
Along line b	3	93 (13/14)	93 (13/14)
Along line c	3	86 (12/14)	100 (14/14)
Along line a	7	92 (22/24)	100 (24/24)
Along line b	7	86 (12/14)	86 (12/14)
Along line c	7	86 (12/14)	100 (14/14)
Along line a	15	96 (23/24)	100 (24/24)
Random	15	88 (15/17)	94 (16/17)
Random	Forced	75 (6/8)	100 (8/8)

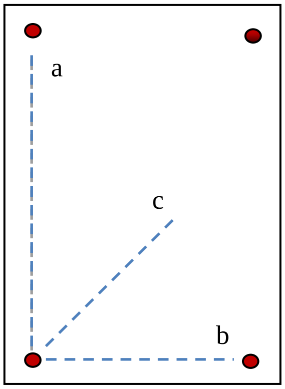


Figure 12. Impact positions used in the drop tests.

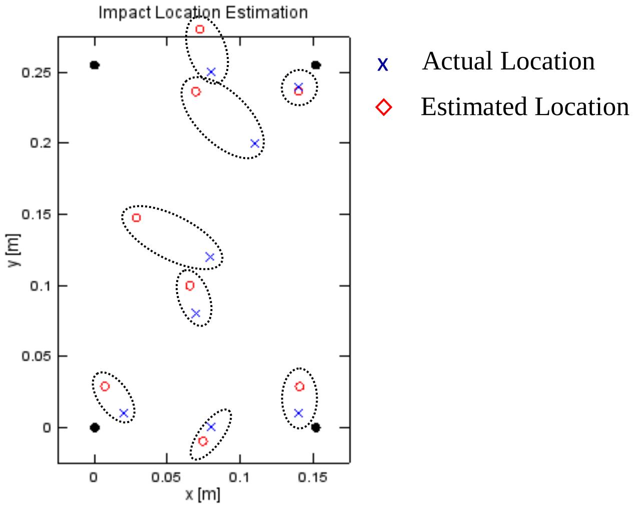


Figure 13. Results from impact localisation of random forced impacts. The dotted cylinders indicate which actual – and estimated location pairs belong together.

4.4.3. CFRP - Preliminary Results

In addition to the aluminium demonstrator a CFRP impact detection demonstrator has been fabricated. The composite plate (40 cm x 40 cm x 1.35 cm) consists of an anti-symmetric honeycomb sandwich structure with a skin thickness of 1 mm and 0.5 mm respectively, and an integrated bronze mesh for lightning protection.

The sensor size and placements are shown in Figure 14. The sensors are placed in the corners of the CFRP plate 4 cm from each side. Instead of connecting at each sensor node electrical connections were painted with Dupont Ag 5028, and wiring was done at a single location as shown in Figure 14. The setup for preliminary impact location tests is shown in Figure 14. Tests have been carried out similarly to the tests of the aluminium demonstrator.

The preliminary results based on 14 impacts with the light hammer are listed in Table 4 and an example of an impact localisation is shown in Figure 15.

Near-future work with this includes energy estimation and the effect on localisation accuracy, effect of impact surface, and effect of impact position on accuracy.

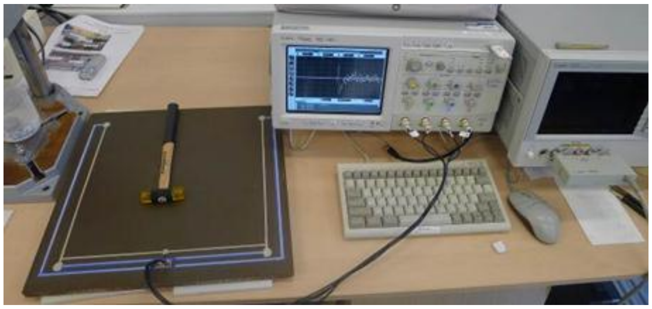


Figure 14. Setup for impact testing on the CFRP plate.

Table 4. Accuracy of impact localisation on CFRP based on the preliminary tests.

	Error < 2.5 cm	Error < 5 cm	All
#	5	14	14
%	35.7	100	100

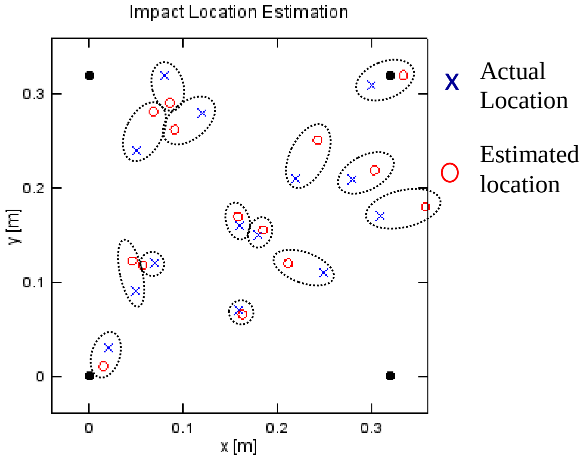


Figure 15. Location estimation on the CFRP plate. The dotted cylinders indicate which actual –and estimated location pairs belong together.

5. Discussion

For an impact detection system to be a viable SHM solution energy estimation is critical. It will enable the system to supply the users with information on the damage extent after impacts, and whether maintenance is needed.

In this study it was shown that the energy can be estimated when the impacts are from a 4 g steel ball at low velocities. However, different geometries and material properties of the impacting objects will influence the generated waves, and thereby the estimated energy.

Thorough studies of impacts from plausible objects and materials e.g. hail, are needed, and should be held up to studies of the damages generated in composites from impacts.

6. Conclusions

- A sensor array based on PiezoPaint™ technology has been successfully developed and demonstrated.
- The sensor system has been directly deposited on aluminium as well as CFRP honeycomb structure, overcoming the typical problems of sensor integration.
- Due to the high sensitivity of the piezoelectric material as well a as the direct contact between the sensing material and the structure the measured output voltage

signal is relatively strong (in the range of volts) and can be directly used for impact detection and localisation without additional signal conditioning.

- According to the experimental data the average detection error was not exceeding 5 cm in the case of an aluminium plate of 20 cm x 30 cm x 0.5 cm and a 4 g steel ball.
- The near–linear dependence of the maximal signal amplitude on the transferred momentum suggests that the printed sensor array can be successfully applied in impact energy estimation, as well.
- Printing of the whole wiring system is a natural extension of the presented technology. The preliminary tests of CFRP sensor array with printed conductors show very good performance.
- It has been shown that the weight and the complexity of SHM system based on acoustical sensors can be significantly reduced by using printed piezoelectric material (PiezoPaint™), while keeping the high sensitivity of the sensors.

Acknowledgement

Karl Elkjaer would like to acknowledge the Meggitt Graduate Programme for supporting the work.

The authors would also like to thank colleagues at Meggitt A/S for their help and valuable feedback.

References

Arlt, K., & Wegener, M., (2010), Piezoelectric PZT/PVDF-copolymer 0-3 composites: Aspects on film preparation and electrical poling, *Dielectrics and Electrical Insulation, IEE Transactions*, 17 (4), 1178-1184, Doi: 10.1109/TDEI.2010.5539688

Baskar Rao, M., Bhat, M.R., Murthy, C.R.L., Venu Madhav, K., & Asokan, S., (2006). Structural health monitoring (SHM) using strain gauges, PVDF film and fiber bragg grating (FBG) sensors: A comparative study, Proc. National Seminar on Non-Destructive Evaluation, Dec.7-9, Hyderabad.

Hillger, W., Szewieczek, A., Schmidt, D., Sinapius, M., Jorge Aldave, I., Venegas Bosom, P. & Vega Gonzalez, L. (2012). Advanced NDT techniques for damage detection in a honeycomb composite helicopter tailboom, Proc. on 5th International Conference on Emerging Technologies of Non-Destructive Testing (ETNDT5), Ioannina, Greece, 19th – 21st September 2011

Kim, H., DeFrancisci, G., Chen, Z. M., Rhymer, J., Funai, S., Delaney, M., Fung, S., Le, J., & White, S., (2012) Impact damage formation on composite aircraft structures, UCSD FAA JAMS Paper, Project Description Paper Supporting Presentation Given at Federal Aviation Administration JAMS 2012 Technical Review Meeting 5 April 2012, Baltimore, MD

Liu, Y., & Nayak, S., (2012). Structural health monitoring: State of the art and perspectives, *JOM* 64 (7), 789-792, Doi: 10.1007/s11837-012-0370-9

Omote, K., Ohigashi, H., & Koga, K., (1997). Temperature dependence of elastic, dielectric, and piezoelectric properties of "single crystalline" films of vinylidene fluoride trifluoroethylene copolymer, *J. Appl. Phys.*, 81, 2760-2769. Doi: 10.1063/1.364300

Patwari, N., Ash, J. N., Kyperountas S., Hero III, A. O., Moses, R. L., & Correal, N. S., (2005). Locating the nodes: Cooperative localization in wireless sensor networks, *IEEE Signal Processing Magazine*, 22 (4), 54-69. Doi: 10.1109/MSP.2005.1458287

Payo, I. & Hale, J.M., Dynamic characterization of piezoelectric paint sensors under biaxial strain, (2010). *Sensors and Actuators A*, 163, 150-158, Doi: 10.1016/j.sna.2010.08.005

Raghavan, A., & Cesnik, E. S., (2007). Review of guided-wave structural health monitoring, *The Shock and Vibration Digest*, 39 (2), 91-114. Doi: 10.1177/0583102406075428

Ringgaard, E., Zawada, T., Porchez, T., Bencheikh, N., & Claeyssen, F., (2011) Multi-element piezo-composite transducers for structural health monitoring using Lamb waves, Proc. on 5th International Conference on Emerging Technologies of Non-Destructive Testing (ETNDT5), Ioannina, Greece, 19th – 21st September 2011

Schäfer, F., & Janovsky, R., (2007). Impact sensor network for detection of hypervelocity impacts on spacecraft, *Acta Astronautica*, 61, 901-911, Doi: 10.1016/j.actaastro.2007.02.002

Silva, M. P., Costa, C. M., Sencadas, V., Paleo, A. J., & Lanceros-Méndez, S., (2011). Degradation of the dielectric and piezoelectric response of β-poly(vinylidene fluoride) after temperature annealing, *Journal of Polymer Research*, 18 (6), 1451-1457. Doi: 10.1007/s10965-010-9550-x

Staszewski, W. J., Worden, K., Wardle, R., & Tomlinson, G. R. (2000). Fail-safe sensor distributions for impact detection in composite materials. *Smart Materials and Structures*, 9 (3), 298-303. Doi:10.1088/0964-1726/9/3/308

Yang, K., An, J., & Xu, Z. (2008), A quadratic constraint total least-squares algorithm for hyperbolic location, *I.J. Communications, Network and System Sciences*, 2, 130-135. Doi: 10.4236/ijcns.2008.12017

Karl Elkjaer received his BSc (2009) and MSc (2011) in Physics and Nanotechnology from the Technical University of Denmark. His master thesis involved designing, fabricating and testing a biochip for investigation of transport through cells and cell membranes. In 2011 he joined the R&D department at Meggitt Sensing Systems Denmark, where he worked on a customer sponsored project concerning high intensity focused ultrasound for treatment of glaucoma. He is currently working on a project concerning structural health monitoring applications of a newly developed material - PiezoPaint™.

Konstantin Astafiev is an R&D Project Manager at Meggitt A/S with the main field of activity in the area of environmentally friendly piezoelectric materials, flexible piezoelectric materials and composites, and piezoelectric thick films. He authored and co-authored over 40 scientific papers and conference articles and holds MSc in Electronics and Microelectronics (2000) and PhD in Materials Science (2004). He has extensive experience in international work, being involved in a number of EC and nationally funded projects.

Erling Ringgaard is Principal Materials Scientist at Meggitt A/S and has more than 15 years of experience in the field of piezoelectric ceramics (M.Sc. in chemistry, PhD in materials science). He authored and co-authored over 60 scientific papers and conference articles. He has participated in a number of EC-funded RTD projects and been the coordinator of two of these. Currently he is the project leader of a nationally funded project on energy harvesting.

Tomasz Zawada, holds MSc degree in Electronic Engineering (2000), PhD degree in Microelectronic Engineering (2004) and MBA degree in Management of Technology (2012). He has more than 10 years of experience in academic and industrial research in the fields of ceramic microsystems, microfluidics, piezoelectric materials and devices. He has authored and co-authored more than 60 scientific publications and reports. Currently, he is working as Engineering and Research Manager at Meggitt A/S, Denmark. His main professional interest is focused on applications of piezoelectric thick films in sensors, transducers and energy harvesting.

Towards Real-time, On-board, Hardware-supported Sensor and Software Health Management for Unmanned Aerial Systems

Johann Schumann[1], Kristin Y. Rozier[2], Thomas Reinbacher[3], Ole J. Mengshoel[4], Timmy Mbaya[5], and Corey Ippolito[6]

[1] *SGT, Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA*
johann.m.schumann@nasa.gov

[2,6] *NASA Ames Research Center, Moffett Field, CA 94035, USA*
Kristin.Y.Rozier@nasa.gov
corey.ippolito@nasa.gov

[3] *Vienna University of Technology, Treitlstrasse 3, 1040 Wien, Austria*
treinbacher@ecs.tuwien.ac.at

[4] *Carnegie Mellon University, Moffett Field, CA 94035, USA*
ole.mengshoel@sv.cmu.edu

[5] *University of Southern California, Los Angeles, CA 90033, USA*
mbaya@usc.edu

Abstract

Unmanned aerial systems (UASs) can only be deployed if they can effectively complete their missions and respond to failures and uncertain environmental conditions while maintaining safety with respect to other aircraft as well as humans and property on the ground. In this paper, we design a real-time, on-board system health management (SHM) capability to continuously monitor sensors, software, and hardware components for detection and diagnosis of failures and violations of safety or performance rules during the flight of a UAS. Our approach to SHM is three-pronged, providing: (1) real-time monitoring of sensor and/or software signals; (2) signal analysis, preprocessing, and advanced on-the-fly temporal and Bayesian probabilistic fault diagnosis; (3) an unobtrusive, lightweight, read-only, low-power realization using Field Programmable Gate Arrays (FPGAs) that avoids overburdening limited computing resources or costly re-certification of flight software due to instrumentation. Our implementation provides a novel approach of combining modular building blocks, integrating responsive runtime monitoring of temporal logic system safety requirements with model-based diagnosis and Bayesian network-based probabilistic analysis. We demonstrate this approach using actual data from the NASA Swift UAS, an experimental all-electric aircraft.

1. Introduction

Modern unmanned aerial systems (UASs) are highly complex pieces of machinery combining mechanical and electrical subsystems with complex software systems and controls, such as the autopilot. Rigorous requirements for safety, both in the air and on the ground, must be met so as to avoid endangering other aircraft, people, or property. Even after thorough pre-flight certification, mission-time diagnostics and prognostics capabilities are required to react to unforeseeable events during operation. In case of problems and faults in components, sensors, or the flight software, the on-board diagnosis capability must be able to detect and diagnose the failure(s) and respond in a timely manner, possibly by triggering mitigation actions. These actions can range from a simple

Johann Schumann et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

mode change to following a pre-programmed flight path (in case of minor problems, such as a lost communications link) to a controlled emergency landing in a remote and safe area (in case of more severe problems).

Most current UAS systems, however, only have very rudimentary fault detection systems. There is a need for advanced health management systems that, in case of anomalies, can quickly and reliably pinpoint failures, carry out accurate diagnosis of unexpected scenarios, and, based upon the determined root causes, make informed decisions that maximize capabilities to meet mission objectives while maintaining safety requirements and avoiding safety hazards.

In this paper, we describe a novel framework for the design and realization of a powerful, real-time, on-board sensor and software health management system that can (a) dynamically monitor a multitude of sensor and software signals; (b) perform substantial reasoning for fault diagnosis; and (c) avoid interfering in any way with the flight software or hardware or impeding on scarce on-board computing resources.

To this end, we have developed a three-pronged approach that combines the capabilities of temporal logic runtime monitors, model-based analysis, and powerful probabilistic reasoning using Bayesian networks (BNs) (Pearl, 1988; Darwiche, 2009). Models are designed using a number of different building blocks for advanced temporal monitoring, model-based filtering, signal processing, prognostics, and Bayesian reasoning. Figure 1 shows a high-level representation of such a model. In this example, raw sensor or software signals are first fed into a smoothing block to weed out sensor noise. Then, one signal is fed into a temporal monitor, which produces a value indicating whether the temporal formula is valid, not valid, or unknown at this point in time. The other signal is fed as a measurement into a Kalman filter. The outputs of both blocks are fed into a Bayesian network block, which performs statistical reasoning and produces posterior probabilities of a fault mode (see also Mengshoel et al., 2008; Ricks & Mengshoel, 2009a, 2009b, 2010; Mengshoel et al., 2010).

The simple example in Figure 1 shows how an SHM capability can be constructed in a scalable, modular, and hierarchical manner and highlights the potential benefit of our three-pronged approach. It separates temporal properties, model-specific properties, and the (time and memory-free) probabilistic reasoning into separate components that are *easy* to model and *efficient* to execute. Our framework encourages this separation of concerns.

In this paper, we discuss in detail the three major building blocks and describe a novel method to implement such a health management system on a Field Programmable Gate Array (FPGA) for highly efficient processing and minimal intrusiveness. We detail how to instrument NASA's Swift UAS with this new SHM capability.

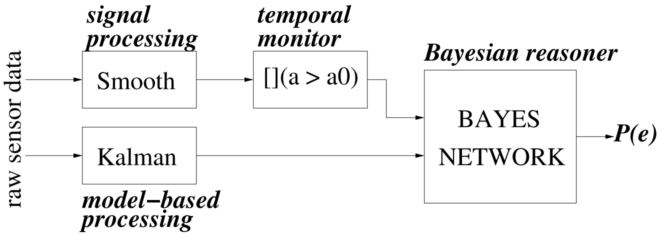


Figure 1. An example instantiation of an SHM model, illustrating one possible interconnection between signal processing, temporal monitoring, model-based processing, and Bayesian reasoning blocks.

Monitoring Sensors and Software. On-board sensor readings are used by on-board software during flight; any flight-time sensor failures should be detected and diagnosed. However, there can be dangerous interactions between the sensors and the software. Perfectly working sensors can trigger software faults, when, for example, operating in an unexpected environment. Alternatively, a faulty sensor can cause unexpected software behavior, e.g., originating from a dormant software bug. Finally, sensor and software failures can trigger issues in entirely different subsystems. For example, a software failure in the navigation system can cause the communication system to fail, as happened when F-22 Raptors crossed the international date-line on their deployment to Kadena in 2007 (Johnson, 2007).[1]

Although pre-deployment verification and validation (V&V) can be very effective in minimizing bugs in on-board software, it is impossible to eliminate all software bugs due to the size and complexity of the software as well as unanticipated, and therefore unmodelled, environmental conditions. The need to catch fault scenarios not detected by pre-deployment V&V is even more pressing when considering software in unmanned systems, since these systems often do not have to undergo the same highly rigorous and costly V&V processes required for manned systems (e.g., according to DO-178C (RTCA, 2012) for commercial transports).

It is therefore mandatory for *both* sensor and software monitoring to be performed during flight, for failure and faults to be detected and diagnosed reliably, and for root-cause analysis to be performed in real time. Only then can appropriate mitigation strategies be activated to recover the UAS or return it to the ground in the safest possible manner.

Temporal and Model-based Data Processing. The collection of all readings of sensors and software state comprises a high-dimensional and fast (around 20Hz or more) real-time data stream that needs to be processed by our health management system. On a high level, our approach to coordinated, multilevel system-wide sensor and software monitor-

[1] See Section 8.3 for a more detailed discussion.

ing transforms this fast, high-dimensional data stream into an informed basis for making intelligent decisions. We discuss a new method for runtime monitoring of temporal logic system safety requirements, in order to enable better probabilistic reasoning compared to what was previously possible. Model-based data processing components include, for example, Kalman filters, Fast Fourier Transforms, or a model-based prognostics component. We can thus reason about sensors, software, and the outputs of prognostics components (e.g., end of useful component life) in a single framework.

Bayesian Reasoning. Our framework uses a Bayesian network to perform diagnostic reasoning and root cause analysis. Although dynamic BNs (DBNs) have, in theory, the capability to directly process high-dimensional time-series data, such an approach may not be realistic in many applications due to scalability limitations and high computational requirements. We therefore *separate* the processing of temporal and model-based aspects of the data from the actual statistical reasoning part.

In order to address practical considerations including sensor-software interdependencies, the demands of real-time temporal and model-based data processing, and Bayesian reasoning for decision making, we present a novel modeling framework for sensor and software health monitoring. The framework separates model-based analysis, temporal monitoring, and statistical reasoning, thus making SHM more efficient, easier to model, and more robust. To enable its application in real-time systems, e.g., on-board of unmanned aerial systems, we will demonstrate how this framework, using temporal logic monitors, model-based preprocessing units, and static Bayesian networks, facilitates modular model design and can be executed highly efficiently on FPGA hardware.

The rest of this paper is structured as follows. After discussing related approaches in Section 2, we introduce our problem domain in Section 3, including the architecture of NASA's Swift UAS and the requirements that must be met for its safe operation. In Section 4, we discuss major design requirements for our approach and present an overview of the building blocks comprising our SHM framework. In the following sections, we give further details of the major components of this framework, namely monitors using temporal logic in Section 5, model-based monitors in Section 6, and Bayesian reasoning components in Section 7. We then provide further details on our implementation of all these components, and discuss experimental results for flight test data from the Swift UAS in Section 8. Section 9 discusses future work and concludes.

2. Related Work

2.1. System Health Management

Vehicle health management performs similar tasks to Fault Detection, Diagnosis, and Recovery (FDDR). There exist many FDDR approaches and (commercial) tools that are being actively used in the aerospace industry. For example, TEAMS[2] is a model-based tool used for diagnosis and test planning. It enables hierarchical, multi-signal diagnosis, but does not model temporal or probabilistic relationships. The underlying paradigm of FACT[3] is fault propagation with temporal constraints. More complex diagnosis systems like HyDE[4] execute simplified dynamical models on various abstraction levels and compare model results against signal values for fault detection and diagnosis. Livingston[5] is a NASA open-source diagnosis and recovery engine that uses a set of high-level qualitative models; the behaviors are specified in temporal logic. Formal V&V for such models have been carried out using the SMV model checker (Lindsey & Pecheur, 2004).

Bayesian networks are also useful for fault detection, diagnosis, and decision making because of their ability to perform deep reasoning using probabilistic models. Likelihood of failures, for example, expressed as mean-time between failure (MTBF), can be cleanly integrated. Whereas there are a number of tools for Bayesian reasoning (e.g., SamIam[6] or Hugin Expert[7]), they have not been used extensively for system health management, in part because of computationally intensive reasoning algorithms.

Fortunately, this situation has started to change. A testbed for electrical power systems in aerospace vehicles, the NASA ADAPT testbed (Poll et al., 2007), has been used to benchmark several system health management techniques. One of them is ProADAPT, a system health management algorithm using Bayesian networks (Ricks & Mengshoel, 2009a, 2009b, 2010). ProADAPT uses compilation of Bayesian networks into arithmetic circuits (Darwiche, 2003; J. Huang, Chavira, & Darwiche, 2006; Chavira & Darwiche, 2007) for efficient sub-millisecond computation. In addition, ProADAPT demonstrates how to diagnose a comprehensive set of faults, including faults of a continuous and dynamic nature, by means of discrete and static Bayesian networks. This work also shows how Bayesian system health models can be generated automatically from electrical power system wiring diagrams (Mengshoel et al., 2008, 2010).

[2] http://www.teamqsi.com/products/teams-designer/

[3] http://w3.isis.vanderbilt.edu/Projects/Fact/Fact.htm

[4] http://ti.arc.nasa.gov/tech/dash/diagnostics-and-prognostics/hyde-diagnostics/

[5] http://ti.arc.nasa.gov/opensource/projects/livingstone2/

[6] http://reasoning.cs.ucla.edu/samiam/

[7] http://www.hugin.com/

2.2. Runtime Verification

Existing methods for Runtime Verification (RV) (Barringer et al., 2010) assess system status by automatically generating (mainly software-based) observers to check the state of the system against a formal specification. Observations in RV are usually made accessible via software instrumentation (Havelund, 2008); they report only when a specification has passed or failed, e.g., through adding hooks in the code base to detect changes in the state of the system being monitored. Such instrumentation may make re-certification of the system onerous, alter the original timing behavior, or increase resource consumption (Pike, Niller, & Wegmann, 2011); we seek to avoid this problem. Also, reporting only the outcomes of specifications does not provide the real-time responsiveness we require for our SHM framework.

Systems in our applications domain often need to adhere to timing-related flight rules like: *after receiving the command "takeoff" reach an altitude of 600 ft within five minutes*. These flight rules can be easily expressed in temporal logics; often in some flavor of Linear Temporal Logic (LTL) (Bauer, Leucker, & Schallhart, 2010). To reduce runtime overhead, restrictions of LTL to its past-time fragment have been used for RV applications previously, mainly due to promising complexity results (Basin, Klaedtke, & Zălinescu, 2011; Divakaran, D'Souza, & Mohan, 2010). Though specifications including past time operators may be natural for some other domains (Lichtenstein, Pnueli, & Zuck, 1985), flight rules like those we must monitor for the Swift UAS require future-time reasoning. To enable more intuitive specifications, others have studied monitoring of future-time claims; see (Maler, Nickovic, & Pnueli, 2008) for a survey and (Geilen, 2003; Thati & Roşu, 2005; Divakaran et al., 2010; Maler, Nickovic, & Pnueli, 2005, 2007; Basin, Klaedtke, Müller, & Pfitzmann, 2008) for algorithms and frameworks. Most of these RV algorithms, however, were designed with a software implementation in mind and require powerful computers that would far exceed the weight, size, power, bandwidth, and other limits of the Swift UAS.

2.3. Hardware Architectures

The above approaches to system health management are typically implemented in software executing on traditional CPUs. However, with the recent developments in parallel computing hardware, including in many-core graphics processing units (GPUs), Bayesian inference can be performed more efficiently (Kozlov & Singh, 1994; Namasivayam & Prasanna, 2006; Xia & Prasanna, 2007; Silberstein, Schuster, Geiger, Patney, & Owens, 2008; Kask, Dechter, & Gelfand, 2010; Linderman et al., 2010; Jeon, Xia, & Prasanna, 2010; Low et al., 2010; Bekkerman, Bilenko, & Langford, 2011; Zheng, Mengshoel, & Chong, 2011; Zheng & Mengshoel, 2013). Several of the recent many-core algorithms are based on the junction tree data structure, which can be compiled from a BN (Lauritzen & Spiegelhalter, 1988; Dawid, 1992; C. Huang & Darwiche, 1994; Jensen, Lauritzen, & Olesen, 1990). Junction trees can be used for both marginal and most probable explanation (MPE) inference in BNs. A data parallel implementation for junction tree inference was developed already in the mid-1990s (Kozlov & Singh, 1994), and the basic sum-product computation has been implemented in a parallel fashion on GPUs (Silberstein et al., 2008). Based on the cluster-sepset mapping method (C. Huang & Darwiche, 1994), node-level parallel computing techniques have recently been developed for GPUs (Zheng et al., 2011; Zheng & Mengshoel, 2013), resulting in as much as a 20-fold speed-up in processing compared to sequential techniques.

Other authors have used the benefits of a hardware architecture to natively answer statistical queries on BNs. For example, Lin, Lebedev, and Wawrzynek (2010) discuss a BN computing machine with a focus on high throughput. Their architecture contains two switching crossbars to interconnect process units with memory. Their implementation, however, targets a resource intensive grid of FPGAs, making this approach unsuitable for our purposes. Kulesza and Tylman (2006) present another approach to evaluate Bayesian networks on reconfigurable hardware. Their approach targets embedded systems as execution platforms and is based on evaluating Bayesian networks through elimination trees. The major drawback of their approach is that the hardware structure is tightly coupled with the elimination tree and requires that the hardware be re-synthesized with every change in the BN.

3. System Background

Due to the increasing interest in using unmanned aircraft for different military, civilian, and scientific applications, NASA has been engaged in UAS research since its inception. The Swift aircraft was designed to support NASA's research interests in aeronautics and earth science—particularly in autonomy, intelligent flight control, and green aviation. For safe operation, the UAS must meet a large number of requirements that in large part come from NASA and FAA processes and standards. In the following, we will briefly describe the characteristics of the Swift UAS and discuss types of safety requirements and flight rules.

3.1. The NASA Swift UAS

For full scale flight testing of new UAS concepts, the NASA Ames Research Center has developed the Swift UAS (Ippolito, Espinosa, & Weston, 2010), a 13 meter wingspan all-electric experimental platform based upon a high-performance sailplane (Figure 2). Swift has a full-featured flight computer and control for sensor payloads. The individual components are connected via a common bus interface and running a C/C++ reflection architecture, which provides a component-based plug-and-play infrastructure. Typical sensors include barometric altitude sensor, airspeed indicator, GPS, and a laser altimeter to measure the altitude above ground.

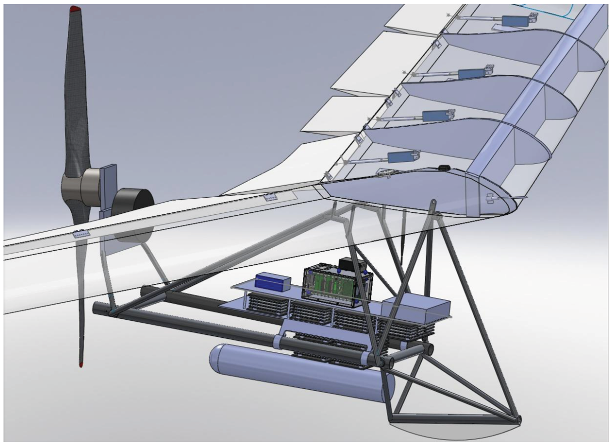


Figure 2. The Swift all-electric UAS.

3.2. Requirements and Flight Rules

The system safety requirements we want to monitor during operation of the Swift UAS can be categorized into these three types: value checks, relationships, and flight rules.

Value Checks test whether data values are plausible. Examples in this category include range checks, e.g., the maximal safe climb or descent rate. For safe operation, the values must always stay within certain ranges. Such checks can be combined with additional conditions (e.g., during the flight phase or above a minimal altitude) or temporal ranges (e.g., the maximal current drawn from the battery must not exceed 50A for more than 60 seconds to avoid overheating).

Relationships encode dependencies among sensor data that may originate from different subsystems. For example, altitude readings obtained by GPS and barometric altitude should be highly correlated. For another example, whenever the Swift UAS is in the air, its indicated airspeed reading should be greater than its stall speed; if not there is certainly a problem.

Finally, *Flight Rules* are defined by national or international institutions (e.g., part 91 of the Federal Aviation Regulations (FAR) in the USA (Federal Aviation Administration, 2013)) or by mission/system constraints that govern flights. For example, a common flight rule defines the minimum altitude an aircraft needs to climb to after takeoff: reach an altitude of 600ft within five minutes after takeoff. In a similar way, we can specify a timeout for the landing procedure of the Swift UAS: after receiving the landing command, touchdown needs to take place within three minutes. We discuss in detail how these requirements and flight rules can be specified in our framework and how they can be translated into efficient hardware.

4. System Health Management Framework

Our modeling framework for sensor and software health management separates signal processing and model-based analysis, temporal monitoring, and statistical reasoning with BNs. We first discuss the overarching design requirements before we focus on the description of the design framework. Each of the framework's three prongs will then be described in detail in the subsequent sections.

4.1. Design Requirements

For autonomous systems running on ultra-portable hardware such as the Swift UAS, the following properties are required for a deployable SHM framework.

Unobtrusiveness The SHM framework must not alter crucial properties of the Swift UAS, such as: functionality (not change its behavior), certifiability (avoid re-certification of, e.g., autopilot flight software or certified hardware), timing (not interfere with timing guarantees), and tolerances (not exhaust size, weight, power, or telemetry bandwidth constraints). The framework must be able to run and perform analysis externally to the (previously developed and tested) Swift architecture.

Responsiveness The framework must continuously and in real time monitor adherence to the safety requirements of the Swift UAS. Changes in the validity of monitored requirements must be detected within tight and a priori known time bounds. Responsive monitoring of specifications enables responsive input to the BN-based probabilistic reasoner. In turn, the BN reasoner must efficiently support decision-making to mitigate any problems encountered (e.g., for the Swift UAS an emergency landing in case the flight computer fails) to avoid damage to the UAS and its environment.

Realizability The framework must operate in a plug-and-play manner by connecting via a read-only interface to existing communication interfaces of the Swift UAS. The framework must be usable by test-engineers without assuming in-depth knowledge of hardware design and must be able to operate on-board existing UAS components without requiring significant re-configuration or additional components. The framework must be reconfigurable so that health models can be updated without a lengthy re-compilation process and can be used both during testing of the UAS and after deployment.

Considering these requirements, it seems natural to implement our SHM framework in hardware. This allows us to build a self-contained unit, operating externally to the established Swift UAS architecture, thereby complying with the UNOBTRUSIVENESS requirement. Multiple safety requirements can be monitored in parallel, with status updates delivered at every tick of the system clock, establishing the RESPONSIVENESS requirement. Previous implementations of system monitors in hardware, however, have often violated the REALIZABILITY requirement as a reconfiguration, e.g., due to changes in the SHM model, necessitates a redesign of the framework's hardware.[8] To provide greater flexibility in this regard, we design an efficient, highly parallel hardware architecture that runs on the Swift UAS' native FPGA hardware, yet keep it programmable and modular to quickly adapt to changes in our SHM models.

4.2. Design Framework

Our SHM model is constructed hierarchically in a graphical manner out of powerful building blocks. In contrast to most existing systems, we do not separate between an (informal) signal preprocessing step and the proper health management model. Rather, we elevate all processing steps to first class status and model them all in the same framework. With that approach, we can, in a principled way, deal with all temporal, probabilistic, and model-based aspects of our health management model. This uniform way of describing the health model not only enables more powerful techniques for V&V but it also directly leads to efficient implementations in programmable FPGA hardware.

All signals considered in our SHM model are streams of data, which are processed at fixed time stamps. At each tick of the system clock, a component reads the input values and calculates the output values. The order of execution is defined by a model graph. In this paper, we assume that there exists one fixed update rate for all of the building blocks of the model.[9]

Such a stream of individual elements of type T is denoted by $T^\star$; vectors are defined naturally. Table 1 shows the different data types that are used. Please note that Boolean data types are implicitly converted into a $\{0, 1\}$ discrete representation.

All data blocks have a number of inputs I_j of a given stream type, and will produce a number of outputs, again as elements of a stream. Table 2 shows a list of selected blocks. Only the current values of the signals are presented to the model. Depending on its functionality, a block can be memory-less (e.g., a Boolean function or a Bayesian reasoning block), or it can contain internal memory (of fixed length) to deal with previous signal values. Blocks with internal memory include the Linear Temporal Logic (LTL) processing blocks and blocks for data smoothing, integration, or model-based Kalman filters (Table 2).

Table 1. Data types for SHM components.

Signal Symbol	Data Type Description
R	floating point number, e.g., sensor reading
D	discrete set $\{1, \ldots, n\}$
B	Boolean
B^+	*true*, *false*, *maybe*
p	probability
P	probability density

Table 2. Typical SHM building blocks.

Name	Function	Memory	Description
BF	$B^n \to B^m$		Boolean function
LTL$_s$	$B^n \to B^{+m}$		synchronous LTL observer
LTL$_a$	$B^n \to B^m$	∘	asynchronous LTL observer
THR	$R^n \to D^m$		discretizer/threshold
FLT	$R^n \to R^n$	∘	smoothing filter
KF	$R^n \to R^{2m+n}$	∘	Kalman filter with $\hat{x}$, residual and $diag(P^-)$
BN	$D^n \to R^m$		discrete Bayes reasoner
BN$_p$	$R^l \to R^{2m}$		Bayes reasoner with evidence inputs and posterior outputs
P	$R^l \to R^2$		Prognostics unit

For example, a block to discretize sensor readings would take a floating-point number and calculate its discretized value or a Boolean value for a simple threshold. A smoothing filter, calculating a moving average, would have the functionality $R^n \to R^n$ and obviously require internal memory. A temporal monitoring component has Boolean inputs and produces a 2 or 3-valued logical output that indicates whether the monitored requirement is $true$, $false$, or $unknown$ given the inputs. In Section 5 we will discuss such monitors in detail. Our BN (see Section 7 for details) takes discrete values as inputs, called *evidence*, and produces a posterior probability. Model-based prognostics units, which take sensor signals as inputs and output estimates of remaining useful life for specific components, can substantially increase the modeling power and reasoning capabilities of our SHM framework. For example, a loss in propeller RPM might be diagnosed differently if it is known that the battery might already be fairly weak.

Beyond the building blocks shown in Table 2, additional filters, Fourier transforms, or model-based components can easily be added to improve the modeling capabilities of our SHM

[8] Or at least a run of a logic synthesis tool, which can easily take tens of minutes to complete.

[9] If signals are to be considered with different rates, rate conversion blocks or sample and hold blocks can be defined and used as needed. Note also that Bayesian networks handle missing data in a natural way and do not need conversion, sample, or hold blocks.

framework. For most of the components, efficient designs for programmable hardware are available; for our temporal monitors and Bayesian reasoning building blocks, our hardware implementations will be discussed in the sections below.

The main goal of our SHM framework is to provide a modeling paradigm that allows the modeler to separate temporal, functional, model-based, and probabilistic reasoning in a clear way while retaining the expressive power of the various formalisms. This framework also avoids powerful but complex modeling mechanisms, like dynamic Bayesian networks (DBNs).

5. Monitoring of Temporal sensor data using temporal logic

In order to encapsulate the safety requirements of the Swift UAS in a precise and unambiguous form that can be analyzed and monitored automatically, we write them in temporal logic. Specifically, we use a Linear Temporal Logic (LTL), which allows the expression of requirements over timelines and also pairs naturally with the original English expression of the requirements.[10] For requirements that express specific time bounds, we use a variant of LTL that adds these time bounds, called Metric Temporal Logic (MTL). We can automatically generate runtime monitors for requirements expressed in these logics, enabling real-time analysis of sensor data as well as system health assessment.

Linear temporal formulas consist of:

1. **Variables representing system state:** We include variables representing the data streaming from each sensor aboard the Swift UAS.
2. **Propositional logic operators:** These include the standard operators, logical AND ($\wedge$), logical OR ($\vee$), negation ($\neg$), and implication ($\rightarrow$).
3. **Temporal operators:** These operators express temporal relationships between events including Always, Eventually, NextTime, Until, and Release where the following hold for example system Boolean variables p and q.
   - Always p ($\Box p$) means that p must be true at all times along the timeline.
   - Eventually p ($\Diamond p$) means that p must be true at some time, either now or in the future.
   - NextTime p ($\mathcal{X} p$) means that p must be true in the next time step; in this paper a time step is a tick of the system clock aboard the Swift UAS.
   - p Until q ($p \mathcal{U} q$) signifies that either q is true now, at the current time, or else p is true now and p will remain true consistently until a future time when q must be true. Note that q must be true sometime; p cannot simply be true forever.
   - p Releases q ($p \mathcal{R} q$) signifies that either both p and q are true now or q is true now and remains true unless there comes a time in the future when p is also true. Note that in this case there is no requirement that p will ever become true; q could simply be true forever. The Release operator is often thought of as a "button push" operator: pushing button p triggers event $\neg q$.

For MTL, each of these temporal operators are accompanied by upper and lower time bounds that express the time period during which the operator must hold. Specifically, MTL includes the operators $\Box_{[i,j]}\ p$, $\Diamond_{[i,j]}\ p$, $p\ \mathcal{U}_{[i,j]}\ q$, and $p\ \mathcal{R}_{[i,j]}\ q$ where the temporal operator applies in the time between time i and time j, inclusive. Additionally, we use a *mission bounded* variant of LTL where these time bounds are implied to be the start and end of the mission of the UAS. In all cases, time steps refer to ticks of the system clock. So, a time bound of $[3, 8]$ would designate the time bound between 3 and 8 ticks of the system clock from now. Note that this bound is relative to "now" so that continuously monitoring a formula $\Diamond_{[3,8]}\ p$ would produce *true* at every time step t for which p holds anytime between 3 and 8 time steps after t, and *false* otherwise.

Figures 3 and 4 give pictorial representations of how these logics (mission-bounded LTL and MTL) enable the precise specification of temporal safety requirements in terms of timelines.

Examples of System Requirements in Temporal Logic. Due to their intuitive nature and a wealth of tools and algorithms for analysis of LTL and MTL formulas, these logics are frequently used for expressing avionics system requirements (Zhao & Rozier, 2012; Gan, Dubrovin, & Heljanko, 2011; Bolton & Bass, 2013; Alur & Henzinger, 1990). Recall the example system safety requirements from Section 3.2. We can straightforwardly encode each of value checks, relationship requirements, and flight rules as temporal logic formulas to enable runtime monitoring:[11]

Value Checks:

- The maximal safe climb and descent rate V_z of the Swift UAS is limited by its design and engine characteristics.

$$\Box(-200\frac{\mathrm{ft}}{\mathrm{min}} \leq V_z \leq 150\frac{\mathrm{ft}}{\mathrm{min}})$$

[10] In the temporal logic formulas of this paper, we follow the standard syntax for evaluating temporal properties where = means assignment and == means equality comparison. For example, $(a == b)$ returns *true* if a and b are equal and *false* otherwise. At the same time, we follow the tradition in probability, where = means equality and not assignment. It should be clear from the context whether we are dealing with a temporal logic expression or a probability expression.

[11] The numbers given below are for illustration purpose only and do not reflect the actual properties of the Swift UAS.

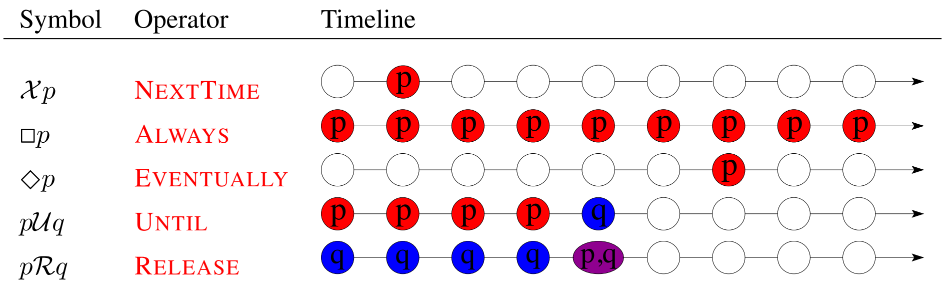


Figure 3. Pictorial representation of LTL temporal operators. For a formal definition of LTL, see for example (Rozier, 2011).

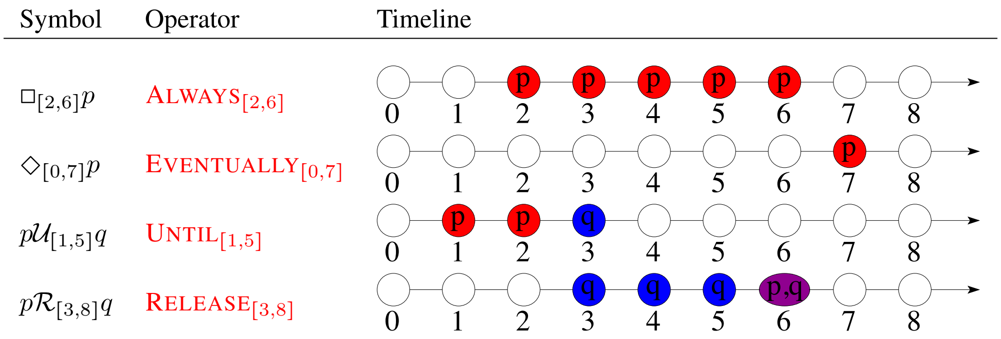


Figure 4. Pictorial representation of MTL temporal operators. For a formal definition of MTL, see for example (Alur & Henzinger, 1990).

- The maximal angle of attack α is limited by design characteristics.

$$\Box(\alpha \le 15^\circ)$$

- The Swift roll (p), pitch (q), and yaw rates (r) are for safe operation limited to remain below maximum bounds.

$$\Box(p < 0.99\frac{\text{rad}}{\text{s}} \wedge q < 4.0\frac{\text{rad}}{\text{s}} \wedge r < 2.2\frac{\text{rad}}{\text{s}})$$

- The battery voltage U_{batt} and the current I_{batt} must remain within certain bounds during the entire flight. Furthermore, no more than 50A should be drawn from the battery for more than 30 consecutive seconds in order to avoid battery overheating.

$$\begin{aligned}\Box(\quad & (20V \le U_{batt} \le 26.5V) & \wedge \\ & (I_{batt} \le 75A) & \wedge \\ & ((I_{batt} > 50A)\mathcal{U}_{[0,29s]}(I_{batt} \le 50A)))\end{aligned}$$

Relationships:

- Pitching up (i.e., increasing α) for a sustained period of time (more than 20 seconds) should result in a positive change in altitude, measured by a positive vertical speed V_z. This increase in vertical speed should occur within two seconds after pitch-up.

$$\Box(\Box_{[0,20s]}(\alpha > \alpha_0) \rightarrow \Diamond_{[0,2s]}V_z > 0)$$

This relationship can be refined to only hold if the engine has enough power (as measured by the electrical current to the engine I_{eng}) to cause the aircraft to actually climb.

$$\Box(\Box_{[0,20s]}((\alpha > \alpha_0) \wedge I_{eng} > 30A) \rightarrow \Diamond_{[0,2s]}V_z > 0)$$

Similarly, a rule for the descending can also be defined.

$$\Box(\Box_{[0,20s]}((\alpha < \alpha_0) \vee I_{eng} < 10A) \rightarrow \Diamond_{[0,2s]}V_z < 0)$$

- Whenever the Swift UAS is in the air, its indicated airspeed (V_{IAS}) must be greater than its stall speed V_S. The UAS is considered to be air-bound when its altitude alt is larger than that of the runway alt_0.

$$\Box((alt > alt_0) \rightarrow (V_{IAS} > V_S))$$

- The sensor readings for the vertical velocity V_z and the barometric altimeter alt_b are correlated, because V_z corresponds to the changes in the altitude. This means that whenever the vertical speed is positive, we should measure a certain increase of altitude Δ_{alt_b} within 2 seconds.

$$\Box(V_z > 0 \rightarrow \Diamond_{[0,2s]}\Delta_{alt_b} > \theta)$$

- The precision of the position reading P_{GPS} from the

GPS subsystem depends on the number of visible GPS satellites N_{sat}.

$$
\begin{array}{lll}
\Box(& & \\
\Box(N_{sat} == 1) & \rightarrow P_{GPS} \leq P^{1}_{GPS} & \wedge \\
\Box(N_{sat} == 2) & \rightarrow P_{GPS} \leq P^{2}_{GPS} & \wedge \\
\Box(N_{sat} == 3) & \rightarrow P_{GPS} \leq P^{3}_{GPS} & \wedge \\
\Box(N_{sat} \geq 4) & \rightarrow P_{GPS} \leq P^{+}_{GPS}) &
\end{array}
$$

Flight Rules:

- After receiving a command (cmd) for takeoff, the Swift UAS must reach an altitude of 600ft within 40 seconds.

$$\Box((\text{cmd == takeoff}) \rightarrow \Diamond_{[0,40s]}(alt \geq 600\text{ ft}))$$

- After receiving the landing command, touchdown needs to take place within 40 seconds, unless the link (lnk) is lost. Otherwise, the aircraft should reach a loitering altitude around 425ft within 20 seconds.

$$
\begin{array}{l}
\Box((\text{cmd == landing}) \rightarrow \\
((s_{lnk} == \text{ok}) \rightarrow \Diamond_{[0,40s]}(alt < 10\text{ ft}) \vee \\
(s_{lnk} == \text{lost}) \rightarrow \Diamond_{[0,20s]}(400\text{ft} \leq alt \leq 450\text{ft})))
\end{array}
$$

- The Swift default mode is to stay on the move; it should not loiter in one place for more than a minute unless it receives the loiter command (which may not ever happen during a mission). Let *sector_crossing* represent a Boolean variable, which is true if the UAS crosses the boundary between the small subdivision of the airspace in which the UAS is currently located, as determined by the GPS, and another subdivision. After receiving the loiter command, the UAS should stay in the same sector, at an altitude between 400 and 450ft until it receives a landing command. The UAS has 30 seconds to reach loitering position.

$$
\begin{array}{l}
\Box(\big[(\text{cmd == loiter})\,\mathcal{R}\,(\Diamond_{[0,60s]}\ \text{sector\_crossing})\big] \wedge \\
\quad \big[(\text{cmd == loiter}) \rightarrow \\
\qquad (\Box_{[30s,end]}((\neg\text{sector\_crossing}) \wedge \\
\qquad\qquad (400\text{ft} \leq alt \leq 450\text{ft})) \\
\qquad\ \mathcal{U}\,(\text{cmd == landing})) \\
\quad \big])
\end{array}
$$

- All messages sent from the guidance, navigation and control (GN&C) component to the Swift actuators must be logged into the on-board file system (FS). Logging has to occur before the message is removed from the queue. In contrast to the requirements stated above, this flight rule specifically concerns properties of the flight software.

$$
\begin{array}{l}
\Box((\text{addToQueue}_{\text{GN\&C}} \wedge \Diamond\text{removeFromQueue}_{\text{Swift}}) \quad \rightarrow \\
\quad \neg\text{removeFromQueue}_{\text{Swift}}\ \mathcal{U}\ \text{writeToFS})
\end{array}
$$

Advantages of Temporal Logic Requirements. Encoding the set of system requirements in temporal logic offers several significant advantages. It yields a very precise, unambiguous list of the system requirements that aids in project planning and organization. It enables us to automatically synthesize runtime monitors to track these requirements onboard the Swift UAS directly from the temporal logic specifications. It also enables other types of automatic checks, such as automatic requirements debugging (i.e., satisfiability checking (Rozier & Vardi, 2010)) and design-time V&V techniques such as model checking (Rozier, 2011).

5.1. Monitoring Approach

From each temporal logic requirement, we automatically generate two kinds of monitors, which we call *synchronous* and *asynchronous* monitors, working in coordination to provide real-time system health updates. A *synchronous monitor* provides an update on the requirement with every update of the system clock. This is important because it provides blocks such as the Bayesian reasoner with better real-time information and therefore improves prognostics capabilities by enabling monitoring input to be considered by the reasoner. Our synchronous runtime monitors keep up-to-date information on how much time is left until a requirement must pass. An *asynchronous monitor* provides an update on the final outcome of the requirement at an a priori known time. Our asynchronous monitors report if a requirement is satisfied or fails earlier than expected or yield the final result (pass or fail) of the requirement when its time bound has elapsed. For details on the construction of these monitors, and formal proofs that our constructions are correct, see (Reinbacher, Rozier, & Schumann, 2013).

This dual-monitor construction is a key element of our SHM framework, because it enables our runtime monitors to be used as building blocks in combination with the other blocks described in this paper. Traditional runtime monitoring techniques only operate asynchronously and only report when a monitored property *fails*. Our monitors provide much more useful output. For example, it can be important in computing prognostics to know that a requirement that must happen within a specified time bound has not yet been satisfied and that the time bound is almost up. This allows mitigating actions to be considered in time. For another example, if a requirement states that ($\text{E}{\small\text{VENTUALLY}}_{[3,2005]}\ p$) and p occurs at time 5 it is important to utilize this information for real-time calculations of system health. Traditional monitoring techniques do not yield any output in this case, either at time 5 or 2005 since no property failure occurred. Finally, it is key that our runtime monitors can provide this information without any modifications to certified flight software or hardware, operating in isolation aboard an FPGA with a read-only interface, whereas most runtime monitoring techniques utilize more obtrusive techniques for gathering system data.

6. Model-based monitoring of Temporal sensor data

Highly accurate and detailed information about system health could be obtained if the actual system is compared with a high-fidelity simulation model. Model complexity and resource limitations make such an approach infeasible in most cases. However, a comparison of system behavior with an abstracted dynamical model is an attractive option. HyDE, for example, performs health management using simplified and abstracted system models.

For our framework, we provide the capability to use model-based monitoring components to various degrees of abstraction. The most common of such components is a Kalman filter. Here, a linearized model of the (sub-)system dynamics is used to predict the system state from past sensor readings. Besides this state prediction, the residual of the Kalman filter is of importance for our purposes, as it reflects how well the model represents the actual behavior (Brown & Hwang, 1997). A sudden increase of the filter residual, for example, can give an indication of a malfunctioning sensor. For implementation, we use our tool AutoFilter (Whittle & Schumann, 2004) to automatically generate customized Kalman filter algorithms from high-level requirements. As we refine our configuration to handle more complex SHM capabilities needed for future flight tests of the Swift UAS, we are planning to extend the AutoFilter tool in order to directly produce corresponding FPGA designs (see, e.g., Pasricha & Sharma, 2009). In a similar manner, non-linear models could be handled using Particle Filters (Ristic, Arulampalam, & Gordon, 2004), though these require more computational efforts.

A very simple temporal monitoring technique is the use of FFT in order to obtain an estimate of the frequency spectrum of the monitored signals. This information is, for example, important to detect oscillations of the aircraft (see Section 8.3), or to detect unexpected software behavior, like a reboot loop.

Though our implementation at this time is limited to standard filtering monitors, we envision creating more powerful model-based monitors using prognostics models to produce statistical distributions for the end-of-life of system components based upon sensor readings. For example, a prognostics model to estimate the remaining useful life of the laser altimeter could be used to effectively encode a dynamical MTBF into our health management system. Again, both the mean remaining life as well as information about its probability distribution can be directly used for reasoning. Although such model-based health management components can be very powerful, a number of issues still need to be addressed, including model validity, implementation in efficient hardware, and possible model adaptation to better detect and handle certain kinds of failures.

7. Bayesian Health Management Reasoning

The major reasoning component in our SHM framework is a Bayesian network (BN) used to perform diagnostic reasoning and root causes analysis. A BN is a multivariate probability distribution that enables reasoning and learning under uncertainty (Pearl, 1988; Darwiche, 2009). In a BN, random variables are represented as nodes in a Directed Acyclic Graph (DAG), while conditional dependencies and independencies between variables are induced by graph edges (see Figure 5 for an example). A BN's graphical structure often represents a domain's causal structure, and is typically a compact representation of a joint probability table. Each node in the BN's graphical structure is associated with a corresponding conditional probability table (CPT) that captures its (causal) relationship with parent nodes.

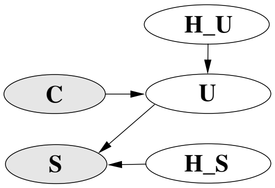


Figure 5. Simple Bayesian network.

In our framework, the BN inputs are comprised of discrete or discretized values (e.g., *low*, *high*), and reasoning is performed at each tick of the system clock. We are using discrete and static BNs, which do not perform any reasoning in the temporal domain. All temporal reasoning, as well as other processing, has been cleanly separated out within our modeling framework. Although, in general, multiple different probabilistic queries can be formulated, our framework aims to compute marginal posterior probabilities of selected nodes, which then give an indication (probability) of component or system health. Thus our Bayesian reasoning components compute a posteriori probabilities as their output. Different BN inference algorithms can be used to compute a posteriori probabilities. These algorithms include junction tree propagation (Lauritzen & Spiegelhalter, 1988; Jensen et al., 1990; Shenoy, 1989), conditioning (Darwiche, 2001), variable elimination (Li & D'Ambrosio, 1994; Zhang & Poole, 1996), stochastic local search (Park & Darwiche, 2004; Mengshoel, Roth, & Wilkins, 2011; Mengshoel, Wilkins, & Roth, 2011), and arithmetic circuit evaluation (Darwiche, 2003; Chavira & Darwiche, 2007).

7.1. Bayesian Health Models

For the Bayesian models, we follow an approach that "glues together" Bayesian network fragments (Schumann, Mengshoel, & Mbaya, 2011; Schumann et al., 2013). For example, consider the Bayesian network in Figure 5. It consists of four different types of interconnected nodes, namely: command node C, health node H, sensor node S, and status node U. The health node H has subtypes H_S for sensor nodes and H_U for status nodes. Inputs to a BN is provided by connecting an input signal to the state of a node ("clamping"). Command nodes C are handled as ground truth and used to indicate commands, actions, or other known states. Command nodes do not have incoming edges. Sensor nodes S are also input nodes, but the input signal is not necessarily correct (e.g., it could result from a failed sensor). This behavior is modeled by connecting S to a health node H that reflects the health of the input to S. Note that inputs to the BN can be outputs of any block in our framework, for example, a smoothed and discretized sensor reading, the result (binary or ternary) of a temporal monitor, or the output of another reasoning block.

Status nodes U, and similar behavior nodes B, are internal nodes and reflect the (latent) status of the subsystem or component, or recognize a specific behavior, such as pilot-induced oscillation. Typically, health nodes H are attached to status nodes, but not to behavior nodes. Associated with each node is a Conditional Probability Table (CPT), which defines the conditional probability of node X, given the states of the parent nodes of X.

For modeling the edges of the BN, we follow the rule that an edge from node X to node Y indicates that the state of X has a (causal) influence on the state of Y. Table 3 gives an overview of the different kinds of edges in our modeling framework.

Table 3. Types of edges typically used in BN models for the SHM reasoning blocks.

edge E	E represents how ...
$\{H, C\} \overset{E}{\rightarrow} U$	status U, with health H, is controlled through unreliable command C
$\{C\} \overset{E}{\rightarrow} U$	status U is controlled through unreliable command C
$\{H, U\} \overset{E}{\rightarrow} S$	status U influences sensor S, which may fail as reflected in health H
$\{H\} \overset{E}{\rightarrow} S$	health H directly influences sensor S without modeling of status
$\{U\} \overset{E}{\rightarrow} S$	status U influences sensor S

7.2. Compilation to Arithmetic Circuits

We select arithmetic circuit evaluation as the inference algorithm used in our framework, and therefore compile our Bayesian network into an arithmetic circuit. In real-time avionics systems, where there is a strong need to align the resource consumption of diagnostic computation with resource bounds (Musliner et al., 1995; Mengshoel, 2007), algorithms based upon arithmetic circuit evaluation are powerful, as they provide predictable real-time performance (Chavira & Darwiche, 2005; Mengshoel et al., 2010).

An arithmetic circuit (AC) is a DAG in which leaf nodes λ represent parameters and indicators while other nodes represent addition and multiplication operators.

Posterior marginals in a Bayesian network can be computed from the joint distribution over all variables $X_i \in \mathcal{X}$:

$$p(X_1, X_2, \ldots) = \prod_{\lambda_x} \lambda_x \prod_{\theta_{x|\mathbf{u}}} \theta_{x|\mathbf{u}}$$

where $\theta_{x|\mathbf{u}}$ are the parameters of the Bayesian network, i.e., the conditional probabilities that a variable X is in state x given that its parents $\mathbf{U}$ are in the joint state $\mathbf{u}$, i.e., $p(X = x \mid \mathbf{U} = \mathbf{u})$. Further, λ_x indicate whether or not state x is consistent with the observations. For efficient calculation, we rewrite the joint distribution into the corresponding network polynomial f (Darwiche, 2003):

$$f = \sum_{\mathbf{x}} \prod_{\lambda_x} \lambda_x \prod_{\theta_{x|\mathbf{u}}} \theta_{x|\mathbf{u}}$$

An arithmetic circuit is a compact representation of a network polynomial (Darwiche, 2009) which, in its uncompact form, is exponential in size and thus unrealistic in the general case. Hence, answers to probabilistic queries, including marginals and MPEs, are computed using algorithms that operate directly on the arithmetic circuit. The marginal probability (see Corollary 1 in (Darwiche, 2003)) for x given evidence **e** is calculated as

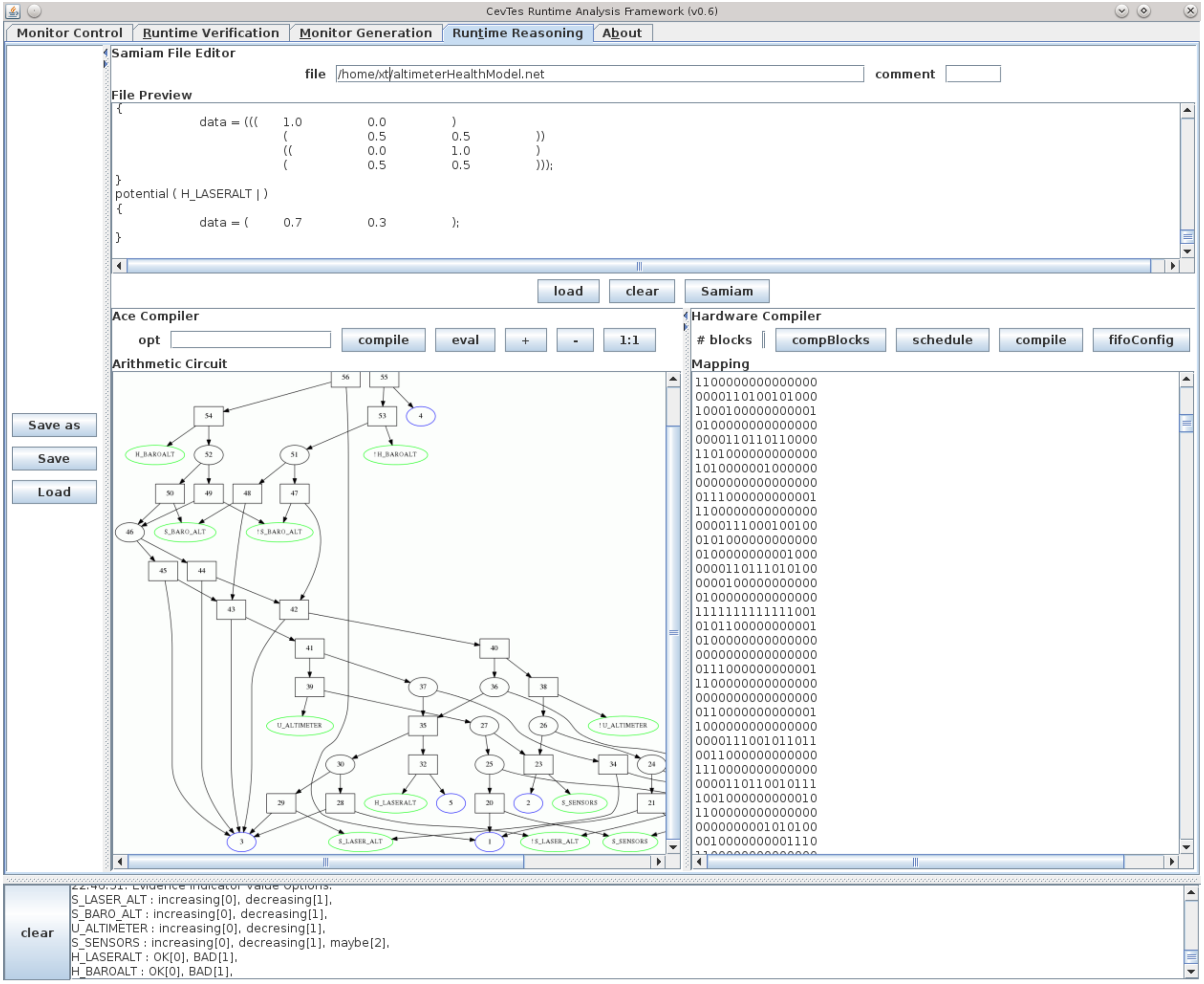
$$\Pr(x \,|\, \mathbf{e}) = \frac{1}{\Pr(\mathbf{e})} \cdot \frac{\partial f}{\partial \lambda_x}(\mathbf{e}) \qquad (1)$$

where $\Pr(\mathbf{e})$ is the probability of the evidence. In a bottom-up pass over the circuit, the probability of a particular evidence setting (or clamping of λ parameters) is evaluated. A subsequent top-down pass over the circuit computes the partial derivatives $\frac{\partial f}{\partial \lambda_x}$. This mechanism can also be used to provide information about how change in a specific node affects the whole network (sensitivity analysis), and to perform MPE computation (Darwiche, 2003, 2009).

7.3. Efficient Hardware Realization

Bayesian reasoning blocks in our framework are provided with values produced by other blocks, as input to C and S nodes. In our BN hardware implementation, these evidence values are used to calculate posterior marginals for the health nodes H of the Bayesian SHM model. For efficient hardware realization of this kind of BN reasoning, we note that posterior marginals are evaluated in the arithmetic circuit by traversing the nodes of the circuit in a bottom-up and a subsequent top-down manner.

We make the following observations regarding the structure



file /home/xt/altimeterHealthModel.net

opt

```
{
            data = (((    1.0        0.0        )
                    (                0.5        0.5        ))
                    ((               0.0        1.0        )
                    (                0.5        0.5        )));
}
potential ( H_LASERALT | )
{
            data = (      0.7        0.3        );
}
```

```
1100000000000000
0000110100101000
1000100000000001
0100000000000000
0000110110110000
1101000000000000
1010000001000000
0000000000000000
0111000000000001
1100000000000000
0000111000100100
0101000000000000
0100000000001000
0000110111010100
0000100000000000
0100000000000000
1111111111111001
0101100000000001
0100000000000000
0000000000000000
0111000000000001
1100000000000000
0000000000000000
0110000000000001
1000000000000000
0000111001011011
0011000000000000
1110000000000000
0000110110010111
1001000000000010
1100000000000000
0000000001010100
0010000000001110
```

```
22:40:51: Evidence indicator value options:
S_LASER_ALT : increasing[0], decreasing[1],
S_BARO_ALT : increasing[0], decreasing[1],
U_ALTIMETER : increasing[0], decresing[1],
S_SENSORS : increasing[0], decreasing[1], maybe[2],
H_LASERALT : OK[0], BAD[1],
H_BAROALT : OK[0], BAD[1],
```

Figure 6. Screenshot of our GUI-based BN synthesis tool. There is a textual description of the altimeter health model Bayesian network (top), a compiled arithmetic circuit of the network (bottom left), and a binary configuration for our μBayes unit (bottom right).

of arithmetic circuits:

(i) The structure alternates between addition and multiplication nodes. Nodes labeled with "+" are addition nodes; those labeled with "×" are multiplication nodes.

(ii) Each multiplication node has a single parent.

(iii) Input nodes (i.e., leaf nodes) are always children of multiplication nodes.

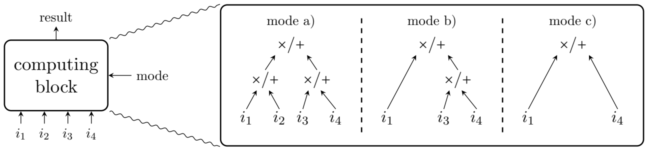


Figure 7. A computing block and its three modes of operation.

Hardware Architecture of μBayes. The above observations, concerning the structure of arithmetic circuits, led us to a hardware architecture that evolves around parallel units called computing blocks. A computing block, as shown in Figure 7, is designed to match the structural properties (i-iii) of an arithmetic circuit. A single computing block supports three basic modes to process the different shapes found in subtrees of arithmetic circuits. Re-arrangement of the arithmetic circuit using commutativity properties of the operators enables us to tile the entire AC with instances of these three modes in Figure 7.

These computing blocks are the building blocks of our

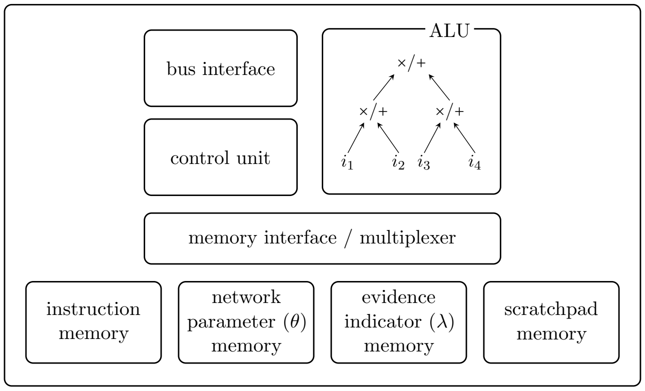

bus interface
control unit
ALU
×/+
×/+
×/+
i_1
i_2
i_3
i_4
memory interface / multiplexer
instruction memory
network parameter (θ) memory
evidence indicator (λ) memory
scratchpad memory

Figure 8. Internals of a computing block.

Bayesian SHM hardware unit, which we call μBayes.

Figure 8 shows the internals of a computing block. The unit is loaded with network parameters from the CPT of the health model at configuration time. At each SHM update cycle, inputs are provided as evidence indicators and stored in a separate evidence indicator memory. An offline compiler translates the structure of the arithmetic circuit into native instructions for the μBayes unit. Instructions encode the operation (either addition or multiplication) of each individual node of the Arithmetic Logic Unit (ALU), control the multiplexer to load/store operands from/to memory, trigger transfers of results, and coordinate loads of inputs. Instructions are decoded and forwarded by the control unit. Each computing block manages a scratchpad memory to save intermediate local results, computed during the bottom-up traversal, which can be reused during the top-down traversal. The memory blocks of the μBayes unit are mapped to block RAMs of the FPGA.

Figure 9 shows the architecture of our Bayesian health management hardware unit. It interconnects and controls multiple computing blocks to process arithmetic circuits in parallel.

The master unit manages bus accesses, stores intermediate global results, and computes posterior marginals according to Equation 1. The inverse of the probability of the evidence, $\frac{1}{\Pr(\mathbf{e})}$, in this equation can be computed within the master in parallel to the top-down traversal of the arithmetic circuit once the bottom-up traversal is completed. Posterior marginals can then be computed efficiently by multiplying the partial derivatives $\frac{\partial f}{\partial \lambda_x}$ obtained by the top-down traversal with the cached value of $\frac{1}{\Pr(\mathbf{e})}$.

For our implementation, we designed the μBayes unit in the hardware description language VHDL and use the logic-synthesis tool ALTERA QUARTUS II[12] to synthesize the design onto an Altera Cyclone IV EP4CE115 FPGA. In our implementation, we chose to represent fractional values in a fixed-point representation. This decision avoids the considerable blow-up in hardware requirements that we would incur if all of the computing blocks had to be equipped with a full-fledged floating-point unit. Instead, we instantiate fixed-point multipliers, available on our target FPGA, to realize the arithmetic operations within the computing blocks. Modern-day FPGAs provide several hundred of such multiplier units.

Synthesizing an Arithmetic Circuit into a μBayes Program. A (GUI-based) application (see Figure 6) on a host computer compiles an arithmetic circuit into a tuple $\langle \Pi, C \rangle$, where Π is a native program for the μBayes unit and C is a configuration for the network parameter memory. The synthesis of $\langle \Pi, C \rangle$ from an arithmetic circuit involves the following steps:

(1) Parse the circuit into a DAG and use compile-time information from the Ace package[13] to relate nodes in the DAG to evidence indicators and network parameters. Assemble network parameter values according to the CPTs and add them to C. Perform equivalence transformations on the DAG to ensure that the available modes of a computing block are able to cover all parts of the arithmetic circuit.

(2) Apply a variant of the Bellman-Ford algorithm (Bellman, 1958) to the DAG to determine the distance of each node to the root node. Based on the distances and the width of the arithmetic circuit, determine the number of required computing blocks. Rearrange computing blocks to optimize the number of results that can be reloaded from the same computing block in the next computation cycle.

(3) For each computing block c: generate an instruction π for each node in the arithmetic circuit that is computed by c. Finally, add π to Π.

To configure the μBayes unit, the tuple $\langle \Pi, C \rangle$ is transferred at configuration time, i.e., before deployment, to the master unit, which then programs the individual computing blocks. During operation, the entries for the evidence indicator memory are broadcast by the master unit at each tick of the system clock when new input values are available.

Hardware Resource Consumption. We synthesized the hardware design of the μBayes unit for various target FPGAs using the industrial logic synthesis tool ALTERA QUARTUS II. To study the hardware resource consumption of our design we synthesized the design several times with varying numbers of computing blocks. For our implementation, we used a fixed-point number representation with 18 bits to internally represent probabilities. We have chosen this representation

[12] Available at http://www.altera.com. We used v11.1 in our experiments.

[13] http://reasoning.cs.ucla.edu/ace/

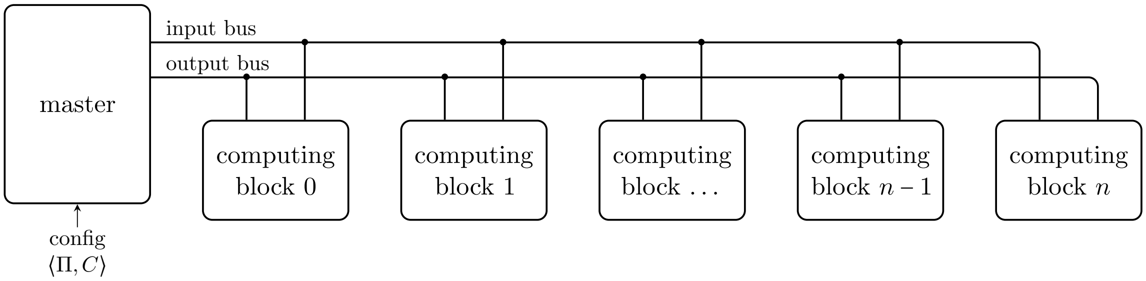


Figure 9. Architecture of the μBayes unit with parallel computing blocks.

mainly because our target FPGA provides fixed point multipliers that support vectors of up to 18 bits.

For example, an instantiation of the μBayes unit with 7 parallel computing blocks accounts for a total of 25,719 logic elements (22.5 % of the total logic elements) and 20,160 memory bits (2.5 kByte, 0.5 % of the total memory bits) and allows for a maximum operating frequency f_{max} of 115 MHz (for the slow timing model at 85 °C) on an Altera Cyclone IV EP4CE115 FPGA. We note that the operating frequency can easily be increased by moving to a more powerful FPGA. Figure 10 shows the influence of the number of computing blocks on maximum operating frequency, number of logic elements, and the number of memory bits.

8. Experiments and Results

In this section, we present results of experiments. In order to illustrate our three-pronged approach, we first discuss monitoring of requirements using examples of temporal logic monitors as presented in Section 5. In all of the examples, actual sensor and signal values are prefixed by "s_", e.g., s_baroAlt comprises a stream of sensor readings of the barometric altitude. We next discuss an example of how to determinate the health of sensors using BNs and show results using actual flight data, where the laser altimeter failed. The final part of this section is devoted to an example of how our framework can be used for reasoning about software.

8.1. Monitoring of Requirements

Recall from Section 3.2 and Figure 1 that our SHM framework operates on a set of requirements, which are interpreted via paths through a network of building blocks to achieve our diagnostics and prognostics goals. We create model-based monitors (Section 6) and Bayesian reasoning components (Section 7) to support monitoring these requirements. We create synchronous and asynchronous runtime monitors in hardware, aboard FPGAs, from our temporal logic translations of the requirements (Section 5). In this way, requirements form the backbone of our SHM framework.

Here, we exemplify the monitoring process for our temporal logic-based runtime monitors, including how they take input from and pass input to other blocks in our SHM framework. We demonstrate the power of generating monitors from temporal logic requirements.

For example, consider the requirement $\square((\text{s\_cmd} == \text{takeoff}) \rightarrow \Diamond_{[0,40s]}(\text{s\_baroAlt} \geq 600 \text{ ft}))$ from Section 3.2 that states, "*After takeoff, the Swift UAS must reach an altitude of* 600*ft within 40 seconds*." Recall that we encoded this requirement in MTL in Section 5 and discussed creating a pair of runtime monitors that yield both a synchronous monitor that updates with each tick of the system clock and an asynchronous monitor that determines the satisfaction of the requirement as soon as there is enough information to do so.

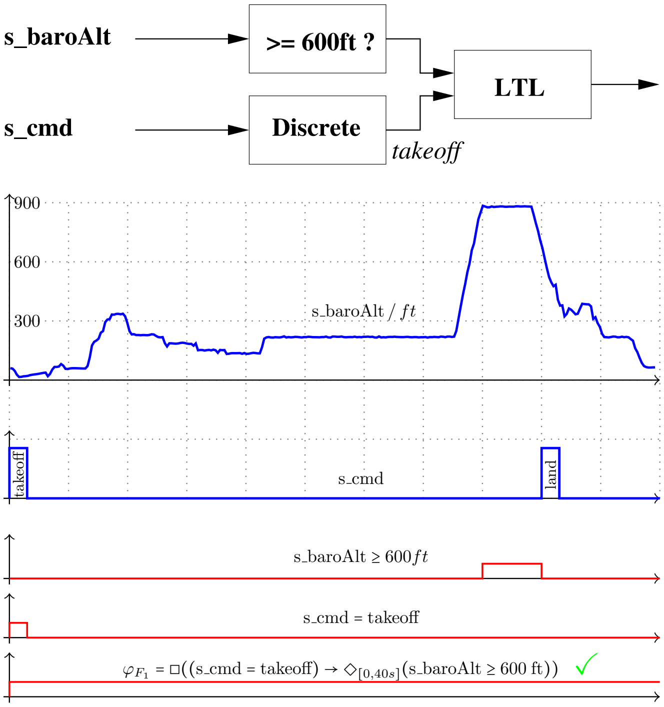


Figure 11. Top panel: SHM block diagram for monitoring requirement $\square((\text{s\_cmd} == \text{takeoff}) \rightarrow \Diamond_{[0,40s]}(\text{s\_baroAlt} \geq 600 \text{ ft}))$. Middle two panels: flight data collected from the Swift UAS. Bottom three panels: output of our runtime monitors for flight rules.

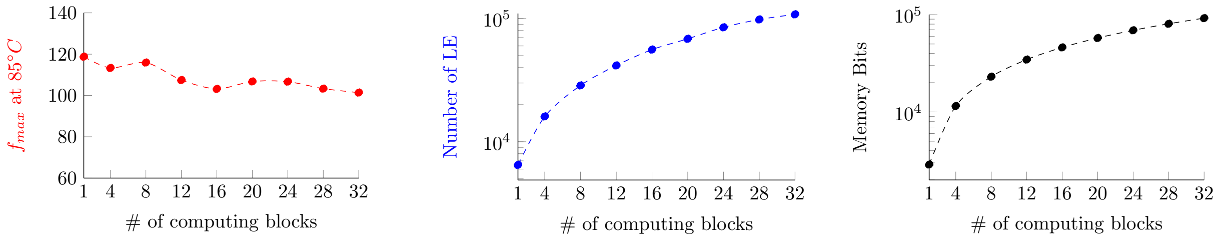


Figure 10. Logic synthesis results of our μBayes unit for an Altera Cyclone IV EP4CE115 FPGA: maximum operating frequency f_{max}, number of logic elements (LE), and required memory bits versus number of parallel computing blocks

Figure 11 breaks down how we monitor this requirement. First, the raw data from the barometric altimeter is passed through one of our smoothing filter blocks, as described in Section 4, to take out sensor noise that might serve to obscure the altimeter readings. The data stream from this smoothing filter and the raw data from the flight command data stream are the two inputs to our pair of temporal logic monitors for this requirement. These two inputs are shown in blue in Figure 11. In the bottom three panels, in red, are the output data streams from the asynchronous monitor. The top line is the result of monitoring the subformula (s_baroAlt ≥ 600 ft) and the middle line is the result of monitoring the subformula (s_cmd == takeoff). These two signals are combined inside our compositional monitor construction to form the result illustrated in the bottom panel. The panel's straight red line shows that the requirement holds at every time point during the flight. This bottom line is the output from our asynchronous monitor and can be used as the input to another block in our SHM framework, such as a Bayesian reasoning block.

During UAS flight, this output data stream will not be provided in real time, but will experience delays as there is not enough data at every time point of flight to determine that this requirement always holds. Therefore, any blocks in our SHM framework making real-time decisions could utilize the output from the paired synchronous monitor for this formula. It would differentiate, in real time, when we know that the flight rule holds and when we do not have enough information, at the present time, to know.

For another example, consider the requirement

$$\Box(\Box_{[0,5s]}(\text{v\_vel} > 0) \rightarrow \Diamond_{[0,2s]}\Delta_{\text{s\_baroAlt}} > \theta)$$

stating that a significant positive vertical velocity needs to be followed by a climb in altitude. Figure 12 breaks down how we monitor this requirement.

Again, we take the raw data from the barometric altimeter, pass it through one of our smoothing filter blocks to reduce the sensor noise, and feed this stream as an input to our temporal monitor. (Again, the smoothed barometric altimeter data stream appears in blue.) We also need to reason about the vertical velocity reading; we show the raw data stream in red. We feed this sensor data stream through a moving average filter; the result is shown in blue.

These two filtered data streams are then processed by components of our asynchronous runtime monitor; results are shown in the bottom three panels of Figure 12. The red line at the top, our vertical velocity monitor, checks for a "significant positive vertical velocity." System designers equate this to a steady positive reading of the filtered vertical velocity reading for five seconds. The red line in the middle, our barometric altimeter monitor, flags time points that fall within a two second time interval when the change in altitude is above the threshold θ. These components comprise our runtime monitor, which continuously verifies that "every occurrence of significant positive vertical velocity is indeed followed by a corresponding positive change in altitude." This is reflected by the straight red line in the bottom-most panel of Figure 12.

8.2. Sensor Health Management

The continuous monitoring of the UAS's flight-critical sensors is very important. Faulty, iced, or clogged pitot tubes for measuring speed of the aircraft has caused several catastrophes. For example, the crash of Birgenair Flight 301, which claimed 189 lives, was caused by a pitot tube being blocked by wasp nests[14]. Similarly, faults in the determination of the aircraft's altitude can lead to dangerous situations. In many cases, however, the health of a sensor cannot be established independently. Only by taking into account information from other sources can a reliable result be obtained. However, these other sources of information are also not independently reliable, thus creating a non-trivial SHM problem.

In the following example, we use information from a barometric altimeter measuring altitude above sea level, a laser altimeter measuring altitude above ground level (AGL), and information about the vertical velocity and the pitch angle provided by the Inertial Measurement Unit (IMU). Table 4 lists the signals and their intended meanings. Our corresponding SHM framework instantiation is shown in Figure 13. The input signals are smoothed and the current vertical velocity is estimated from the laser and barometric altimeters by calculating $x_t - x_{t-1}$ using a single delay block. Then, the val-

[14] http://en.wikipedia.org/wiki/Birgenair_Flight_301

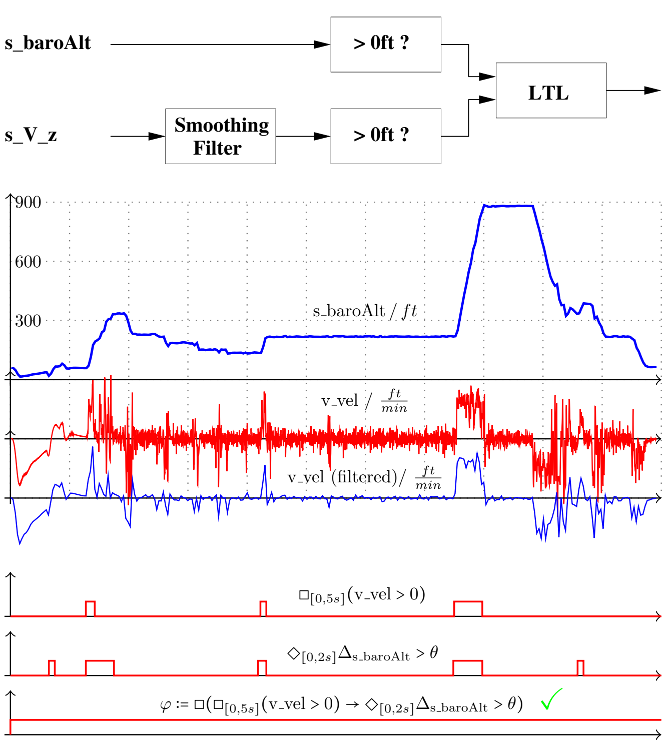


Figure 12. Top panel: SHM block diagram for monitoring a requirement. Middle three panels: Smoothed barometric altimeter (blue) and vertical velocity readings, raw (red) and smoothed (blue), as collected from the Swift UAS. Bottom three panels: outputs of temporal logic monitors.

ues are discretized into increasing (*inc*) and decreasing (*dec*), before the information is fed into the reasoning component.

Table 4. Signals and their intended meanings.

Signal name	Description
s_baroAlt	altitude reading from barometric altimeter
s_laserAlt	altitude reading from laser altimeter
s_velUp	vertical velocity reading from IMU
s_pitch	Euler pitch reading from IMU

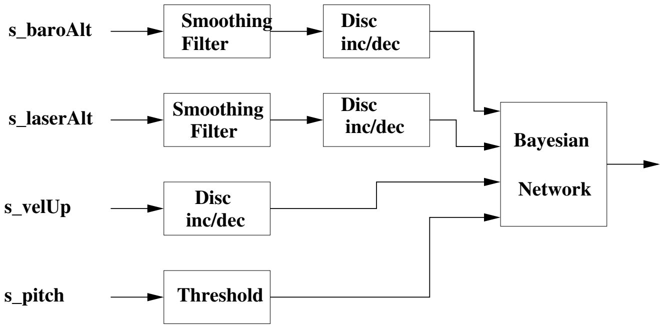
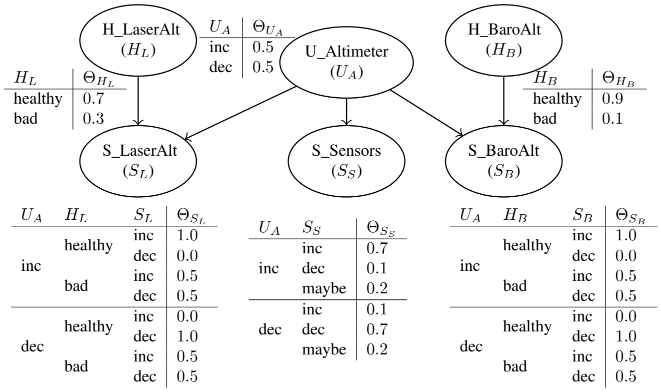


Figure 13. SHM framework instantiation: model for altimeter health.



H_L	Θ_{H_L}
healthy	0.7
bad	0.3

U_A	Θ_{U_A}
inc	0.5
dec	0.5

H_B	Θ_{H_B}
healthy	0.9
bad	0.1

U_A	H_L	S_L	Θ_{S_L}
inc	healthy	inc	1.0
		dec	0.0
	bad	inc	0.5
		dec	0.5
dec	healthy	inc	0.0
		dec	1.0
	bad	inc	0.5
		dec	0.5

U_A	S_S	Θ_{S_S}
inc	inc	0.7
	dec	0.1
	maybe	0.2
dec	inc	0.1
	dec	0.7
	maybe	0.2

U_A	H_B	S_B	Θ_{S_B}
inc	healthy	inc	1.0
		dec	0.0
	bad	inc	0.5
		dec	0.5
dec	healthy	inc	0.0
		dec	1.0
	bad	inc	0.5
		dec	0.5

Figure 14. Bayesian network and CPT tables for reasoning about altimeter failure.

Figure 14 shows the BN model for reasoning about altimeter failures. Sensor nodes (inputs) for each of the different sensor types are at the bottom. The latent state U_A, describing whether the altitude of the UAS is increasing or decreasing, obviously influences the sensor readings, hence there are edges from U_A to S_L, S_S, and S_B. The laser altimeter can fail. Therefore, the sensor node S_L is connected with a node H_L, reflecting the health of the laser altimeter. A similar structure can be found for the barometric altimeter. Because the laser altimeter is prone to errors, its probability of being healthy is only 0.7, compared to the more reliable barometric altitude with a probability of being healthy of 0.9. For simplicity, the health of the IMU is not modeled here.

The CPT tables for the sensors are read as follows: if the (latent) status U_A is increasing and the laser altimeter is healthy, then the probability that it is reading an increasing value is 1; no decreasing measurement is reported ($p = 0$). In the case of a failing laser altimeter, no result can be obtained; hence $p = 0.5$. The same model is used for the barometric altitude. The IMU sensors are modeled somewhat differently. If they report an upward velocity, it is likely ($p = 0.7$) that this has been caused by an upward movement of the UAS ($U_A = inc$). Due to high sensor and integration noise, the results are not unique and $U_A = maybe$ indicates a value within a zero-centered deadband. Figure 15 breaks down how we evaluate this Bayesian network and how our architecture automatically detected a temporary outage of the laser altimeter.

With our current implementation of the μBayes unit and a configuration as shown in Figure 13, running at a system clock frequency of 115 MHz, the unit is able to evaluate the Altimeter Health Model 660 times per second.

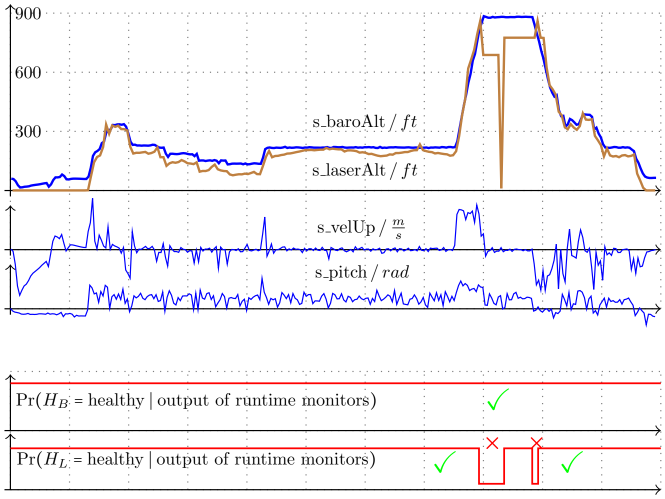


Figure 15. Flight data collected from the Swift UAS (top three panels) and output of our Bayesian SHM model, given as probabilities (bottom two panels).

8.3. Reasoning about Software

In principle, SHM models for software components are structured in a similar way to those for sensor monitoring. Signals are extracted from a communications bus between components, from specific memory locations using shared variables, or from the operating system. No specific instrumentation of the safety-critical control code is necessary. Compared to hardware and sensor management, the complexity of software health models is usually higher, because of the often substantial functionality of the code including the existence of modes. Furthermore, substantial reasoning can be required, because individual failures (due to dormant software bugs or problematic hardware-software interaction) might pervade large portions of the software system and can cause seemingly unrelated failures in other components. Such a situation occurred when a group of six F-22 Raptors was first deployed to the Kadena Air Base in Okinawa, Japan (Johnson, 2007). When crossing the international dateline (180° longitude), a dormant software bug caused multiple computer crashes. Not only was the navigation completely lost, but also the seemingly unrelated communications computer crashed. "The fighters were able to return to Hawaii by following their tankers in good weather. The error was fixed within 48 hours and the F-22s continued their journey to Kadena" (Johnson, 2007).

We now consider how such an unfortunate interplay between software design and poor implementation could cause adverse effects on the flight hardware. Figure 16 shows a mock-up of a flawed architecture for a flight-control computer. In this architecture, all components, like GN&C, the drivers for the aircraft sensors and actuators, as well as payload components including a science camera and the transmitter for the video stream, communicate via a message queue. The message queue is fast enough to push through all messages at the required speed. However, for debugging and logging purposes, all message headers are written (in blocking mode) into an on-board file system. A corresponding requirement appears as an example flight rule in Section 5:

$$\square((\text{addToQueue}_{\text{GN\&C}} \wedge \Diamond\text{removeFromQueue}_{\text{Swift}}) \rightarrow \neg\text{removeFromQueue}_{\text{Swift}}\,\mathcal{U}\,\text{writeToFS}).$$

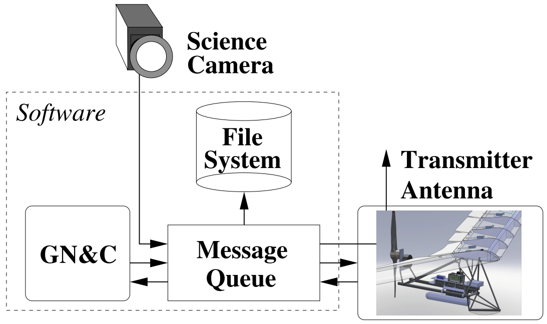


Figure 16. Flawed system architecture for file system-related scenario.

This architecture works perfectly when the system is started and the file system is empty or near empty. However, after some time of operation, as the file system becomes increasingly populated (but writes can still occur), sudden aircraft oscillations, similar to pilot-induced-oscillations (PIO), take place. No software error whatsoever (e.g., overfull file system or overfull message queue) is reported and the situation worsens if the science camera, which also uses this message queue, is in operation.

The underlying root cause is that writes into the file system take an increasing amount of time as the file system fills up (due to long searches for free blocks). This situation accounts for longer delays in the message queue, which cause delays in the seemingly unrelated control loop, ultimately causing oscillations of the entire UAS. For a software health model, therefore, non-trivial reasoning is important, because such failures can manifest themselves in seemingly unrelated components of the aircraft.

Table 5 and Figure 17 show details of our model. All signals except the barometric altitude are extracted from the operating system running on the flight computer. In the diagram in Figure 17, discrete signals are fed directly into the Bayesian networks; continuous signals like the length of the message queue or the amount of data in the file system are categorized to enable discretization into threshold blocks, e.g., the file system is empty, filled to more than 50%, filled to more than 90%, or full. The barometric altitude is fed through a Fast Fourier Transform (FFT) in order to obtain the frequency

Table 5. Signals and their intended meanings.

Signal name	Description
s_FS_Error	error in file system
s_W_FS	writing into file system
s_FS	“df” of file system
s_Queue_lng	length of message queue
s_baroAlt	barometric altitude
s_delta_q	dynamic queue behavior (derived)
s_osc	UAS oscillation (derived)

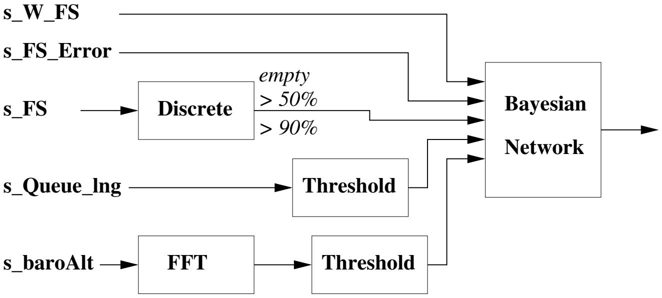


Figure 17. Structure of the SHM model for the file system scenario.

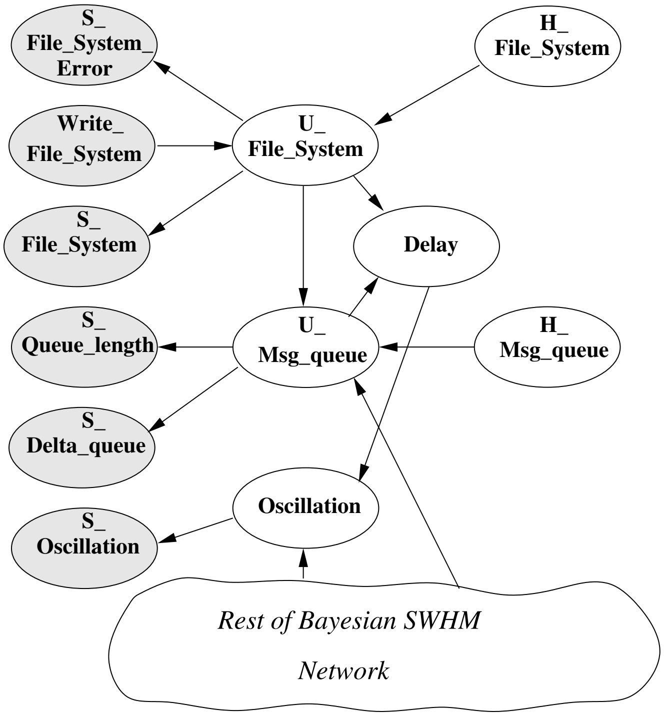


Figure 18. Relevant nodes from Bayesian system health model for oscillation detection.

spectrum. Again, a threshold block is used to determine if amplitudes are above a certain threshold indicating oscillation (low frequency) or strong vibrations (higher frequency).

Figure 18 shows the relevant excerpt from our Bayesian SHM model for this scenario, including the file system and the message queue. The software-related sensor nodes for this model are located on the left-hand side of the network: a sensor to detect writes to the file system and a sensor providing information on storage capacity in the file system (with states: *empty*, *medium*, *almost- full*, and *full*). Similarly, *S_Queue_length* provides information about the length of the message queue. Finally, *S_Delta_queue* senses whether the length of the message queue is increasing or decreasing. Nodes for the internal status of components, such as the file system and the message queue, are connected via sensor and health nodes. The behavior nodes for system oscillation and delay build the foundation for reasoning about this and similar scenarios.

Figure 19 shows the temporal traces of a file system-induced fault scenario (Schumann, Morris, Mbaya, Mengshoel, & Darwiche, 2011) in simulation. The flight controller’s pitch-up and pitch-down commands to the actuators (top panel) are impacted by faults originating from the file system, causing the aircraft to oscillate up and down rather than maintaining the desired altitude. For the purpose of this experiment, we set the file system to almost full at the start of the simulation run; no other faults or errors occur. After about 30 seconds, the delays caused by the message queue have accumulated in such a way that flight-control induced oscillations of the aircraft occur, indicated by recurrent climbs and descends (middle panel). Eventually, these altitude oscillations are detected and picked up by the Fast Fourier Transform, and a signal is sent to *S_Oscillation*. The bottom panel of Figure 19 shows the posteriors for selected health nodes. It indicates that the actual aircraft sensors and actuators are healthy. However, the health status of the software (blue) decreases substantially a little after 100 seconds, indicating a problem in the on-board software. In this scenario, the health of the file system and of the message queue, when considered individually, do not drop significantly. Also, the software itself does not flag any error.

9. Conclusions

We presented a three-pronged approach to sensor and software health management in real time, on-board a UAS. Health models are constructed in a modular and scalable manner using a number of different building blocks. Major block types provide advanced capabilities of temporal logic runtime monitors, model-based analysis and signal processing, and powerful probabilistic reasoning using Bayesian networks. For our overarching design requirements of unobtrusiveness, responsiveness, and realizability, we automatically transform the health model into efficient FPGA hardware designs. We demonstrated the capabilities of this approach on a set of

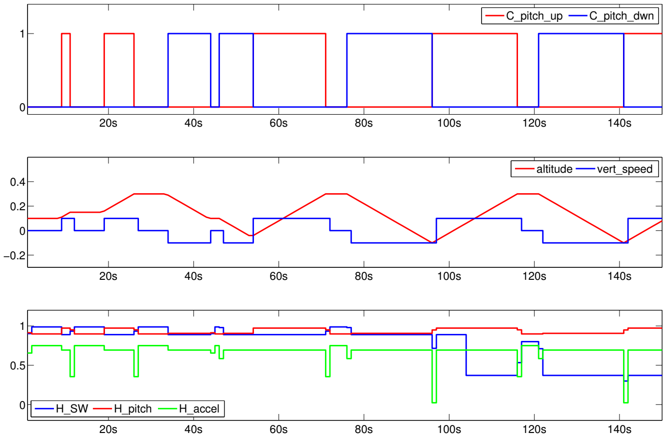


Figure 19. Traces of simulation experiment with file-system related failure scenario. Top panel: actuator messages sent through the message queue. Middle panel: vertical speed and altitude of the aircraft showing oscillations. Bottom panel: posterior probabilities of selected health nodes.

requirements and flight rules, both for sensor and software health management. We presented experimental results for this approach using actual data from the NASA Swift UAS.

Our approach enables the designer to build complex models and reasoning modes. For example, temporal reasoning over the results of probabilistic health outputs can be formulated easily: $(alt < 1000\text{ft}) \rightarrow \Diamond_{[0,10s]}(P(\text{H\_laserAlt} = \text{healthy}) > 0.8)$ would require a working laser altimeter at altitudes of less than 1000ft. In a similar manner, results of prognostics components can be smoothly integrated into our framework.

However, the results shown here are only the first steps towards a real-time on-board sensor and software health management system. For the proof of concept demonstration in this paper, we analyzed recorded data streams from the Swift UAS on the ground as if they were happening in real time. There are two clear options for reading this data on-board the Swift UAS instead: reading sensor data passed on the common bus or having sensor data sent to our framework by the flight computer. In the near future, we plan to define and build unobtrusive read-only interfaces that will enable us to get real-time sensor and software data from the common bus or flight computer while providing the guarantee that under no circumstances would our framework disturb the bus or any other UAS component. This is a major requirement for certification and carrying out actual flight tests on the Swift UAS.

On a broader level, research needs to be performed on how to automatically generate advanced system health management models from requirements, designs, and architectural artifacts. In particular for monitoring the health of a complex and large software system, automatic model generation is essential. We are confident that our approach, which allows us to combine monitoring of sensors, prognostics, and software while separating out (model-based) signal processing, temporal, and probabilistic reasoning will substantially facilitate the development of improved and powerful on-board health management systems for unmanned aerial systems.

Acknowledgments

This work was in part supported by NASA NRA grant NNX08AY50A "ISWHM: Tools and Techniques for Software and System Health Management." The work of Thomas Reinbacher was supported within the FIT-IT project CevTes managed by the Austrian Research Agency FFG under grant 825891.

References

Alur, R., & Henzinger, T. A. (1990). Real-time Logics: Complexity and Expressiveness. In *Lics* (pp. 390–401). IEEE Computer Society Press.

Barringer, H., et al. (Eds.). (2010). *Runtime verification - first international conference, rv 2010, proceedings* (Vol. 6418). Springer Verlag.

Basin, D., Klaedtke, F., Müller, S., & Pfitzmann, B. (2008). Runtime monitoring of metric first-order temporal properties. In *Fsttcs* (pp. 49–60).

Basin, D., Klaedtke, F., & Zălinescu, E. (2011). Algorithms for monitoring real-time properties. In *Rv* (Vol. 7186, pp. 260–275). Springer Verlag.

Bauer, A., Leucker, M., & Schallhart, C. (2010). Comparing LTL semantics for runtime verification. *J. Log. and Comput.*, *20*(3), 651–674.

Bekkerman, R., Bilenko, M., & Langford, J. (Eds.). (2011). *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press.

Bellman, R. (1958). On a routing problem. *Quarterly of Applied Mathematics*, *16*, 87–90.

Bolton, M., & Bass, E. (2013). Evaluating human-human communication protocols with miscommunication generation and model checking. In *Nasa formal methods symposium* (Vol. TBD). Springer.

Brown, R., & Hwang, P. (1997). *Introduction to random signals and applied kalman filtering* (3rd ed.). John Wiley & Sons.

Chavira, M., & Darwiche, A. (2005). Compiling Bayesian networks with local structure. In *Proceedings of the 19th international joint conference on artificial intelligence (ijcai)* (pp. 1306–1312).

Chavira, M., & Darwiche, A. (2007). Compiling Bayesian networks using variable elimination. In *Proc. of the twentieth international joint conference on artificial intelligence (ijcai-07)* (pp. 2443–2449). Hyderabad, India.

Darwiche, A. (2001). Recursive conditioning. *Artificial Intelligence*, *126*(1-2), 5-41.

Darwiche, A. (2003). A differential approach to inference in Bayesian networks. *Journal of the ACM*, *50*(3), 280–305.

Darwiche, A. (2009). *Modeling and reasoning with bayesian* *networks*. Cambridge, UK: Cambridge University Press.

Dawid, A. P. (1992). Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, *2*, 25–36.

Divakaran, S., D'Souza, D., & Mohan, M. R. (2010). Conflict-tolerant real-time specifications in Metric Temporal Logic. In *Time* (p. 35-42). IEEE Computer Society Press.

Federal Aviation Administration. (2013). *Federal Aviation Regulation §91.*

Gan, X., Dubrovin, J., & Heljanko, K. (2011). A symbolic model checking approach to verifying satellite onboard software. *ECEASST*, *46*.

Geilen, M. (2003). An Improved On-The-Fly Tableau Construction for a Real-Time Temporal Logic. In *Cav* (Vol. 2725, pp. 394–406). Springer Verlag.

Havelund, K. (2008). Runtime verification of C programs. In *Testcom/fates* (pp. 7–22). Springer Verlag.

Huang, C., & Darwiche, A. (1994). Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, *15*(3), 225-263.

Huang, J., Chavira, M., & Darwiche, A. (2006). Solving MAP exactly by searching on compiled arithmetic circuits. In *Proc. of the twenty-first national conference on artificial intelligence* (p. 143-148). Boston, MA.

Ippolito, C., Espinosa, P., & Weston, A. (2010, April). Swift UAS: An electric UAS research platform for green aviation at NASA Ames Research Center. In *CAFE eas iv.*

Jensen, F. V., Lauritzen, S. L., & Olesen, K. G. (1990). Bayesian updating in causal probabilistic networks by local computations. *SIAM Journal on Computing*, *4*, 269–282.

Jeon, H., Xia, Y., & Prasanna, V. K. (2010). Parallel exact inference on a CPU-GPGPU heterogenous system. In *Proc. of the 39th international conference on parallel processing* (p. 61-70).

Johnson, D. (2007). *Raptors Arrive at Kadena.* Retrieved from http://www.af.mil/news/story.asp?storyID=123041567

Kask, K., Dechter, R., & Gelfand, A. (2010). BEEM: bucket elimination with external memory. In *Proc. of the 26th annual conference on uncertainty in artificial intelligence (uai-10)* (pp. 268–276).

Kozlov, A. V., & Singh, J. P. (1994). A parallel Lauritzen-Spiegelhalter algorithm for probabilistic inference. In *Proc. of the 1994 acm/ieee conference on supercomputing* (pp. 320–329).

Kulesza, Z., & Tylman, W. (2006). Implementation of Bayesian network in FPGA circuit. In *Mixdes* (p. 711 -715). IEEE Computer Society Press.

Lauritzen, S. L., & Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, *50*(2), 157–224.

Li, Z., & D'Ambrosio, B. (1994). Efficient inference in Bayes nets as a combinatorial optimization problem. *International Journal of Approximate Reasoning*, *11*(1), 55–81.

Lichtenstein, O., Pnueli, A., & Zuck, L. (1985). The glory of the past. In *Logics of programs* (Vol. 193, p. 196-218). Springer Verlag.

Lin, M., Lebedev, I., & Wawrzynek, J. (2010). High-throughput Bayesian computing machine with reconfigurable hardware. In *Fpga* (pp. 73–82). ACM Press.

Linderman, M. D., Bruggner, R., Athalye, V., Meng, T. H., Asadi, N. B., & Nolan, G. P. (2010). High-throughput Bayesian network learning using heterogeneous multicore computers. In *Proc. of the 24th acm international conference on supercomputing* (p. 95-104).

Lindsey, A. E., & Pecheur, C. (2004). Simulation-based verification of autonomous controllers via livingstone pathfinder. In K. Jensen & A. Podelski (Eds.), *Proceedings tacas 2004* (Vol. 2988, pp. 357–371). Springer.

Low, Y., Gonzalez, J., Kyrola, A., Bickson, D., Guestrin, C., & Hellerstein, J. (2010). GraphLab: A new framework for parallel machine learning. In *Proc. of the 26th annual conference on uncertainty in artificial intelligence (uai-10)* (p. 340-349).

Maler, O., Nickovic, D., & Pnueli, A. (2005). Real time temporal logic: Past, present, future. In *Formats* (Vol. 3829, p. 2-16). Springer Verlag.

Maler, O., Nickovic, D., & Pnueli, A. (2007). On synthesizing controllers from bounded-response properties. In *Cav* (Vol. 4590, p. 95-107). Springer Verlag.

Maler, O., Nickovic, D., & Pnueli, A. (2008). Checking temporal properties of discrete, timed and continuous behaviors. In *Pillars of comp. science* (p. 475-505). Springer Verlag.

Mengshoel, O. J. (2007). Designing resource-bounded reasoners using Bayesian networks: System health monitoring and diagnosis. In *Proceedings of the 18th international workshop on principles of diagnosis (dx-07)* (pp. 330–337). Nashville, TN.

Mengshoel, O. J., Chavira, M., Cascio, K., Poll, S., Darwiche, A., & Uckun, S. (2010). Probabilistic model-based diagnosis: An electrical power system case study. *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, *40*(5), 874–885.

Mengshoel, O. J., Darwiche, A., Cascio, K., Chavira, M., Poll, S., & Uckun, S. (2008). Diagnosing faults in electrical power systems of spacecraft and aircraft. In *Proc. of the twentieth innovative applications of artificial intelligence conference (iaai-08)* (pp. 1699–1705). Chicago, IL.

Mengshoel, O. J., Roth, D., & Wilkins, D. C. (2011). Portfolios in stochastic local search: Efficiently computing most probable explanations in Bayesian networks. *Journal of Automated Reasoning*, *46*(2), 103–160.

Mengshoel, O. J., Wilkins, D. C., & Roth, D. (2011). Initialization and restart in stochastic local search: Computing a most probable explanation in Bayesian networks. *IEEE Transactions on Knowledge and Data Engineering*.

Musliner, D., Hendler, J., Agrawala, A. K., Durfee, E., Strosnider, J. K., & Paul, C. J. (1995, January). The challenges of real-time AI. *IEEE Computer*, *28*, 58–66. Retrieved from citeseer.comp.nus.edu.sg/article/musliner95challenges.html

Namasivayam, V. K., & Prasanna, V. K. (2006). Scalable parallel implementation of exact inference in Bayesian networks. In *Proc. of the 12th international conference on parallel and distributed system* (p. 143-150).

Park, J. D., & Darwiche, A. (2004). Complexity results and approximation strategies for MAP explanations. *Journal of Artificial Intelligence Research (JAIR)*, *21*, 101-133.

Pasricha, R., & Sharma, S. (2009). An FPGA-based design of fixed-point Kalman filter. *ICGST International Journal* *on Digital Signal Processing, DSP*, *9*, 1–9.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Mateo, CA: Morgan Kaufmann.

Pike, L., Niller, S., & Wegmann, N. (2011). Runtime verification for ultra-critical systems. In *Rv* (Vol. 7186, pp. 310–324). Springer Verlag.

Poll, S., Patterson-Hine, A., Camisa, J., Garcia, D., Hall, D., Lee, C., ... Koutsoukos, X. (2007). Advanced diagnostics and prognostics testbed. In *Proc. of the 18th international workshop on principles of diagnosis (dx-07)* (pp. 178–185). Nashville, TN.

Reinbacher, T., Rozier, K. Y., & Schumann, J. (2013). Temporal-logic based runtime observers for system health mangement of real-time systems.. (under submission)

Ricks, B. W., & Mengshoel, O. J. (2009a). The diagnostic challenge competition: Probabilistic techniques for fault diagnosis in electrical power systems. In *Proc. of the 20th international workshop on principles of diagnosis (dx-09)* (pp. 415–422). Stockholm, Sweden.

Ricks, B. W., & Mengshoel, O. J. (2009b). Methods for probabilistic fault diagnosis: An electrical power system case study. In *Proc. of annual conference of the phm society, 2009 (phm-09).* San Diego, CA.

Ricks, B. W., & Mengshoel, O. J. (2010). Diagnosing intermittent and persistent faults using static Bayesian networks. In *Proc. of the 21st international workshop on principles of diagnosis (dx-10).* Portland, OR.

Ristic, B., Arulampalam, S., & Gordon, N. (2004). *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House.

Rozier, K. Y. (2011). Linear temporal logic symbolic model checking. *Computer Science Review*, *5*(2), 163–203.

Rozier, K. Y., & Vardi, M. Y. (2010, March). LTL satisfiability checking. *International Journal on Software Tools for Technology Transfer (STTT)*, *12*(2), 123 - 137. Retrieved from http://dx.doi.org/10.1007/s10009-010-0140-3 doi: DOI10.1007/s10009-010-0140-3

RTCA. (2012). *DO-178C/ED-12C: Software considerations in airborne systems and equipment certification.* Retrieved from http://www.rtca.org

Schumann, J., Mbaya, T., Mengshoel, O. J., Pipatsrisawat, K., Srivastava, A., Choi, A., & Darwiche, A. (2013). Software health management with Bayesian networks. *Innovations in Systems and Software Engineering*, *9*(2), 1-22.

Schumann, J., Mengshoel, O. J., & Mbaya, T. (2011). Integrated software and sensor health management for small spacecraft. In *Proc. of the 2011 ieee fourth international conference on space mission challenges for information technology* (pp. 77–84).

Schumann, J., Morris, R., Mbaya, T., Mengshoel, O. J., & Darwiche, A. (2011). *Report on Bayesian approach for dynamic monitoring of software quality and integration with advanced IVHM engine for ISWHM* (Tech. Rep.). USRA/RIACS.

Shenoy, P. P. (1989). A valuation-based language for expert systems. *International Journal of Approximate Reasoning*, *3*(5), 383 – 411.

Silberstein, M., Schuster, A., Geiger, D., Patney, A., & Owens, J. D. (2008). Efficient computation of sum-products on GPUs through software-managed cache. In *Proc. of the 22nd acm international conference on supercomputing* (pp. 309–318).

Thati, P., & Roşu, G. (2005). Monitoring algorithms for Metric Temporal Logic specifications. *ENTCS*, *113*, 145–162.

Whittle, J., & Schumann, J. (2004, December). Automating the implementation of Kalman filter algorithms. *ACM Transactions on Mathematical Software*, *30*(4), 434–453.

Xia, Y., & Prasanna, V. K. (2007). Node level primitives for parallel exact inference. In *Proc. of the 19th international symposium on computer architecture and high performance computing* (p. 221-228).

Zhang, N. L., & Poole, D. (1996). Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, *5*, 301-328. Retrieved from citeseer.nj.nec.com/article/zhang96exploiting.html

Zhao, Y., & Rozier, K. Y. (2012). Formal specification and verification of a coordination protocol for an automated air traffic control system. In *Proceedings of the 12th international workshop on automated verification of critical systems (avocs 2012)* (Vol. 53). European Association of Software Science and Technology.

Zheng, L., & Mengshoel, O. J. (2013, August). Optimizing parallel belief propagation in junction trees using regression. In *Proc. of 19th ACM SIGKDD conference on knowledge discovery and data mining (kdd-13).* Chicago, IL.

Zheng, L., Mengshoel, O. J., & Chong, J. (2011). Belief propagation by message passing in junction trees: Computing each message faster using gpu parallelization. In *Proc. of the 27th conference in uncertainty in artificial intelligence (uai-11).* Barcelona, Spain.

Degradation Modeling and Remaining Useful Life Prediction of Electrolytic Capacitors under Thermal Overstress Condition Using Particle Filters

Hamed Khorasgani[1], Chetan Kulkarni[2], Gautam Biswas[3], José R. Celaya[4], and Kai Goebel[5]

[1,3] *Vanderbilt University, Nashville, TN, 37235, USA*
hamed.g.khorasgani@vanderbilt.edu
gautam.biswas@vanderbilt.edu

[2,4] *SGT Inc. NASA Ames Research Center, Moffett Field, CA, 94035, USA*
chetan.kulkarni@vanderbilt.edu
jose.r.celaya@nasa.gov

[5] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
kai.goebel@nasa.gov

Abstract

Prognostic and remaining useful life (RUL) predictions for electrolytic capacitors under thermal overstress condition are investigated in this paper. In the first step, the degradation process is modeled as a physics of failure process. All of the relevant parameters and states of the capacitor are considered during the degradation process. A particle filter approach is utilized to derive the dynamic form of the degradation model and estimate the current state of capacitor health. This model is then used to get more accurate estimation of the Remaining Useful Life (RUL) of the capacitors as they are subjected to the thermal stress conditions. The paper includes an experimental study, where the degradation of a set of identical capacitors under thermal overstress conditions is studied to demonstrate and validate the performance of the degradation modeling approach.

1. Introduction

Prognostic and Remaining Useful Life (RUL) prediction is essential for determining the safety and reliability of critical systems. In the cases where the operators have access to the system RUL prediction, it becomes easier for them to establish condition-based maintenance schedules, and thus avoid failures and expensive system downtime. For autonomous systems, RUL prediction provides necessary information for the system to schedule future tasks and missions in an effective manner.

Electronic systems need reliable power supplies. Failure in the power supplies can damage other system elements through a cascading process. Many electronic modules are sensitive to the potential level of the supplied voltage and undesired change in this level can cause failure in these modules as well. Also unpredicted voltage variations can damage processors and make control modules in the systems unreliable. Switch mode power supplies (SMPS) can provide different voltage levels from a single power source by changing the duty cycle of switching. They are also efficient, light and have a small size. However the small size affects heat dissipation, and reliability becomes an important issue in these systems. Consequently RUL predictions for different elements of these systems are becoming increasingly important (Goebel et al., 2008; Celaya et al., 2010; Kulkarni et al., 2009). One of the key elements in power supply modules and dc-dc converters are electrolytic capacitors. Failures in these elements is one of main reasons for the module failures (Goodman et al., 2005). For the prognostic and RUL prediction of electrolytic capacitors like any other element in addition to the current state of the element, it is necessary to predict its future behavior. If we can determine and model the main reasons for the degradation of the capacitors, future behavior prediction becomes a feasible task.

Evaporation of the electrolyte is a main reason for the degradation and finally complete failure of the capacitor. In fact evaporation of the electrolyte decreases the effective surface area of the capacitors and consequently their capacitance. One of the complexities of this problem is that the evaporation

Hamed Khorasgani et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

of the electrolyte in the closed can of the capacitor causes internal pressure increase, which affects the evaporation rate. It is also possible that due to the internal pressure the capacitor body pops exposing the electrolyte to the atmosphere. Under these circumstances, the capacitor electrolyte first shows abrupt increase in evaporation rate, but after its top pops, the evaporation rate becomes constant. Work on capacitor degradation modeling in (Celaya et al., 2012) considered the evaporation rate to be constant and derived the degradation model for electrolytic capacitors under electrical overstress condition. With this simplification, the derived model is linear and the authors could apply the Kalman filter scheme to estimate the current state of health of the system. In this paper, the evaporation rate is not held to a constant value. Two different degradation trajectories are considered: one for the situation where the capacitor casing remains perfect, therefore, the evaporation rate decreases with time, and a second when the casing cracks, therefore, the evaporation rate for the capacitor electrolyte remains constant. A general model is derived by combining these two models. Since the general derived degradation model is nonlinear, unlike (Celaya et al., 2012), we adopt the particle filter approaches (Arulampalam et al., 2002) for state estimation of the nonlinear dynamic system. The particle filter approach is also utilized to estimate parameters of the model for each capacitor. To determine the effectiveness and performance of the particle filter approach, we used experimental data generated in (Kulkarni et al., Sep 2012) to validate our approach.

The paper is structured as follows. Section 2 discusses the degradation processes and the physics of failure model corresponding to the degradation processes. Section 3 discusses the particle filter approach for state estimation in the prognostic algorithm. Particle filtering method for parameter estimation is discussed in this section as well. An algorithm for computing the RUL of the capacitor is presented as the last part of this section. Section 4 is about the experimental setup and parameter estimation. 15 electrolytic capacitors are studied under thermal over stress condition. The leave one out method is utilized to validate the parameters of the model and the derived model is used to make prognostic predictions. Section 5 demonstrates RUL prediction results and the conclusions are presented in Section 6.

2. Degradation Modeling

Prognostics approaches start with the current state of a component or device and employ systematic methods to predict future system behavior. Having an accurate model of the degradation process provides a methodology for predicting future component behavior. In this section, the degradation model of the electrolytic capacitor is discussed and developed. We start with the structure of electrolytic capacitors and then discuss the degradation process. Last, a discretized form of the continuous degradation model is derived.

2.1. Electrolytic Capacitors Structure and Capacitance

If we open the aluminium can of an electrolytic capacitor we see the anode and cathode foils, and the electrolyte soaked in the separator paper all wrapped up together as shown in Figure 1. In the electrolytic capacitors that we study, ethyl glycol is the chosen electrolyte. The cathode and anode foils are aluminium. Also, the oxide layer on the surface of the anode acts as the dielectric. This oxide layer in contact with the electrolyte acts as a perfect insulator. By wrapping up the papers high capacitance is achieved in the minimum space.

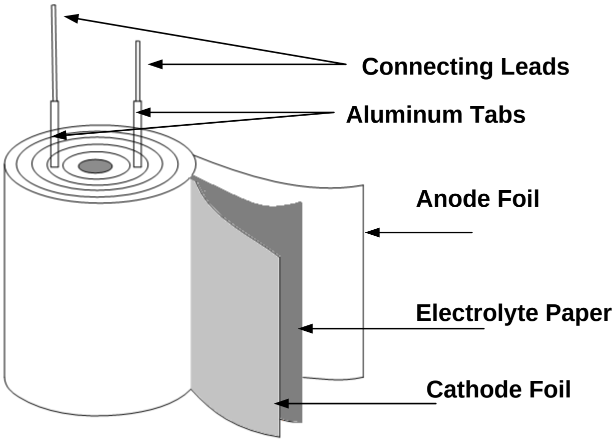


Figure 1. Cylindrical Electrolytic Capacitor Structure (Kulkarni et al., Sep 2012).

The capacity of the capacitance is defined in terms of its geometric structure. From the first principles of electromagnetism the total lumped capacitance of a flat plate electrolytic capacitor is:

$$C = \frac{\epsilon A}{d_o}, \tag{1}$$

where ϵ is dielectric constant of the electrolyte, A is the effective surface of the electrolyte and d_o is the oxide thickness. Rolling the plates of electrolytic capacitors double the capacitance of these capacitors (Tasca, 1981). This improvement achieves due to utilizing both sides of foils in the new structure. Consequently the capacitance of the presented structure in Figure 1 can be expressed as follows.

$$C = \frac{2\epsilon A}{d_o}. \tag{2}$$

Assuming that the separator paper thickness is negligible and the space between anode and cathode is completely filled by the electrolyte the volume of the electrolyte can be computed as:

$$V = A * d, \tag{3}$$

where d is the distance between anode and cathode foils. Therefore, the capacitance can be expressed based on electrolyte volume by:

$$C = \frac{2\epsilon V}{d_o d}. \tag{4}$$

Equation (4) demonstrates how the evaporation of the electrolyte decreases the capacitance of the electrolytic capacitors. The electrolytic capacitor capacitance, C decreases directly in proportion to electrolyte volume, V, as it evaporates due to raised ambient temperatures. In the next section, we study the evaporation process for the electrolyte in greater detail.

2.2. Modeling Electrolyte Evaporation

The rate of evaporation and consequent decrease in the liquid volume depends on the surface area of the liquid, its molecular volume, and the evaporation rate (Rdner et al., 2002). So we can write:

$$\frac{dV}{dt} = -AJ\omega, \tag{5}$$

where A is the surface of electrolyte, J is the evaporation rate and ω is the volume of electrolyte (ethyl glycol) molecule. If we substitute A by $\frac{V}{d}$ in (5) we have:

$$\frac{dV}{dt} = -J\omega\frac{V}{d} \tag{6}$$

An important parameter in the derived equation is the evaporation rate. Typically the evaporation rate is not a constant parameter; it increases as the ambient temperature increases, therefore, the temperature at the capacitor core increases. In fact, if the molecules of electrolyte have more kinetic energy the probability that they leave the surface increases and consequently the evaporation rate increases. In our study, the ambient temperature in the experimental chamber is maintained constant. The other factor which affects the evaporation rate is pressure. Increasing the pressure at the surface of the liquid decreases the probability that molecules leave the surface, which means that the evaporation rate decreases. Further, a large pressure increase has other consequences, e.g., the top of the capacitor casing may pop. This releases all of the built up gases, bringing the surface pressure on the electrolyte surface back to the atmospheric pressure levels. We discuss the evaporation rates for two different scenarios: (1) capacitor casing not pierced, implying the inside of the capacitor is a closed system and (2) capacitor casing damaged and popped. $Capacitor\ as\ Closed\ System:$ As long as the capacitor casing is not pierced the capacitor is a closed system. In this situation, by evaporation the escaped molecules accumulate in the can and increase the pressure on the electrolyte surface. The increase in the pressure decreases the evaporation rate. The decrease in the evaporation rate continues and after some time it becomes so small that practically the evaporation stops. At this point the numbers of molecules which leave the liquid and return to it are almost the same and the vapor is said to be saturated. In the enclosed system, we use the exponential function to model the evaporation rate, i.e:

$$J = J_0 e^{-\beta t}, \tag{7}$$

where J_0 is the initial value of the evaporation rate which is function of temperature and density of the liquid. β determines how fast the evaporation rate decreases and depends on the volume of the enclosed space. $Open\ System\ with\ Capacitor\ Casing\ Pierced:$ The primary reason to covering the capacitors with an aluminum can is to protect the electrolyte from evaporation (Kulkarni et al., Jun 2012). However it is possible that a large increase in the internal pressure can pop or crack the cover in some way and the electrolyte escapes into to the atmosphere. When this happens, the evaporation rate after an initial increase becomes constant because the capacitor can is not a closed space anymore. In this case the evaporation rate is:

$$J = J_{open}, \tag{8}$$

where J_{open} depends on the temperature, atmosphere pressure and liquid concentration. As a next step we discuss the capacitor degradation model next.

2.3. State Space Model for Capacitance Degradation

To model capacitor degradation, a state space model with capacitance and evaporation rate as the state variables is considered. Using (4) one can write:

$$\frac{dC}{dt} = \frac{2\epsilon}{d_o d}\frac{dV}{dt}. \tag{9}$$

And by substituting (6) in (9) we have:

$$\frac{dC}{dt} = -2J\omega\epsilon\frac{V}{d_o d^2}. \tag{10}$$

Using (4) and doing some algebraic manipulations we can write:

$$\frac{dC}{dt} = -J\omega\frac{C}{d}. \tag{11}$$

For the enclosed case J can derived from (7) as:

$$\frac{dJ}{dt} = -\beta J_0 e^{-\beta t}. \tag{12}$$

By substituting (7) in (12) we have:

$$\frac{dJ}{dt} = -\beta J. \tag{13}$$

If the can cracks or pops up the evaporation rate is constant and the capacitance dynamic is:

$$\frac{dC}{dt} = -J_{open}\omega\frac{C}{d}. \tag{14}$$

2.4. Converting to Discrete-Time Model

Since measurements are sampled in discrete times we derive a discrete time model of the state space representation. Using a first-order approximation:

$$\frac{dC}{dt}(t_k) = \frac{C(t_{k+1}) - C(t_k)}{t_{k+1} - t_k}. \tag{15}$$

Consequently from (11) for the enclosed case we have:

$$C(t_{k+1}) = C(t_k) - J(t_k)\omega \frac{C(t_k)}{d}(t_{k+1} - t_k). \tag{16}$$

Similarly, using (13) the dynamic equation for J in enclosed space is derived as:

$$J(t_{k+1}) = J(t_k) - \beta J(t_k)(t_{k+1} - t_k). \tag{17}$$

Considering (14) for the capacitor case popped open we have:

$$C(t_{k+1}) = C(t_k) - J_{open}\omega \frac{C(t_k)}{d}(t_{k+1} - t_k) \tag{18}$$

2.5. General Model for the Degradation

An accurate model of capacitor degradation would use the enclosed degradation equation till the capacitor encasing popped, and at the popping time switch to the open model where the evaporation rate becomes constant. Since we do not know the exact time that the switch from the closed to open situation occurs, we employ a blended model that combines the enclosed and open degradation models, i.e.,

$$C = \alpha(t_k)C_{closed} + (1 - \alpha(t_k))C_{open}. \tag{19}$$

$\alpha(t_k)$ starts at a value close to 1.0 for t_k small, and gradually decrease as time advances. How the parameter α changes with time may be unknown, for the sake of simplicity we assume α decreases linearly with slope of c over time as represented by (20) . The value of c may be determined by empirical studies, or based on expert-supplied knowledge.

$$\alpha(t_{k+1}) = \begin{cases} \alpha(t_k) - c(t_{k+1} - t_k) & t_{k+1} \leq \frac{1}{c} \\ \alpha(t_k) & t_{k+1} > \frac{1}{c}. \end{cases} \tag{20}$$

Therefore, an approximation of the capacitor degradation model can be expressed as:

$$\begin{aligned} C(t_{k+1}) &= \alpha(t_k) * [C(t_k) - J(t_k)\omega \frac{C(t_k)}{d}(t_{k+1} - t_k)] + \\ &(1 - \alpha(t_k)) * [C(t_k) - J_{open}\omega \frac{C(t_k)}{d}(t_{k+1} - t_k)] \\ J(t_{k+1}) &= J(t_k) - \beta J(t_k)(t_{k+1} - t_k) \\ \alpha(t_{k+1}) &= \begin{cases} \alpha(t_k) - c(t_{k+1} - t_k) & t_{k+1} \leq \frac{1}{c} \\ \alpha(t_k) & t_{k+1} > \frac{1}{c}. \end{cases} \end{aligned} \tag{21}$$

3. Prognostics Algorithm

Using the degradation model we can design a filter to estimate the current state of health of the system. Using this estimated state of health and degradation model, we can predict the future behaviour of the capacitor and using end of life threshold we can estimate the remaining useful life. As discussed earlier, the degradation model includes three state variables that change with time, capacitance, C, evaporation rate, J and combination factor, α. Since we have run experiments and collected degradation data on a set of identical electrolytic capacitors, we can use our filter model in a predict-estimate-update loop to refine the degradation model as new measurements on the system become available. Applying the predict-estimate-update loop, we hypothesize that the estimation of RUL becomes more accurate as more data is obtained on capacitor degradation. Therefore, a combination of a model- and data-driven approach to prognostics will likely result in more accurate and general degradation models than if we employed pure data-driven methods (which don't generalize easily) and pure model-based approaches (which are hard to tune accurately without the availability of data).

Kalman filters have been proven to be optimal state estimators and predictors for linear time-invariant systems (Arulampalam et al., 2002). However, the state equations for capacitor degradation are nonlinear, therefore a Kalman filter approach will have to be approximated by extended Kalman filters (EKF). (Julier & Uhlmann , 1997) used EKF framework to develop unscented Kalman filters (UKF) which is another approximation of Kalman filters for state estimation in nonlinear systems. This method assume Gaussian distribution for the system states and utilizes a set of carefully chosen sample points, propagate them through the non-linear dynamic and uses the result to re-estimate the parameters of the Gaussian distribution in each step. These samples are chosen in a way to capture the exact mean and covariance of the Gaussian distribution completely and when propagate through the nonlinear dynamic capture the mean and covariance of the posterior distribution to the 3rd order Taylor series expansion approximation accurately. Since EKF uses first order Taylor series approximation it is expected that UKF exhibits better performance in the state estimation of nonlinear systems with Gaussian inputs (Wan & Van Der Merwe, 2000). In previous work, we have used unscented Kalman filters for state estimation and obtained fairly accurate degradation models and RUL estimates for both Electrical and Thermal overstress test conditions (Kulkarni, 2013). In this paper we adopt a more powerful approach to modeling and tracking non linear behavior evolution: the Particle filter that uses a sequential Monte Carlo approach for state estimation and does not assume Gaussian inputs for the system. Similar to EKF and UKF, Particle filter provides a suboptimal solution for the state estimation in nonlinear systems, however it is proved that by increasing the number of samples (particles) its solution approaches the optimal Bayesian estimation (Arulampalam et al., 2002).

3.1. Particle Filter

In order to estimate the probability density distribution (pdf) of the state variables of a nonlinear system, the particle filter approach uses a set of particles (samples) and a set of associated weights representing the discrete probability masses of the particles. Particle Filter updates these particles and weights in each step to follow the evolution of the states of the dynamic system. In fact, particle filters use Monte Carlo method to implement a recursive Bayesian filter to estimate the pdf of the state variables. To implement particle filters, it is assumed that the dynamic system can be presented as a first order Markov process:

$$\begin{aligned} x_k &= f(x_{k-1}) + \omega_k \\ y_k &= h(x_k) + \upsilon_k, \end{aligned} \tag{22}$$

where x_k is the system states in step time k, y_k is the measurement and ω_k and υ_k represent system and sensor noises respectively.

$Sequential\ Importance\ Sampling\ (SIS)$: SIS is an algorithm for implementing recursive filtering based on Monte Carlo method. The main idea is to approximate posterior density function $p(x_k|y_{1:k})$ by a set of random samples or particles $\{x_k^i\}_{i=1}^P$ and their associated weights $\{\omega_k^i\}_{i=1}^P$. As mentioned earlier as the number of these particles become very large the Monte Carlo characterization approaches to the posterior density function and SIS algorithm solution approaches the optimal Bayesian estimation.

$$\begin{aligned} p(x_k|y_{1:k}) \approx \sum_{i=1}^{P} w_k^i \delta(x_k - x_k^i) \\ \sum_{i=1}^{P} w_k^i = 1, \end{aligned} \tag{23}$$

where x_k^i are the particles and w_k^i are the normalized weights associated with them. These weights update based on importance distribution function $\pi(x_k|x_{0:k-1}, y_{1:k})$ in each step (Arulampalam et al., 2002) as follows :

$$w_k^i \propto w_{k-1}^i \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{\pi(x_k^i|x_{0:k-1}^i, y_{1:k})}. \tag{24}$$

$Degeneracy\ Problem\ and\ Resampling$: Degeneracy problem happens when the variance of importance weights keep increasing over time and consequently after a certain number of step times most of the particles will have negligible importance weights. In this situation a fairly large percentage of computational effort would be devoted to update the weights associated with particles which have no meaningful contribution to the result. The degeneracy problem is not avoidable in SIS algorithm (Ristic et al., 2004). To overcome to this problem a resampling process is considered in each step time to replace the particles with low importance weight with particles which have higher importance weights. Resampling is a mapping from $\{x_k^i, \omega_k^i\}_{i=1}^P$ to $\{x_k^{i*}, \frac{1}{P}\}_{i=1}^P$ where new particles x_k^{i*} are chosen from the set of previous particles randomly with the probability equal to the importance weight of the particle so we have:

$$P\{x_k^{i*} = x_k^j\} = \omega_k^j. \tag{25}$$

$Sampling\ Importance\ Resampling\ (SIR)$: By choosing importance distribution function equal to $p(x_k|x_{k-1})$, one can rewrite equation (24) as:

$$w_k^i \propto w_{k-1}^i p(y_k|x_k^i). \tag{26}$$

Also considering resampling procedure at each step time we assign $w_{k-1} = \frac{1}{P}$ for all the particle weights so we have:

$$w_k^i \propto p(y_k|x_k^i). \tag{27}$$

This filter is proposed by (Gordon et al., 1993) and is called SIR or bootstrap filter. To implement SIR filter we only need to know states space dynamic, measurements and noise model (22) and the likelihood function $p(y_k|x_k)$.

$Particle\ Filters\ for\ Parameter\ Estimation$: State estimation by particle filters is discussed in this section. Particle filters can be used for parameter estimation in nonlinear systems as well. The parameters do not change over time significantly so its dynamic in state space representation can be written as:

$$\begin{aligned} \rho_k &= \rho_{k-1} + \omega_k \\ y_k &= h(\rho_k, x_k) + \upsilon_k, \end{aligned} \tag{28}$$

where ρ represents system parameters and x is system states. ω and υ represents parameters uncertainty and sensor noise models respectively and y is the measurements. Based on our knowledge about the variance and expected value of the parameters we design ω_k and consider proper initial values for the parameters and the rest of the process is quit similar with state estimation procedure. Parameter estimation can be done simultaneously with state estimation or off line. In the off line scenario, we assume that the system states in different time steps are already measured and saved and we feed them as the inputs to equation (28) during the parameter estimation process. In the cases that states and parameters should be estimated simultaneously, parameters will be considered as the new states for the system and to calculate the value of the new states (which are the combination of original states and unknown parameters) in each time step we use the estimated value of them in previous time step as it was done in state estimation.

3.2. Remaining Useful Life Prediction

The particle filter estimates the current value of the capacitance considering the measurement values and the degradation model. It also updates the degradation model by providing a new value for evaporation rate and combination factor based on the measurement and the degradation model. To calculate the remaining useful life we use the current value of the capacitance as the initial value and the updated degradation model to predict the future behavior of the capacitor. Remaining useful life threshold is value provided by the factory which determines after a specific percentage of degradation in the capacitor the capacitor cant perform its normal function in the circuit. Having this value we can compare it with predicted capacitance of the electrolytic capacitor in future to estimate remaining useful life. The procedure is presented in figure 2.

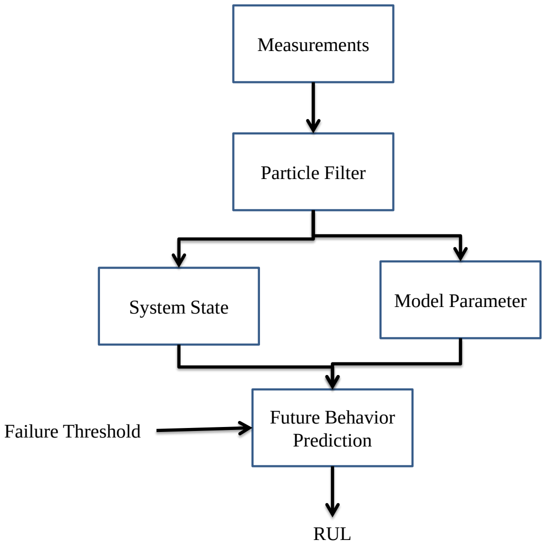


Figure 2. Prognostic flowchart.

4. SETUP AND MODEL ESTIMATION

In the first part of this section the experimental setup and the test conditions are presented. Then the parameters of the degradation method are estimated. Finally the remaining useful life prediction results are demonstrated and the performance of the algorithm is discussed.

4.1. Experimental Setup and Test Conditions

In order to present the performance of the proposed strategy for RUL prediction of the electrolytic capacitors 15 identical capacitors with $2200uF$ capacitance are utilized. The nominal working condition of the capacitance is $10V$ and 85^oC. The thermal over stress condition is provided by a temperature controlled chamber presented in figure 3. The capacitors were in 105^oC with 3.4% humidity factor for 3400 hours. And the capacitances of the capacitors were measured during the test regularly [13].

Figure 3. Thermal Chamber.

4.2. Parameters of the Degradation Model

In the derived degradation model for the electrolytic capacitor, thickness of electrolyte paper, d, can be measured for each class of capacitors after removing the cover. Volume of the electrolyte molecule is also a constant value that need to be determined based on the material used as the electrolyte. These parameters for the 15 identical capacitors that used in our case study are presented in Table 1. Dielectric constant, ϵ, was used in the modeling too, however since it did not appear in the final model for the degradation (21) we don't need to use it for RUL predictions.

There are also some other parameters in the degradation model which need to be determined in the prognostic design procedure. These unknown parameters as one can see in equation (21) are J_{open} , β and c. Also three initial conditions C_0 , J_0 and α_0 need to be determined. C_0 is considered equal to the nominal capacitance of each capacitor and J_0 is assumed equal to J_{open}. This approximation is not far away from reality because the internal pressure due to the evaporation is the main reason for decreasing J in the closed space and in the beginning there should not be significant evaporation. Finally α_0 is considered 1 because we assumed in the beginning of the thermal overstress test all the capacitors are healthy and non of the cans is not popped up. To estimate J_{open} , β and c for each capacitor the data associated with other 14 capacitors is considered and since the model is not linear, the particle filter scheme is applied for the estimation. The derived value for each capacitor is presented in Table 2. Figure 4 represents real degradation process and its corresponding degradation model for the first capacitor. It can be seen that the degradation model tracks real degradation process within an acceptable range of error. Mean square error between each capacitor's model derived by the parameters of Table 2 and its real degradation process is calculated by (29) and is presented in Table 3.

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(C_{tk} - C_{tk}^{'})^2, \tag{29}$$

where C_{tk} is the capacitance of the capacitor measured at time sample t_k and $C_{tk}^{'}$ is the value of capacitor obtained

Table 1. Parameter values of the degradation model

Parameter	Description	Value
d	Thickness of the electrolyte paper	$2.22 * 10^{-5} cm$
ω	Volume of ethyl glycol molecule	$5.66 * 10^{-9} cm^3$

Table 2. Parameter values of the degradation models

Capacitor	J_{open}	β	c
C_1	$0.2167 hr^{-1} cm^{-2}$	0.0041	0.0003
C_2	$0.2213\ hr^{-1} cm^{-2}$	0.0039	0.0003
C_3	$0.2183\ hr^{-1} cm^{-2}$	0.0029	0.0003
C_4	$0.1936\ hr^{-1} cm^{-2}$	0.0020	0.0003
C_5	$0.2232\ hr^{-1} cm^{-2}$	0.0029	0.0003
C_6	$0.2242\ hr^{-1} cm^{-2}$	0.0020	0.0003
C_7	$0.2177\ hr^{-1} cm^{-2}$	0.0042	0.0003
C_8	$0.2101\ hr^{-1} cm^{-2}$	0.0029	0.0003
C_9	$0.2076\ hr^{-1} cm^{-2}$	0.0033	0.0003
C_{10}	$0.2226\ hr^{-1} cm^{-2}$	0.0024	0.0003
C_{11}	$0.2186\ hr^{-1} cm^{-2}$	0.0028	0.0003
C_{12}	$0.2199\ hr^{-1} cm^{-2}$	0.0037	0.0003
C_{13}	$0.2211\ hr^{-1} cm^{-2}$	0.0038	0.0003
C_{14}	$0.2034\ hr^{-1} cm^{-2}$	0.0027	0.0003
C_{15}	$0.2180\ hr^{-1} cm^{-2}$	0.0026	0.0003

Table 3. Mean square errors for the capacitors models

Capacitor	mse
C_1	1.0e-08 * 0.0637
C_2	1.0e-08 *0.1069
C_3	1.0e-08 *0.0149
C_4	1.0e-08 *0.0379
C_5	1.0e-08 *0.0732
C_6	1.0e-08 *0.0482
C_7	1.0e-08 *0.0968
C_8	1.0e-08 *0.2286
C_9	1.0e-08 *0.0476
C_{10}	1.0e-08 * 0.0649
C_{11}	1.0e-08 *0.1447
C_{12}	1.0e-08 * 0.0258
C_{13}	1.0e-08 *0.1237
C_{14}	1.0e-08 *0.0204
C_{15}	1.0e-08 * 0.1958

from its degradation model at the same time. It should be mentioned here that in the prognostic procedure system states like J and α update during the process as the system measure the capacitance of the capacitor in each step. So it is expected that in the prognostic simulation the model demonstrates even better tracking performance.

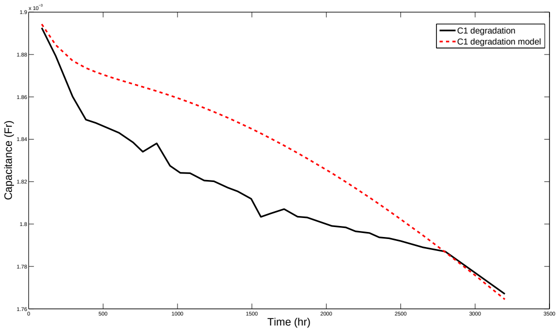


Figure 4. Degradation models.

5. Experimental Results

Using the proposed degradation model and the estimated variables, the remained useful life prediction for the capacitors are done according to the proposed methodology in section 3. The remaining useful life is considered when the capacitance of the capacitor decreases below 90% of its nominal value (Kulkarni et al., Sep 2012). Also in designing the particle filter 300000 sample points were considered. Figure 5 shows the actual remaining useful life and the estimated remaining useful life of the first capacitor from the beginning of the thermal overstress test to the end of its useful life. One can see from the this figure that by getting closer to the end of life of the capacitor the error of the prediction decreases and the predicted value converges to the real value of RUL. To explain this observation we can say as the system gets more data and the particle filter updates the degradation model, the degradation model becomes closer to the real degradation procedure and the predicted RUL based on the degradation model becomes more exact.

The RUL prediction relative accuracy (RA) is defined as:

$$RA = 100(1 - \frac{\mid RUL - RUL_e \mid}{RUL}) \tag{30}$$

where RUL is the remained useful life and RUL_e is the estimated remaining useful life. relative accuracy of the prediction of remaining useful life for each capacitor in each step time is presented in Table 3. The results show the prediction method is completely reliable and relative accuracy is improved in comparison with the previous works.

Table 4. Relative Accuracy

time(hr)	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
87.7	91.14	79.98	98.72	98.1	90.89	86.46	93.15	78.9	98.1	82.41	85.99	86.17	98.45	95.25	67.41
181.7	87.14	84.49	96.51	95.15	91.2	79.88	92.48	74.61	96.88	81.4	84.32	87.58	96.13	94.36	70.96
295.4	78.56	90.57	95.76	94.58	94.51	81.62	99.23	85.42	93.33	82.68	86.34	86.55	97.88	92.7	74.9
384.5	89.59	90.58	94.44	96.03	95.56	79.59	96.76	85.42	91.19	83.45	91.34	88.02	92.54	95.92	73.55
450.9	89.46	87.17	95.35	95.70	95.15	82.4	99.42	88.12	94.31	85.11	91.92	87.69	94.2	99.63	75.82
540.8	91.79	83.73	92.48	98.22	95.36	84.57	99.61	84.95	93.29	84.72	88.54	87.04	96.84	99.8	76.97
607.1	94.94	88.82	95.37	95.82	93.65	85.16	97.69	84.12	94.46	83.56	87.36	89.41	94.58	99.7	77.12
701.6	96.5	86.47	94.29	92.43	97.66	87.59	99.34	82.25	93.08	85.29	90.63	86.59	93.78	97.73	74.4
766.8	97.76	81.34	93.27	96.55	93.27	85.19	95.1	81.34	98.65	79.03	88.56	89.11	96.21	98.35	72.66
860.4	94.93	79.47	96.22	91.42	99.06	92.53	94.66	85.86	90.66	87.55	94.66	83.86	92.75	93.11	77.81
950.1	99.22	89.18	97.09	90.73	98.94	93.07	92.98	83.44	87.02	93.3	92.98	84.68	92.57	91.01	76.84
1019	94.49	95.01	97.1	87.82	96.38	94.11	91.37	85.04	86.83	94.16	91.28	83.24	92.64	91.56	71.56
1084.5	89.63	98.09	94.7	87.89	95.91	93.34	92.1	85.02	87.3	93.68	89.03	80.07	92.69	90.76	71.41
1179.5	86.15	89.25	95.8	87.05	98.76	92.28	92.52	81.8	86.69	93.87	88.7	79.27	94.67	93.54	73.36
1244.8	96.77	94.21	98.08	85.77	96.62	91.39	92.65	82.21	87.92	92.9	90.8	76.86	92.06	92	72.16
1338.2	94.96	97.24	97.22	82.98	93	90.33	92.45	82.48	87.16	91.78	87.94	74.74	92.73	92.39	73.24
1404.5	83.71	98.63	92.53	84.53	95.56	86.48	93.62	81.84	92.1	85.95	84.89	78.55	95.44	93.68	75.86
1495.4	75.2	92.83	92.79	76.24	88.31	93.07	86.67	91	82.64	93.81	94.78	68.98	84.53	82.58	86.43
1560.5	93.04	97.17	99.08	71.81	85.98	95.19	84.65	93.48	77.15	95.77	95.26	65.9	85.09	81.28	91.19
1626.5	86.49	91	90.16	79.74	89.58	93.01	90.39	93.69	79.81	93.28	90.52	67.07	87.83	83.72	89.1
1716.6	96.27	92	94.62	71.14	84.71	97.12	87.07	90.8	75.95	97.34	97.19	65.61	84.49	78.32	90.37
1807.0	86.95	96	93.79	76.18	86.19	89.94	88.83	93.47	77.61	98.29	95.43	67.3	86.79	82.88	87.98
1871.6	91.48	83.05	99.78	72.04	84.6	93.04	89.32	98.89	76.17	95.28	94.24	64.3	85.37	81.41	80.92
2036.9	78.98	82.07	99.89	79.16	89.64	87.98	94.79	93.13	82.81	95.75	91.07	68.31	92.82	82.06	82.56
2131.3	86.56	84.77	97.01	77.79	88.35	91.02	90.04	93.71	79.98	98.39	93.51	64.94	87.35	75	82.55
2196.1	87.15	95.84	92.53	77.04	92.21	97.15	95.39	87.45	87.59	92.4	84.38	67.92	96.49	81.98	73.49
2290.1	97.75	85.9	96.9	72.62	94.42	92.17	98.69	90.64	82.22	91.35	85.31	69.42	93	81.63	81.63
2356	88.42	79.3	79	74.92	90.73	80.19	98.96	93.56	82.51	95.87	99.21	65.41	88.42	65.28	87.28
2421.9	95	63	89.81	72.42	88.42	77.28	98.28	86.71	78.28	88.42	87.14	51.57	80.28	60.28	92.57
2500	94.55	61	84.75	75.27	90.72	74.72	96.54	83.81	78.36	91.09	82.72	44.9	81.63	59.81	99.09
2650	97.5	69.25	59.5	81.5	83.5	67.75	93.5	85	68.5	88.75	78.25	60.5	82.5	48.75	96.75
2800	100	97	100	64.5	86	74.5	84	80	87.5	88	76.5	86	84	54	93

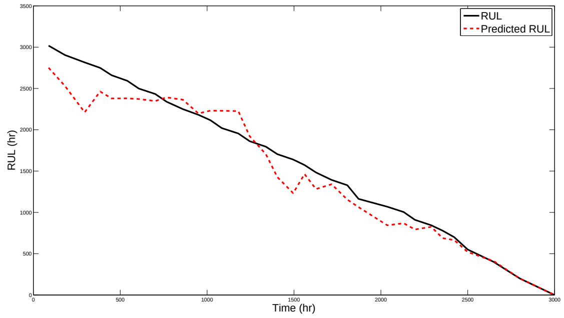


Figure 5. RUL Estimation.

6. CONCLUSION

In this paper a general model with three state variables is considered to represent the degradation process of the electrolytic capacitors. There is no doubt that degradation process is a nonlinear complex phenomenon. To perform prognostics with high performance and acceptable range of error we had to take in to account these nonlinearities in the degradation model. The results of experimental study shows acceptable performance in relative accuracy of RUL predictions. However, by adding these complexities to the model the number of parameters we have to estimate for the model increased as well. This can be a potential problem in designing a remaining useful life predictor model. In the estimation of each capacitor's model parameters we did not use that capacitor's data. In fact we considered 14 other capacitors to estimate its parameters. So we can claim that derived values for the system parameters can be used for any similar capacitor and we expect to get acceptable results as well. One of the advantages of this work is as the system comes close to the failure, the performance of RUL prediction improves. The reason of this observation is discussed in the paper but it has an important practical value. In fact from the safety and mission critical point of view it is much more important for the operators to have an exact estimation of RUL when we are close to the end of life or failure.

In this paper degradation of the electrolytic capacitors under thermal over stress condition is studied. However, the suggested degradation model and prognostic algorithm can be applied for RUL prediction of a capacitor under electrical overstress condition as well. In fact, in electrical overstress condition because of additional chemical reactions in the capacitors (Gmez-Aleixandre et al., 1986), capacitor popping is more likely, making this model more relevant. In future work, we will apply the suggested degradation model and prognostic method for RUL prediction of electrolytic capacitors under electrical overstress condition.

References

Arulampalam, M. S., Maskell, S., Gordon, N., Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. Signal Processing, IEEE Transactions on, 50(2), 174-188.

Celaya, J. R., Wysocki, P., Vashchenko, V., Saha, S., Goebel, K. (2010, September). Accelerated aging system for prognostics of power semiconductor devices. In AUTOTESTCON, 2010 IEEE (pp. 1-6). IEEE.

Celaya, J. R., Kulkarni, C. S., Biswas, G., Goebel, K. (2012). Towards A Model-based Prognostics Methodology for Electrolytic Capacitors: A Case Study Based on Electrical Overstress Accelerated Aging. International Journal of Prognostics and Health Management.

Gmez-Aleixandre, C., Albella, J. M., Martnez-Duart, J. M. (1986). Pressure build-up in aluminium electrolytic capacitors under stressed voltage conditions. Journal of applied electrochemistry, 16(1), 109-115.

Goebel, K., Saha, B., Saxena, A., Celaya, J., Christophersen, J. (2008). Prognostics in battery health management. Instrumentation and Measurement Magazine, IEEE, 11(4), 33-40.

Goodman, D. L., Vermeire, B., Spuhler, P., Venkatramani, H. (2005, March). Practical application of PHM/prognostics to COTS power converters. In Aerospace Conference, 2005 IEEE (pp. 3573-3578). IEEE.

Gordon, N. J., Salmond, D. J., Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In IEE Proceedings F (Radar and Signal Processing) (Vol. 140, No. 2, pp. 107-113). IET Digital Library.

Julier, S. J., Uhlmann, J. K. (1997, July). New extension of the Kalman filter to nonlinear systems. In AeroSense'97 (pp. 182-193). International Society for Optics and Photonics.

Kulkarni, C., Biswas, G., Koutsoukos, X. (2009, October). A prognosis case study for electrolytic capacitor degradation in DC-DC converters. In PHM Conference.

Kulkarni, C. S., Celaya, J. R., Biswas, G., Goebel, K. (2012, June). Physics based Modeling and Prognostics of Electrolytic Capacitors. Aerospace 2012, 19-21 June, Garden Grove, California.

Kulkarni, C. S., Celaya, J. R., Goebel, K., Biswas, G. (2012, September). Bayesian Framework Approach for Prognostic Studies in Electrolytic Capacitor under Thermal Overstress Conditions.

Kulkarni, C. S. (2013). A Physics-based Degradation Modeling Framework for Diagnostic and Prognostic Studies in Electrolytic Capacitors (Doctoral dissertation, Vanderbilt University).

Rdner, S. C., Wedin, P., Bergstrm, L. (2002). Effect of electrolyte and evaporation rate on the structural features of dried silica monolayer films. Langmuir, 18(24), 9327-9333.

Ristic, B., Arulampalm, S., Gordon, N. J. (2004). Beyond the Kalman filter: Particle filters for tracking applications. Artech House Publishers.

Tasca, D. M. (1981, September). Pulse power response and damage characteristics of capacitors. In EOS/ESD Symposium Proceedings. Las Vegas: ESD Assn, Rome, NY (pp. 174-91).

Wan, E. A., Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000 (pp. 153-158). IEEE.

Hamed Khorasgani is a Ph.D candidate at ISIS, Vanderbilt University, Nashville, TN. He received the M.S.c degree in Mechatronics Engineering, in 2012 from Amirkabir University of Technology, Iran and a B. Sc. in Electronics and Electrical Engineering in 2009 from Isfahan University of Technology, Iran.

Chetan S. Kulkarni received the B.E. (Bachelor of Engineering) degree in Electronics and Electrical Engineering from University of Pune, India in 2002 and the M.S. and Ph.D. degrees in Electrical Engineering from Vanderbilt University, Nashville, TN, in 2009 and 2013, respectively. In 2002 he joined Honeywell Automation India Limited (HAIL) as a Project Engineer. From May 2006 to August 2007 he was a Research Fellow at the Indian Institute of Technology (IIT) Bombay with the Department of Electrical Engineering. From Aug 2007 to Dec 2012, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. Since Jan 2013 he is

Research Engineer II with SGT Inc. at the Prognostics Center of Excellence, NASA Ames Research Center. His current research interests include physics-based modeling, model-based diagnosis and prognosis focused towards electrical and electronic devices and systems. Dr. Kulkarni is a member of the Prognostics and Health Management (PHM) Society, AIAA and the IEEE.

Gautam Biswas received the Ph.D. degree in computer science from Michigan State University, East Lansing. He is a Professor of Computer Science and Computer Engineering in the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. His primary research interests are in modeling and simulation of complex, hybrid systems, fault diagnosis and prognostics, and fault-adaptive control. His research has been supported by DARPA, NASA, and NSF. He has over 400 refereed publications.

José R. Celaya is a research scientist with SGT Inc. at the Prognostics Center of Excellence, NASA Ames Research Center. He received a Ph.D. degree in Decision Sciences and Engineering Systems in 2008, a M. E. degree in Operations Research and Statistics in 2008, a M. S. degree in Electrical Engineering in 2003, all from Rensselaer Polytechnic Institute, Troy New York; and a B. S. in Cybernetics Engineering in 2001 from CETYS University, México.

Kai Goebel received the degree of Diplom-Ingenieur from the Technische Universitt Mnchen, Germany in 1990. He received the M.S. and Ph.D. from the University of California at Berkeley in 1993 and 1996, respectively. Dr. Goebel is a senior scientist at NASA Ames Research Center where he leads the Diagnostics and Prognostics groups in the Intelligent Systems division. In addition, he directs the Prognostics Center of Excellence and he is the technical lead for Prognostics and Decision Making of NASAs System-wide Safety and Assurance Technologies Program. He worked at General Electrics Corporate Research Center in Niskayuna, NY from 1997 to 2006 as a senior research scientist. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion. His research interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He holds 15 patents and has published more than 200 papers in the area of systems health management.

A Framework to Debug Diagnostic Matrices

Anuradha Kodali[1], Peter Robinson[2], and Ann Patterson-Hine[2]

[1]*SGT Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA*
anuradha.kodali@nasa.gov

[2]*NASA Ames Research Center, Moffett Field, CA, 94035, USA*
peter.i.robinson@nasa.gov
ann.patterson-hine@nasa.gov

ABSTRACT

Diagnostics is an important concept in system health and monitoring of space operations. Many of the existing diagnostic algorithms utilize system knowledge in the form of diagnostic matrix (D-matrix, also popularly known as diagnostic dictionary, fault signature matrix or reachability matrix). The D-matrix maps tests on observed conditions to failures. This matrix is mostly gleaned from physical models during system development. But, sometimes, this may not be enough to obtain high diagnostic performance during operation due to system modifications and lag and noise in sensor measurements. In such a case, it is important to modify this D-matrix based on knowledge obtained from sources such as time-series data stream (simulated or maintenance data) within a framework that includes the diagnostic/inference algorithm. A systematic and sequential update procedure, diagnostic modeling evaluator (DME) is proposed to modify D-matrix and wrapper/test logic considering the least expensive update first. The user sets the diagnostic performance criteria. This iterative procedure includes conditions ranging from modifying 0's and 1's in the matrix, adding/removing the rows (failure sources)/columns (tests), or modifying test/wrapper logic used to determine test results. We will experiment this framework on ADAPT datasets from DX challenge 2009.

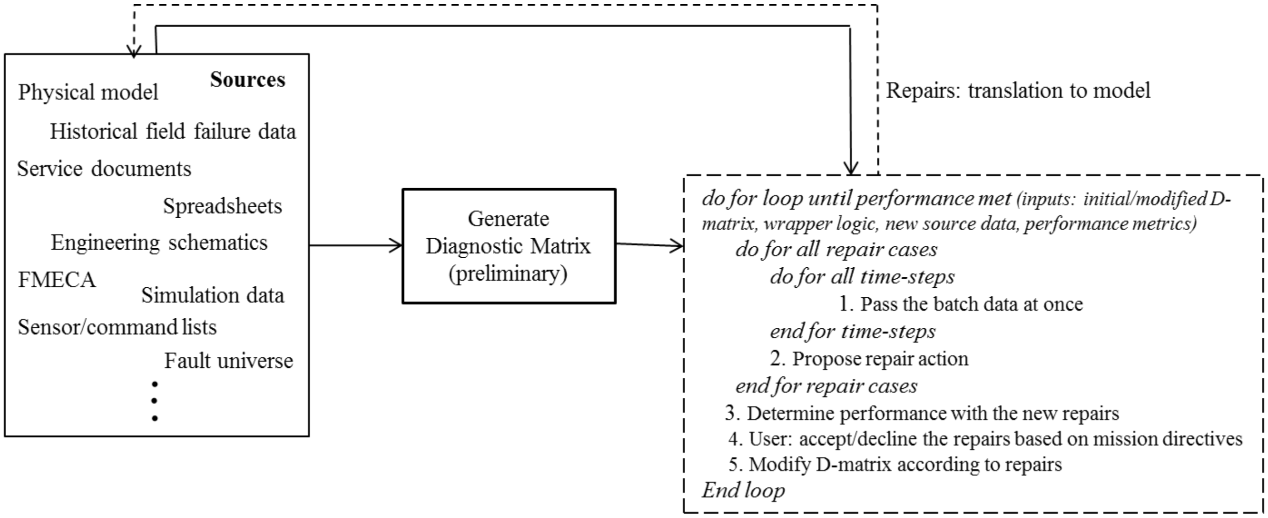
1. INTRODUCTION

Traditionally, diagnostics is performed in the following way: System modeling → List failure causes (faults) → Design tests → Generate D-matrix → diagnosis via inference algorithm (Luo & Pattipati, 2007). Here, the process from system modeling to generate D-matrix is independent of the diagnoser. But, when the diagnostic algorithm based on D-matrix (Singh, Kodali, Choi, Pattipati, Namburu, Chigusa, Prokhorov, & Qiao, 2009) is applied during operations, and the performance is not robust, it is important to reexamine the system model (D-matrix) in terms of its correctness and utility towards diagnosability. Thus, we propose a debugging architecture, termed diagnostic modeling evaluator (DME) that includes the diagnoser and repairs the system model (D-matrix) to suit better diagnostic performance based on new/updated information. This updated information is mostly available after system development or during operation.

D-matrix can be developed from physical models, historical field failure data, service documents, engineering schematics, and Failure Modes, Effects and Criticality Analysis (FMECA) data (Singh, Holland, & Bandyopadhyay, 2011) by establishing causal relationship between faults and tests (Luo, Tu, Pattipati, Qiao, & Chigusa, 2006). Initially, D-matrix is generated from any of these sources (e.g., physical model). The initial model, when developed during system development, ignores lag and noise in sensor measurements during operation and other system advancements during deployment. Then other sources (e.g., operations data (time-series)) that contain these critical changes can be used as reference material in DME framework to repair the initial D-matrix. This provides a debugging environment to the initial model. This also provides an effective platform to represent information from different sources (model-based, data-driven, or knowledge-based) in a unified D-matrix concept.

Diagnostic modeling evaluator (DME) is developed as an automated debugging process to update/repair D-matrix that best suits user-defined performance requirements. This includes assessing the level of fault definitions (component or failure mode level), number of tests required, test logic by considering the thresholds for faulty behavior, and most importantly the fault-test relationships. Conditions (repairs) ranging from modifying 0's and 1's in the matrix, or modifying the rows to accommodate lower-level fault modeling with failure modes, or adding or removing tests, or changing their test logic are identified to experiment for

Anuradha Kodali et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.



```
do for loop until performance met (inputs: initial/modified D-matrix, wrapper logic, new source data, performance metrics)
    do for all repair cases
        do for all time-steps
            1. Pass the batch data at once
        end for time-steps
        2. Propose repair action
    end for repair cases
    3. Determine performance with the new repairs
    4. User: accept/decline the repairs based on mission directives
    5. Modify D-matrix according to repairs
End loop
```

Figure 1. Framework to debug D-matrix

better performance in terms of diagnostics (detection + isolation). This is implemented as an iterative feedback process by tuning D-matrix at every step with repair conditions. Sometimes, more than one repair is applicable on a given iteration. Those repairs are accepted/declined by the user/decision maker based upon their diagnostic performance and also, most importantly, mission directives. The user-defined performance criteria are quantified based on the following metrics: diagnostic efficiency, false positive/false negative rate, diagnostic time and cost. This is again communicated to update procedure and the iterations go on until there are no further changes (shown in Figure 1).

In this paper, section 2 focuses on explaining DME procedure as a debugger and the conditions required to update D-matrix. In section 3, the process of updating D matrix is shown with examples. Two example systems, rover and ADAPT (from DX challenge 2009) are included in this paper.

2. Debugging Framework

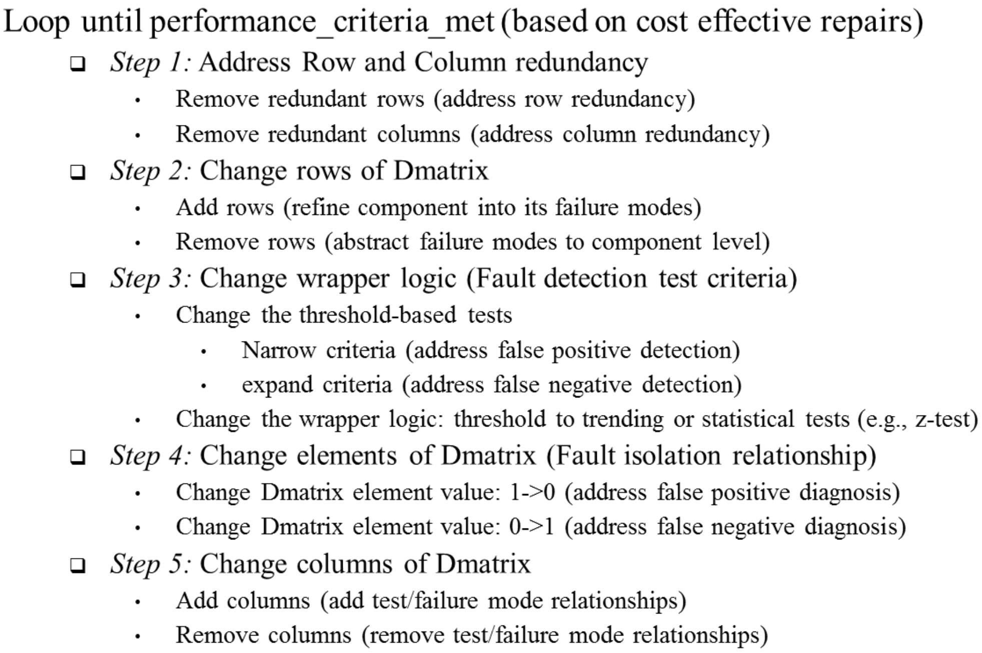
Conventionally, system model doesn't consider diagnostic utility when developing D-matrix. While developing a robust diagnostic system, it is important for both system modeling and diagnostic process to interact coherently resulting in high detection and isolability performance during operation. To make this idea possible, DME acts as a debugger to the initial D-matrix using the available operational or simulations data. It plays the data in batch mode in order to determine which repairs to make. No single time step decisions are made, though this would be required to utilize these techniques during runtime operation (other data sources can also be used to repair D-matrix). These repairs to D-matrix can be translated back to the initial system model. This is pursued as future work. Here, we will explain the modules and process of DME framework as shown in Figure 1:

2.1. Information Sources

The diagnostic modeling, firstly, starts with building the model from the system information, viz., physical model, historical field failure data, service documents, spreadsheets, engineering schematics, FMECA, sensor/commands list, and simulations data (Singh et al., 2011). D-matrix, in DME context, is built from one source initially and then updated/repaired with the other available sources. These sources can be model-based, data-driven, or knowledge-based informative sources. This repair process can be performed during system development or operational phase. The data collected during real-time operations accommodates lags and noise ignored during system development, thus providing operational information about the relationship between faults and tests. But, sometimes, the information from different sources can be counter-explanative and this needs to be dealt carefully during corrective actions to the D-matrix.

2.2. Generating D-matrix (preliminary)

Here, the initial D-matrix is generated from any of the available sources listed above. For this purpose, it is important to determine the level at which fault modeling is performed for diagnosis. This can be done at sub-system, or component, or failure mode level depending on the system properties and requirements. Also, the testing criterion for each test is formulated based on sensor measurements. These tests can either be threshold, trending, or statistical tests. Subsequently, D-matrix is generated either by hand or by automated software methods like TEAMS Designer (Qualtech Systems Inc.). The D-matrix generated in this step is preliminary and is modified in the next step.



Loop until performance_criteria_met (based on cost effective repairs)

- ❑ *Step 1:* Address Row and Column redundancy
  - Remove redundant rows (address row redundancy)
  - Remove redundant columns (address column redundancy)
- ❑ *Step 2:* Change rows of Dmatrix
  - Add rows (refine component into its failure modes)
  - Remove rows (abstract failure modes to component level)
- ❑ *Step 3:* Change wrapper logic (Fault detection test criteria)
  - Change the threshold-based tests
    - Narrow criteria (address false positive detection)
    - expand criteria (address false negative detection)
  - Change the wrapper logic: threshold to trending or statistical tests (e.g., z-test)
- ❑ *Step 4:* Change elements of Dmatrix (Fault isolation relationship)
  - Change Dmatrix element value: 1->0 (address false positive diagnosis)
  - Change Dmatrix element value: 0->1 (address false negative diagnosis)
- ❑ *Step 5:* Change columns of Dmatrix
  - Add columns (add test/failure mode relationships)
  - Remove columns (remove test/failure mode relationships)

Figure 2. Top-level iterative loop identifying progression of repairs for DME framework

2.3. Diagnostic Modeling Evaluator (DME)

In the process of repairing D-matrix for better performance, an iterative loop (DME) consists of a sequential procedure with low-cost repairs considered first (as shown in Figure 2). The updates in the D-matrix are made in accordance to the new information from another source. After each iteration, the performance of the updated matrix is determined and the changes are accepted/declined accordingly based on mission directives. Here, the performance can be defined as combination of the required metrics, viz., false-positive, or false-negative rate, accuracy, time to diagnose, or cost involved for test and diagnoser implementation. Sometimes, improvement in one metric can affect others. Thus, balance should be maintained given the mission directives. This process is continued until there can be no further improvement. The necessary steps involved in the iterative updating DME procedure of the D-matrix are listed below:

2.3.1. Repair Cases for Updating D-matrix

1. Address row/column redundancy

Faults/test corresponding to rows and columns of D-matrix are assessed for redundancy in terms of two or more rows or columns having exactly the same signature. Duplicate rows or columns can result in ambiguous/masking faults and bad/duplicate tests, respectively (Simpson & Sheppard, 1992). In such a case, to decrease computational complexity and simplify representation, those faults/tests are grouped in to one. It is better to keep track of this change to avoid ignoring the subsequent repairs and also when they are required for mission critical functions or in other system mode.

2. Modify Fault Modeling (Change Rows of D-matrix)

In general, faults are modeled at component level. But, sometimes, components can be faulty with different severity levels based on their root-cause resulting in different fault signatures in D-matrix. In this case, new rows are added to the D-matrix when failure modes need to be refined. We add a row for each addition of a new component. Similarly, when the diagnosis is required at higher level (e.g Line Replaceable Unit (LRU) or Orbital Replaceable Unit (ORU)) or when components are removed, we remove the corresponding rows in D-matrix.

Table 1. List of tests[1]

Tests	Symbols
Voltage sensors	V1, V2, V3, V4
Battery temp. sensor	TB1, TB2, TB3, TB4
Battery current sensor	i
Position sensors	xFL, xFR, xBL, xBR
Velocity sensors	wFL, wFR, wBL, wBR
Current sensors	iFL, iFR, iBL, iBR
Temperature sensors	TFL, TFR, TBL, TBR

[1] FL, FR, BL, BR represents front left, front right, back left, back right wheels, respectively.

Table 2. D-matrix and the corresponding delays (in seconds) of each test for the rover system

	V1	TB1	V2	TB2	V3	TB3	V4	TB4	I	xBL	wBL	iBL	TBL
Motor Friction Fault BL	1/0.55	1/25.75	1/0.55	1/22.6	1/0.55	1/19.55	1/0.55	1/23.6	1/0.4	1/0.9	1/0.2	1/0.4	1/30.95
Parasitic Load	1/0.2	1/28.8	1/0.2	1/26.85	1/0.2	1/26.1	1/0.2	1/31.9	1/0.15	0	0	0	0
Voltage Sensor 1 Bias	1/0.15	0	0	0	0	0	0	0	0	0	0	0	0

3. Change Test Logic/Wrapper Code

We modify the test logic to attain better detection for each test. The test criteria, especially when defined by thresholds, may not hold during operation or with degradation of component's performance over time. This necessitates changing the logic subsequently either to trending or statistical tests. Additionally, refinement of abstraction of failure modes may in turn require test logic/wrapper code to be refined as well. Changes in test logic should be properly monitored, sometimes, for increased false positive or false negative detection rate with respect to the user defined expected performance measures for the D-matrix.

4. Repair False positives/negatives (Change the Entries of D-matrix)

The most important correction to the D-matrix is updating its entries, i.e., changing 1 to 0 or vice-versa, thus decreasing false positive or false negative isolation rate, correspondingly. This can be reflected as system (physical model) change. This means that a 1 in the D-matrix means two conditions are true: that a path exists between the fault and the test and that a set of signals propagate from the failure are detected at the test. A change in fault test relationships means that change for the paths and signals are applied. Note that both false positives and false negative isolation rates cannot be improved concurrently. This is because, to improve false negative isolation rate, we need to expand the threshold logic (red lines) which will increase false positives. Thus, the required acceptable metric is obtained from the mission directives.

5. Add/remove Tests (Change Columns of D-matrix)

Adding tests incurs additional cost, so we restrict this repair strategy to be done at last. We have to design new tests if some faults are not adequately detected or if they are not isolatable. Tests also can be broken into finer levels, similar to the component to failure mode representation, to be able to detect different fault modes with different severities. On the other hand, sometimes, low reliable and delayed tests hinder the overall diagnostic performance efficiency. Such tests when not detecting critical faults in any other system mode can be removed.

2.3.2. Diagnostic Algorithm

Here, any standard diagnostic procedure based on D-matrix can be applied as a diagnoser. We have applied diagnostic algorithms Dynamic Multiple Fault Diagnosis (DMFD) algorithms based on primal-dual optimization framework that can detect multiple, delay, and intermittent faults over time. Our problem is to determine the time evolution of fault states based on imperfect/perfect test outcomes observed over time and is formulated as one of finding the maximum *a posteriori* configuration to evaluate fault state evolution over time (which is why the time series is process in batch mode) that best explains the observed test outcome sequence. More details of these algorithms can be found in (Singh et al., 2009), (Kodali, Singh, & Pattipati, 2013), (Kodali, Pattipati, & Singh, 2013).

2.3.3. Performance Evaluation

Performance metrics can be overall diagnostic efficiency, false-positive rate, false-negative rate, diagnostic time and cost. The choice of metrics is dependent on the user-set criteria based on mission directives. As an example, user can determine to have less false positives (this may increase false negatives).

3. Simulations and Results

We demonstrate DME framework on two example systems, viz., rover and ADAPT systems. We do not provide the description for these systems. More details can be found in (Narasimhan, Balaban, Daigle, Roychoudhury, Sweet, Celaya, & Goebel, 2012) and (Poll, Patterson-Hine, Camisa, Garcia, Hall, Lee, Mengshoel, Neukom, Nishikawa, Ossenfort, Sweet, Yentus, Roychoudhury, Daigle, Biswas & Koutsoukos, 2007) for rover and ADAPT systems, respectively.

3.1. Example System 1: Rover

The initial D-matrix of the rover system is generated via simulations with three faults and thirteen tests (see Table 1 and Table 2). Three fault scenarios viz., battery parasitic load, motor friction fault, and voltage sensor fault are simulated (Narasimhan et al., 2012) by injecting them in the rover test bed and altering the corresponding measurements

Table 4. List of faults in ADAPT-Lite

No.	Fault ID	Fault modes	Test ID	Sensor description
1	ISH266	Stuck	E235	DC voltage
2	TE228	Stuck, offset	E240	DC voltage
3	IT267	Stuck, offset	E242	DC voltage
4	E267	Stuck, offset	E261	DC voltage
5	IT240	Stuck, offset	E265	AC voltage
6	ESH260A	Stuck	E267	AC voltage
7	E242	Stuck, offset	ESH244A	Actuator position
8	ESH275	Stuck	ESH260A	Actuator position
9	IT261	Stuck, offset	ESH275	Actuator position
10	E261	Stuck, offset	ISH236	Actuator position
11	E240	Stuck, offset	ISH262	Actuator position
12	E235	Stuck, offset	ISH266	Actuator position
13	E265	Stuck, offset	IT240	DC current transmitter (50A Max)
14	TE229	Stuck, offset	IT261	DC current transmitter (50A Max)
15	ISH262	Stuck	IT267	AC current transmitter (12A Max)
16	ST516	Stuck, Offset	ST265	AC frequency transmitter
17, 18, 19	FAN416	FailedOff, under speed, over speed	ST516	Speed (RPM) transmitter
20	EY275	Stuckopen, stuckclosed	TE228	Temperature
21	CB266	Tripped, failedopen	TE229	Temperature
22	INV2	FailedOff	XT267	Phase angle transducer
23	CB262	Tripped, failedopen		
24	EY260	Stuckopen, stuckclosed		
25	EY244	Stuckopen, stuckclosed		
26	CB236	Tripped, failedopen, stuckclosed		
27	ISH236	Stuck		
28	XT267	Stuck, offset		
29	ST265	Stuck, offset		
30	ESH244A	Stuck		

Table 3. Diagnosis time for each fault (fault injected at 50s)

	HyDe	QED	D-matrix diagnoser
Motor Friction Fault BL	50.2s	50.25s	50.2s
Parasitic Load	50.05s	51.2s	50.05s
Voltage Sensor 1 Bias	50.1s	50.3s	50.15s

with erroneous values. All these faults are injected at 50s. Temperature sensors have high detection delays (see Table 2), therefore, the corresponding diagnostic delays will also be longer. The DMFD algorithm with delays (Kodali et al., 2013) delivers intermediate diagnosis at each time-step with partial test information and updates it as the test information becomes available.

The inputs to DME framework are initial D-matrix and delay metrics in Table 2. Then, at iteration 1, the time to diagnosis is high due to detection delays of temperature sensors. There can be 2 repair actions for this case. Either, the test logic can be changed for the temperature sensors or they can be removed. But, changing test logic cannot avoid detection delays. Thus, the tests relating to temperature sensors are removed. This improves the time to diagnosis with out compromising the isolability for the listed fault universe[2]. The results with updated D-matrix are comparable with other diagnostic algorithms (HyDe, QED (Narasimhan et al., 2012) in Table 3 (the faults are injected at 50s).

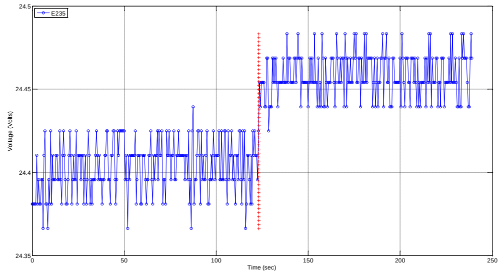


Figure 3. E235 sensor measurement when fault is injected

3.2. Example System 2: ADAPT-Lite

Here, we used the dataset generated from ADAPT-Lite system for DX workshop tier 1 competition 2009 (Kurtoglu, Narasimhan, Poll, Garcia, Kuhn, de Kleer, van Gemund, &

[2] This strategy may not hold if we expand the fault universe. Here, this is demonstrated as an example.

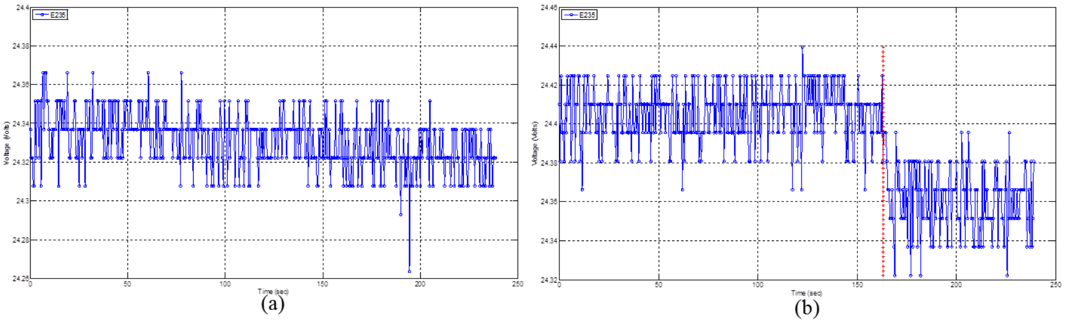
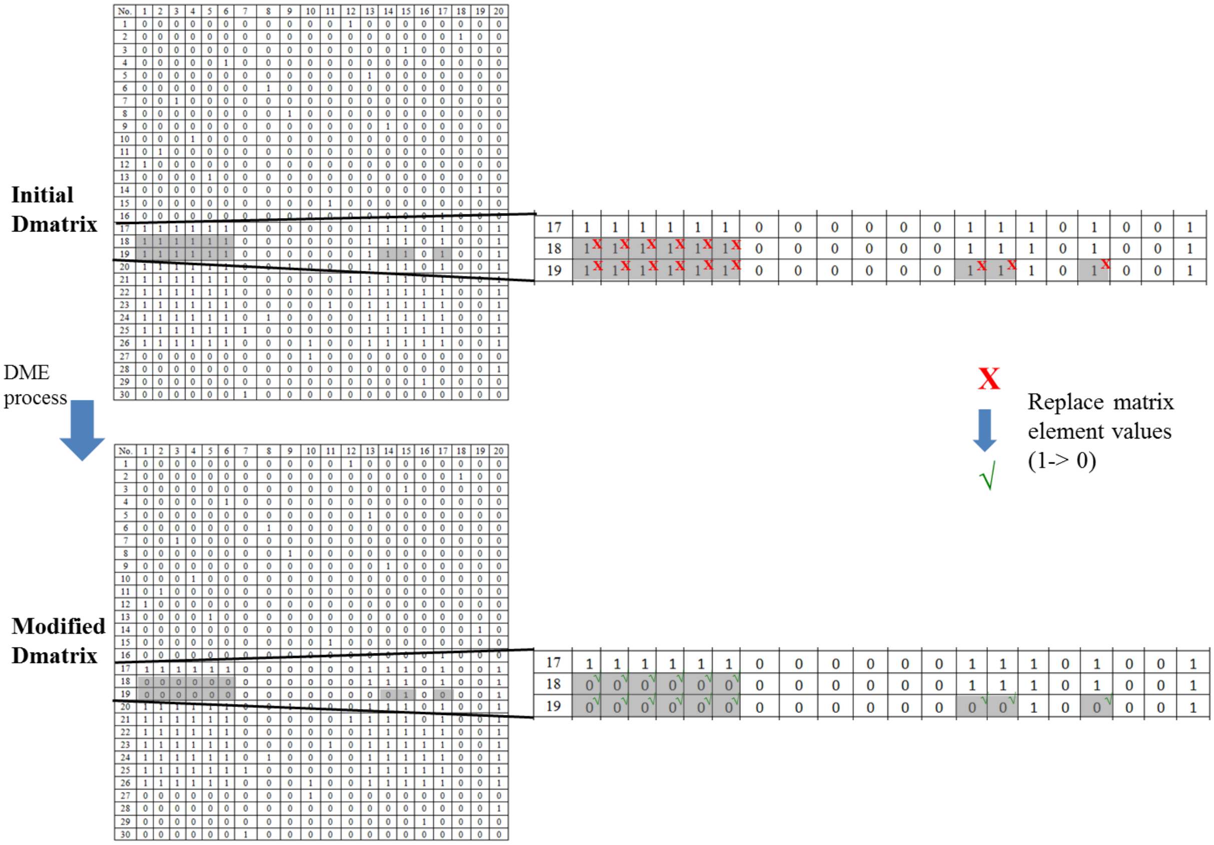


Figure 4. E235 sensor measurement: (a) nominal case (b) when fault 19 is injected (red line indicates the fault injection time)



No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
17	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0	1
18	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0	1
19	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0	1
20	1	1	1	1	1	1	0	0	1	0	0	0	1	1	1	0	1	0	0	1
21	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
22	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	0	1
23	1	1	1	1	1	1	0	0	0	0	1	0	1	1	1	1	1	0	0	1
24	1	1	1	1	1	1	0	1	0	0	0	0	1	1	1	1	1	0	0	1
25	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	1
26	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	1
27	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
30	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

17	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0	1
18	1X	1X	1X	1X	1X	1X	0	0	0	0	0	0	1	1	1	0	1	0	0	1
19	1X	1X	1X	1X	1X	1X	0	0	0	0	0	0	1X	1X	1	0	1X	0	0	1

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
17	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0	1
18	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	1
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
20	1	1	1	1	1	1	0	0	1	0	0	0	1	1	1	0	1	0	0	1
21	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
22	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	0	1
23	1	1	1	1	1	1	0	0	0	0	1	0	1	1	1	1	1	0	0	1
24	1	1	1	1	1	1	0	1	0	0	0	0	1	1	1	1	1	0	0	1
25	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	1
26	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	1
27	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
30	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

17	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0	1
18	0√	0√	0√	0√	0√	0√	0	0	0	0	0	0	1	1	1	0	1	0	0	1
19	0√	0√	0√	0√	0√	0√	0	0	0	0	0	0	0√	0√	1	0	0√	0	0	1

Figure 5. DME process to generate modified D-matrix from initial D-matrix (ADAPT system)

Feldman, 2009). This dataset has 2 parts: sample and competition data. Data (sensor measurements) is collected for 238.5s with a sampling rate of 0.5s, thus the data contains a total of 478 time steps for both nominal case and when faults are injected. Each fault is injected and the corresponding continuous sensor measurements over time are noted (see Table 4 for list of faults and tests). Initially, D-matrix is generated by visualizing the sensor measurements in sample data. As seen in Figure 3, there is a clear change in the mean value of the sensor E235 when fault E235 is injected. Then, the corresponding row-column entry in D-matrix is depicted as 1 in the D-matrix. The corresponding test logic in terms of threshold logic is also generated to suit fault detection.

In DME framework, at iteration 1 using sample data, rows 17-19 corresponding to different fault modes, viz., failed off, under speed, and over speed for the component FAN have similar fault signature. These are identified as ambiguous rows. Also, when faults 18 and 19 occur, they are misdiagnosed as fault 3 (IT 267 sensor fault). This is identified as either incorrect test logic or D-matrix entries in DME framework. But, most of the tests connected to faults

Table 5. Competition data results

(in %)	Initial D-matrix	Modified entries of D-matrix
False positive rate	4.85	6.94
False negative rate	2.39	2.41
Classification accuracy	94.57	96.15

18 and 19 didn't fail. Thus, the former repair is rejected. Therefore, the D-matrix entries are changed to generate new fault signatures for faults 18 and 19 (shown in Figure 5). The entries corresponding to voltage and current sensors are changed to 0. This is because, the effect of these faults on the corresponding sensor values is very low, i.e., the change in the measurement values is minimal. This is evident in Figure 4. Even though there is a clear indication of shift in the measurement value when fault is injected in Figure 4(b), those faulty values are overlapping with the values in the nominal case (24.3-24.36V). The repaired D-matrix is shown in Figure 5. These D-matrix repairs are verified on competition data using DMFD algorithm (Singh et al., 2009) (see Table 5). Classification accuracy is the percentage number of events that are correctly diagnosed (both nominal and faulty cases over 478 time-steps). Evidently, the diagnostic performance with modified D-matrix is better; thus, the corresponding repair strategy is accepted. Note that, the false positive and false negative rates are increased with modified D-matrix. But, here, the classification accuracy is considered as the decisive performance metric. There are no further changes in subsequent iterations.

4. Summary and Future Directions

Traditionally, diagnostics is viewed as an open-loop cascading process with the D-matrix as the input to the inference algorithm. In this context, DME is proposed to allow feedback from the diagnoser to the initial model via D-matrix repairs. Here, most importantly, the D-matrix can be updated to account for noise, lag and other effects by validating it through a time-series data stream or any other information that comes along during or after system development and operational deployment. Thus, in this iterative process, DME updates D-matrix and the corresponding test logic through a sequential procedure in the order of cost-effective repairs using the time-series data stream.

In our future research, DME framework will be updated and implemented as an automatic process. We will also experiment with systems that can accommodate the other repair strategies. We will also define user's role to validate the repair recommendation based on performance and mission directives and experiment with different metrics as the performance criteria. Most importantly, we will provide user-computer interface to communicate repair actions and provide necessary feedback. A systematic process to transverse the repairs back to the system model will also be investigated.

References

Luo, J., & Pattipati, K. (2007). An integrated diagnostic development process for automotive engine control systems. *IEEE Trans. Syst., Man, Cybern. C*, vol. 37, no. 6, pp. 1163–1173, Nov. 2007.

Singh, S., Kodali, A., Choi, K., Pattipati, K., Namburu, S., Chigusa, S., Prokhorov, D.V., & Qiao, L. (2009). Dynamic multiple fault diagnosis: Mathematical formulations and solution techniques. *IEEE Trans. Syst., Man, Cybern*. A, vol. 39, no. 1, pp. 160–176.

Singh, S., Holland, S., & Bandyopadhyay, P. (2011). Trends in the development of system-level fault dependency matrices. *IEEE Aerospace Conference*, Big Sky, Montana.

Luo, J., Tu, H., Pattipati, K., Qiao, L., & Chigusa, S. (2006). Graphical models for diagnostic knowledge representation and inference. *IEEE Instrum. Meas. Mag.*, vol. 9, no. 4, pp. 45–52.

Qualtech Systems Inc., www.teamqsi.com.

Simpson, W., & Sheppard, J. (1992). System Testability Assessment for Integrated Diagnostics. *IEEE Des. Test. Comput.*, vol. 9, no. 1, pp. 40-54.

Kodali, A., Singh, S., & Pattipati, K. (2013). Dynamic set-covering for real-time multiple fault diagnosis with delayed test outcomes. *IEEE Trans. Syst., Man, Cyben. A*, vol. 43, no. 3, pp. 547-562.

Kodali, A., Pattipati, K., & Singh, S. (2013). Coupled factorial hidden Markov models (CFHMM) for diagnosing multiple and coupled faults. *IEEE Trans. Syst., Man, Cyben. A*, vol. 43, no. 3, pp. 522-534.

Narasimhan, S., Balaban, E., Daigle, M., Roychoudhury, I., Sweet, A., Celaya, J., & Goebel, K. (2012) Autonomous decision making for planetary rovers using diagnostic and prognostic information. *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS 2012)*, Mexico City, Mexico.

Poll, S., Patterson-Hine, A., Camisa, J., Garcia, D., Hall, D., Lee, C., Mengshoel, O., Neukom, C., Nishikawa, D., Ossenfort, J., Sweet, A., Yentus, S., Roychoudhury, I., Daigle, M., Biswas, G., & Koutsoukos, X. (2007). Advanced diagnostics and prognostics testbed. *In Proc. DX'07*, pp. 178–185.

Kurtoglu, T., Narasimhan, S., Poll, S., A., Garcia, D., Kuhn, L., de Kleer, J., van Gemund, A., & Feldman, A. (2009). First international diagnostics competition – DXC'09. *In Proc. DX'09*.

Prognostic Algorithm Verification

N. Scott Clements and David S. Bodden

Lockheed Martin Aeronautics Company, Fort Worth, TX, 76131, USA
scott.clements@lmco.com
david.s.bodden@lmco.com

ABSTRACT

A rigorous methodology is presented for both specification and verification of prognostic algorithm performance. The prognostic algorithm specification statement takes the form, "The prognostic algorithm shall provide a minimum of <TTM> hours time-to-maintenance such that between <Lower>% and <Upper>% of failures of component ABC will be avoided with <Confidence>% confidence." The methodology is developed first for a single failure mode case and then extended to the multiple failure mode case. The case of non-prognosable failure modes is also considered. Finally, implications of this approach are presented, including pre-tabulation of confidence bounds, estimation of the minimum amount of data required to reach a given verification confidence, and a method for using a minimum confidence growth curve to account for initial low confidence in a prognostic algorithm.

1. INTRODUCTION

The goal of prognostics is to predict the time to failure (or similar measures, such as remaining useful life or time to maintenance) of a component or system. These predictions, when incorporated into an overall maintenance concept of operation, may provide several benefits, such as increased mission reliability and system availability, optimized spares positioning, and enhanced reliability centered maintenance (Massam & McQuillan, 2002).

At the earliest stages of design, these goals are documented as requirements. Typically, requirements statements are first developed at a higher (system) level, then flowed down to lower sub-systems and, potentially, individual components. As the requirements are developed at the lower tiers, they tend to become more specific and, thus, independently verifiable. Early work in writing requirements for prognostic algorithms relied on basic measurements such as the confidence interval at standard mean time to failure prediction (Kacprzynski et al., 2004), average bias and precision (Roemer, Dzakowic, Orsagh, Byington, & Vachtsevanos, 2005), and minimum time to prediction and minimum improvement of the service interval over legacy methods (Line and Clements, 2006).

More recently, several performance criteria for Prognostics and Health Management (PHM) have been developed, as documented in (Saxena et al., 2008), (Leao, Yoneyama, Rocha, & Fitzgibbon, 2008), and (Wheeler, Kurtoglu, & Poll, 2010). These criteria, though, are usually used as a means of measuring the performance of a prognostic algorithm, often in relation to other algorithms (say, for example, to determine the 'best-performing' algorithm out of a set). While these performance measures could, potentially, be used as the basis for a requirement (Tang, Orchard, Goebel, & Vachtsevanos, 2011), there are two issues with this approach. First, there are not currently accepted performance thresholds related to these measures (particularly for fielded systems). Second, most of these measures require knowing the true state of health of the component being analyzed, or at least the true time of failure. In many fielded systems, it is not acceptable to let a component run to failure, and not cost effective or accurate to determine the remaining life of a component removed before failure.

Often, the data that will be available for verification will be (1) how many instances of a component (across a fleet, for instance) were replaced and (2) of those, how many failed before they were replaced (as opposed to how many were replaced based on a time-to-maintenance prediction).

Due to the inherent uncertainties associated with prognostic algorithms, the remaining useful life prediction is typically given as a probability distribution around a mean predicted time to failure. Instead of trying to characterize and verify the shape of the predicted failure pdf, this paper uses a threshold requirement (i.e., "capture 95% of all failures") as a starting point for verification.

This document addresses two main issues of verification of prognostic algorithms. First, Section 2 discusses what a meaningful and verifiable prognostic requirement statement must include. Then, Section 3 provides a statistical approach to verifying such a requirement by considering the case of a single failure mode. Section 4 extends the analysis presented

Clements et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Approved for Public Distribution, PIRA# AER201305018 (LM Aero)

in Section 3 to the case of multiple failure modes, both prognosable and non-prognosable. Finally, Section 5 discusses some of the real-world implications of this verification approach.

2. Prognostic Requirement Statements

A primary issue with verifying prognostics algorithms is formulating a proper prognostic requirement statement. "Proper" in this sense means that the requirement is verifiable, which implies that the data needed to verify it can realistically be acquired. Unfortunately, the data most likely to be available is rather limited. In the normal course of developing a component, the design, analysis, and perhaps "lab data" will be useful for determining such things as average failure distributions, failure modes and probabilities, and potential precursor signals of failures. While helpful in designing the prognostic algorithm itself, this type of data will usually not be useful in verifying the algorithm. In practice, the data available for verification will be maintenance data, such as how long a component has been in service, how many times a component has been replaced, how many times it has failed, etc.

Consider the example of a flight control surface actuator such as an electro-mechanical actuator (EMA). Suppose a prognostic technique has been developed for this EMA that gives a time-to-maintenance (TTM) indication based on measured performance. Further, assume that maintenance is planned based on the prognostic indication. That is, the part is replaced when indicated, even if it has not failed. In an ideal world, once removed, the part would be analyzed to determine how much useful life remained in the component. Although the component could conceivably be placed in a test bench and operated until failure, doing so would not be economically feasible (not to mention issues such as recreating realistic flight conditions and load profiles). Instead, the part will most likely be repaired, recertified and placed back in the supply chain (or discarded). However, as mentioned before, there is some data available for verification: the number of times that EMA has been removed and whether or not it had failed in place before being removed (i.e., when the prognostic algorithm fails to give a maintenance time before failure). Note that these counts can be aggregated across all aircraft in a squadron (for example) to provide a statistically significant sample.

In addition to having reasonable access to the requisite data, the prognostic requirement statement must be written in such a way that it has an interpretation that is not ambiguous. To demonstrate some of the ambiguities that can arise with interpretation, an initial attempt at a prognostic requirement will be given and then refined as needed.

1st attempt: *The prognostic algorithm shall provide a time-to-maintenance such that at least 95% of failures of component XYZ will be avoided.*

As will be shown, there are several problems with this statement. First, there is no minimum bound on the time to failure of the prediction. Simply declaring "Component XYZ will fail in five minutes" (or some other arbitrarily short time) technically satisfies the requirement, but is practically useless. This minimum time to failure declaration requirement often stems from an analysis of the minimum useful notification, based on factors such as the lead time to procure a replacement component and how often the prognostic algorithm will be run. So, a second attempt is made:

2nd attempt: *The prognostic algorithm shall provide a minimum of 20 hours time-to-maintenance such that at least 95% of failures of component XYZ will be avoided.*

This attempt at a requirement statement at first glance may appear adequate (and, indeed, it is close), but as will be shown more clearly in the next section, there are still two problems with it – the confidence in the prediction and protection against 'overly conservative' predictions. In the next section, the verification approach will be presented as well as further refinements on the requirement statement to address these issues.

3. Verification Approach

It is important to understand that this verification technique is not trying to determine how well the prognostic algorithm is determining the actual remaining useful life distribution of a component. In fact, the prognostic algorithm does not even need to explicitly calculate the remaining life distribution. Rather, this approach to verification is based off the "avoid 95% of failures" portion of the requirement statement. Specifically, this approach evaluates whether the time-to-maintenance value (which the prognostic algorithm does provide) is adequately avoiding the specified percentage of failures.

The basic idea behind this approach is the expectation that the prognostic algorithm is, in fact, expected to "miss" a small percentage of failures. In the example requirement statements given in Section 2, an algorithm that satisfies the requirement will avoid at least 95% of failures; conversely, it will miss at most 5% of failures. So out of 100 replacements, the component can be expected to fail *about* 5 times. If the maintenance records indicate that there were actually 25 failures, the algorithm is probably not meeting the requirement. The rest of this section attempts to apply statistical theory to this approach to better quantify the confidence that the algorithm is meeting the stated requirement.

3.1. Assumptions

The following assumptions are made throughout the rest of this section.

1. The prognostic algorithm being verified provides a minimum time-to-maintenance that satisfies the minimum time constraint of the requirement statement.
2. Maintenance actions are planned based on the time-to-maintenance measures (that is, the part is replaced when indicated, even if it has not failed).
3. The number of times the component has been replaced (both due to failure or prognostic indication) is available.
4. The number of times the component failed before being replaced is available.
5. If a component is removed either due to a failure or based on a prognostic indication of imminent failure, then it will be replaced or serviced to a 'like new' condition before re-entering the supply chain.

There are several measures that may be output from a prognostic algorithm, including a best estimate of the remaining useful life, the shape of the remaining life distribution, and a best estimate of the time-to-maintenance for a given failure avoidance percentage. The first assumption is simply that the TTM value is made available (other measures may or may not be output as well).

The other three assumptions concern maintenance operations. The second assumption is that the time-to-maintenance measure is actually used. As will be discussed in Section 5.3, this assumption can be relaxed a little to provide for a "confidence building" period during which the prognostic algorithm is verified without the risk of excessive failures or unnecessarily maintenance actions. Finally, the last two assumptions provide the data needed for the verification calculations.

3.2. Single Failure Mode Construction

To start the derivation, assume that the prognostic algorithm is perfectly accurate. That is, the algorithm provides a consistent time-to-maintenance measure, which divides the failure rate pdf (whatever its shape) as shown in Figure 1. In this figure, t_0 is the time at which the prediction is made, t_M is the maintenance time, and f is the percentage of failures that would be avoided by performing maintenance at the indicated time.

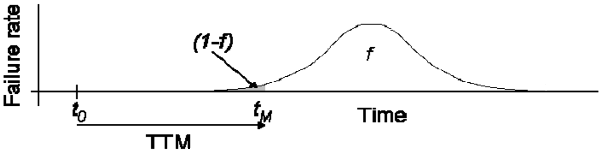


Figure 1. Prognostic Algorithm Time-To-Maintenance Prediction

The following two definitions are now made. Let n be the number of components replaced (failed and not failed) and x be the number of those replaced components that failed before being replaced. With these definitions, the probability of missing exactly x component failures is given by the standard binomial distribution:

$$P(\text{missing}\, x\, \text{failures}) = \binom{n}{x}(1-f)^{x} f^{n-x}\,, \tag{1}$$

where $\binom{n}{x}$ is the binomial coefficient:

$$\binom{n}{x} = \frac{n!}{(n-x)!x!}\,. \tag{2}$$

This expression characterizes the distribution of x given values for n and f. Figure 2 shows the probability mass function (pmf) of this binomial distribution for $n = 50$ and $f = 0.8$. As expected, the highest probability of failure occurs at $x = n\times(1-f) = 50\times(1-0.8) = 10$. Note that this is a discrete distribution – it is only defined on integer values of x. However, a different distribution can be calculated for every possible value of f between zero and one. For example, Figure 3 shows the binomial distributions for eleven different values of f. The distribution corresponding to Figure 2 ($f = 0.8$) is highlighted in blue. Further, since f can take on any value in the range from zero to one, Figure 3 can be filled in, yielding Figure 4.

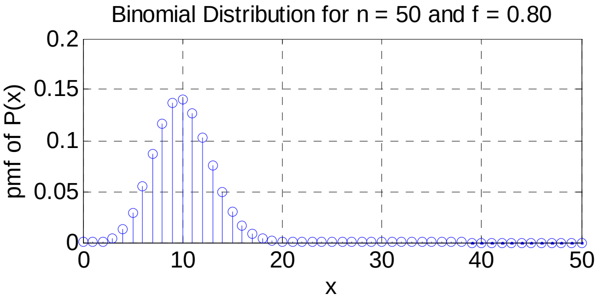


Figure 2. Binomial Distribution for $n = 50$ and $f = 0.80$

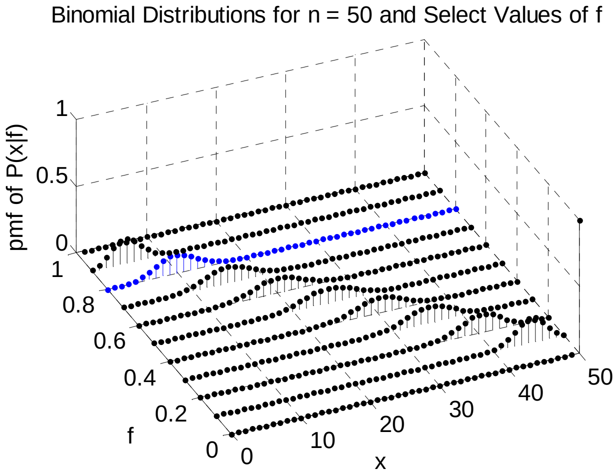


Figure 3. Binomial Distribution for $n = 50$ and Various Values of f

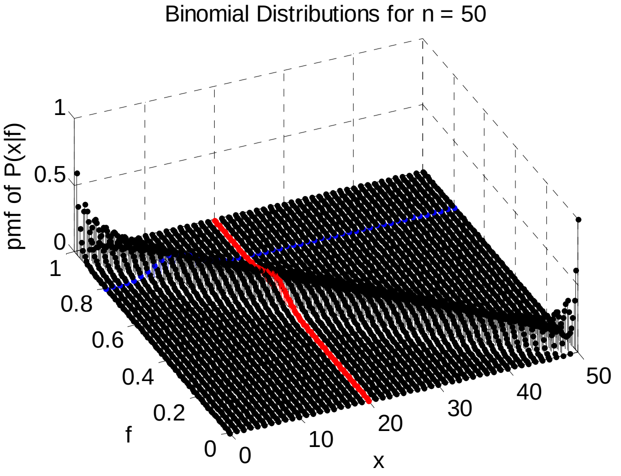


Figure 4. Binomial Distribution for n = 50 and All Values of f

Up to now, the values for f and n have been considered known. Now consider a slightly different problem: given values for n and x, what is the best guess for the value of f? For example, suppose that n = 50 and x = 20, as highlighted by the red line in Figure 4. This red line is the marginalization of the joint distribution of f. Note that this line is continuous in f and only defined over the range $0 \leq f \leq 1$. Before using this marginalization to develop a confidence measure, it must first be normalized to form a proper pdf. This results in a pdf for the marginalization of the joint distribution of f given by

$$(n+1)\binom{n}{x}(1-f)^x f^{n-x} . \tag{3}$$

As an example, the graph of the pdf of f for n = 50 and x = 20 is given in Figure 5. This corresponds to the red line shown in Figure 4.

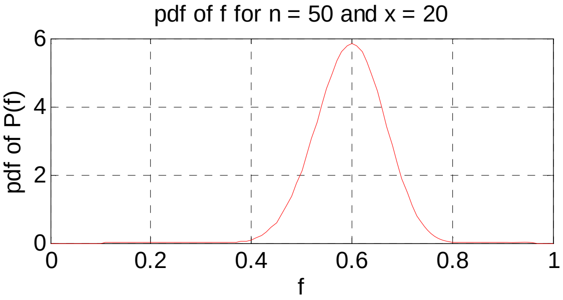


Figure 5. pdf of f for n = 50 and x = 20

Given the pdf for the distribution of f for a given set of values n and x, the confidence (i.e., probability) that the actual value of f is a given value (or, more accurately, that the actual value is within a range of values) can be found by calculating the area under the distribution for that range. Written as a formula, the confidence that the actual value of f is between two values a and b (where $a \leq f \leq b$) is given as follows:

$$\int_a^b (n+1)\binom{n}{x}(1-f)^x f^{n-x} df . \tag{4}$$

For example, suppose that the flight control surface EMA had been replaced 50 times, and 20 of those times were due to a failure of the EMA, the confidence that the prognostic algorithm was avoiding at least 95% of the failures would be equal to the area under the curve in Figure 5 from 0.95 to 1.00 (or 95% to 100%). Obviously, for this curve the confidence would be very close to zero, which intuitively makes sense given the number of failures that were incurred.

However, suppose there had only been one failure out of 50 replacements. The distribution for f in that scenario is shown in Figure 6. In this case the confidence that the failure avoidance is at least 95% is much higher (73.56%):

$$\int_{0.95}^{1.00} (50+1)\binom{50}{1}(1-f)^1 f^{50-1} df = 0.7356 \cdot \tag{5}$$

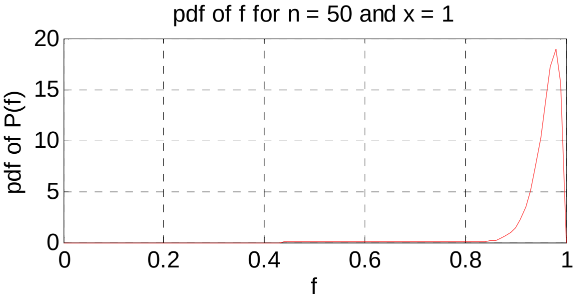


Figure 6. pdf of f for n = 50 and x = 1

3.3. Prognostic Requirement Statement Refinements

In Section 2, the requirement statement was left with two remaining issues. Now that a method of determining the confidence has been shown, the prognostic requirement can be further refined.

3rd attempt: *The prognostic algorithm shall provide a minimum of 20 hours time-to-maintenance such that at least 95% of failures of component XYZ will be avoided with 90% confidence.*

With the addition of the confidence measure, the requirement statement can be verified in a statistically meaningful manner. However, there is no protection against 'overly conservative' predictions. For example, a prognostic algorithm that claims 20 hours time-to-maintenance every time it is run (even if there were actually hundreds or more hours of remaining useful life), would meet the requirement, as it would definitely catch 95% of all failures of that component. Realistically (for components that are not so critical that they should never be allowed to fail in place), there is an expectation that an accurately tuned prognostic algorithm would 'miss' some (small) percentage of failures. Thus, instead of saying "… at least 95% of failures …"

(which implies between 95% and 100%), the upper bound is slightly reduced.

4th attempt: *The prognostic algorithm shall provide a minimum of 20 hours time-to-maintenance such that between 95% and 99% of failures of component XYZ will be avoided with 90% confidence.*

Figure 7 shows a grid plot of the confidence calculation for various values of x and n. Grid coordinates (i.e., combinations of x and n) that meet or exceed 90% confidence (per the example requirement statement above) are colored green. Similarly (for illustrative purposes), yellow indicates a confidence between 70% and 90%, and blue indicates less than 70% confidence. As can be seen, for some values of x, there is no value of n that will satisfy the requirement. Moreover, consider the case where $x = 4$. The requirement is satisfied only if n is between 168 and 237. Thus, if the prognostic algorithm is too conservative (and there have been more than 237 replacements for 4 failures), the confidence will drop below the threshold. Thus, a prognostic requirement written as shown above is not only verifiable from maintenance record data, but it also provides a means of identifying algorithms that are potentially too conservative in their time-to-maintenance predictions.

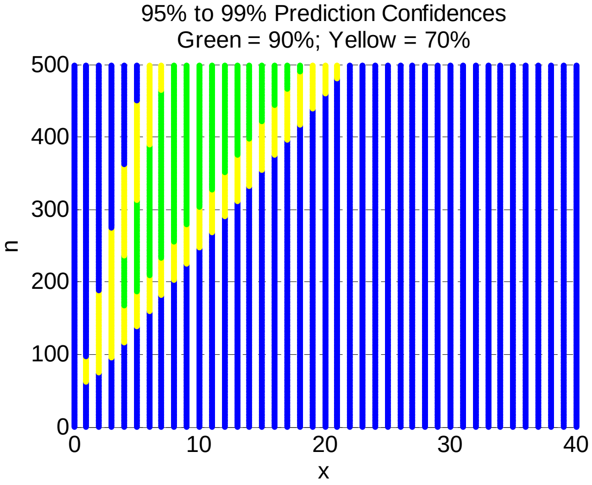


Figure 7. Prediction Confidence Regions

4. Extension to Multiple Failure Modes

The verification approach developed in Section 3 assumed that the component under analysis had a single failure mode. In reality, that is rarely the case, as components often have many failure modes and prognostic algorithms to cover only a few of them (typically, the most severe or frequent one(s)). This section extends the previous approach to verification of multiple failure modes.

As a start, it should be pointed out that the previous (single failure mode) approach still has applicability when considering the *total* prognostic requirement for a component. That is, if the goal is to determine how well a given component is meeting a goal of, say, between 90% and 99% coverage of all failure modes given whatever prognostic algorithm(s) are implemented, the single failure mode verification approach can be used. If, however, the desire is to see how well the individual algorithms (for a single component) are meeting individual goals, a new approach is warranted.

This extension to multiple failure modes is presented in three steps. First, Section 4.1 extends the previous approach to the case of two failure modes for a single component, each of which has its own prognostic algorithm. An overall confidence algorithm is then constructed from probabilistic principles for this two failure mode case. Section 4.2 then further extends this construction to the case of an arbitrary number of failure modes (again, where each has its own prognostic algorithm). Finally, Section 4.3 addresses the situation where one or more failure modes do not have associated prognostic algorithms.

4.1. Two Failure Mode Construction

To describe the approach taken in this extension, consider the case of a component with two failure modes. Further, assume that the component has been replaced four times (n = 4) of which two were due to component failure (x = 2). There are several scenarios that could lead to this result, as shown in Table 1. The first column of the table (n_1) is the number of times the component was replaced due to failure mode m_1. This includes both preemptive replacements based on prognostic indications and replacements required due to a failure of the component due to failure mode m_1. The second column (x_1) is the number of times the component failed due to failure mode m_1 before being replaced (that is, it was not replaced preemptively based on a prognostic indication). The next two columns (n_2 and x_2) represent the corresponding values for failure mode m_2.

Table 1. Possible Scenarios for Two Failure Mode Example

n_1	x_1	n_2	x_2	$n = n_1 + n_2$	$x = x_1 + x_2$
0	0	4	2	4	2
1	0	3	2	4	2
2	0	2	2	4	2
1	1	3	1	4	2
2	1	2	1	4	2
3	1	1	1	4	2
2	2	2	0	4	2
3	2	1	0	4	2
4	2	0	0	4	2

There are several things to note in this table. First, the total number of replacements (n) for each scenario must equal the (known) total number of replacements for the component (n_1 + n_2). Similarly, the total number of failures (x) for each scenario must equal the (known) total number of failures for the component (x_1 + x_2). Also, for each failure mode, the number of replacements due to that failure mode cannot be less than the number of actual failures due to that failure mode (or, algebraically, $n_i \geq x_i$). Finally, the individual values of n_1, x_1, n_2, and x_2 may not be known.

As shown in the derivation for a single failure mode, the probability of missing x_i out of n_i failures given that the failure mode is m_i (with corresponding value f_i) is given by:

$$P(x_i \mid n_i, m_i) = \binom{n_i}{x_i}(1 - f_i)^{x_i} f_i^{n_i - x_i} \,. \tag{6}$$

The total probability, *P(x,n)*, can then be found as

$$\begin{gathered} P(x,n) = \\ \sum P(x_1 \mid n_1, m_1) P(x_2 \mid n_2, m_2) P((m_1, n_1),(m_2, n_2)) \end{gathered} \tag{7}$$

where the summation is taken over all possible scenarios (as given in Table 1). The last term in the above summation, *P((m₁,n₁),(m₂,n₂))*, is the probability that m_1 occurred n_1 times and m_2 occurred n_2 times and is given by

$$P((m_1, n_1),(m_2, n_2)) = \binom{n}{n_1, n_2} P(m_1)^{n_1} P(m_2)^{n_2} \,. \tag{8}$$

The term $\binom{n}{n_1, n_2}$ is the multinomial coefficient (a generalization of the binomial coefficient). In general, the multinomial coefficient is given by

$$\binom{n}{n_1, n_2, \cdots, n_k} = \frac{n!}{n_1! n_2! \cdots n_k!} \,. \tag{9}$$

This coefficient can be thought of as the number of ways that n objects can be placed in k bins with n_1 objects in the first bin, n_2 objects in the second bin, etc.

The term $P(m_i)$ is the *a priori* known relative probability that a failure is due to failure mode m_i. This term can be determined from standard reliability data, such as Mean Time Between Failure (MTBF). If the MTBF for failure mode m_i is given by $MTBF_i$, the relative probability for failure mode m_i is given by

$$P(m_i) = \frac{MTBF_i^{-1}}{\sum_{i=1}^{k} MTBF_i^{-1}} \,. \tag{10}$$

Returning to the example of two failure modes (Table 1) with $n = 4$, $x = 2$, $f_1 = 0.8$, $f_2 = 0.9$, $MTBF_1 = 5000$ hours, and $MTBF_2 = 2000$ hours, evaluating Equation 7 yields $P(2,4) = 0.07532$. This value is interpreted to mean that there is a 7.53% probability that there will be 2 missed failures out of 4 total replacements, given two failure modes with the given MTBF values and prognostic algorithms with the given f_i values. Figure 8 is a 3-D stem plot of this total probability calculated for values of x between 0 and 10 and values of n between 0 and 50. The example calculated previously is highlighted on this plot.

In Figure 8 the values of f_1 and f_2 are held constant while x and n are varied. In application, the values of x and n will be known and the issue will be to determine the most probable ranges of f_1 and f_2. To address this question, we can plot the probability *P(x,n)* for various values of f_1 and f_2 for given values of x and n. Figure 9 shows such a plot for the previous example ($x = 2$ and $n = 4$). The point highlighted on the plot (with $f_1 = 0.9$ and $f_2 = 0.8$) corresponds to the same point highlighted in Figure 8.

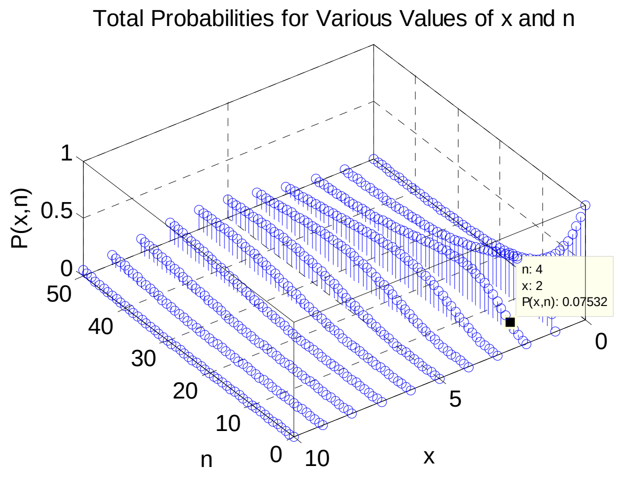


Figure 8. Plot of Total Probabilities for Various Values of x and n with Fixed f_1 and f_2

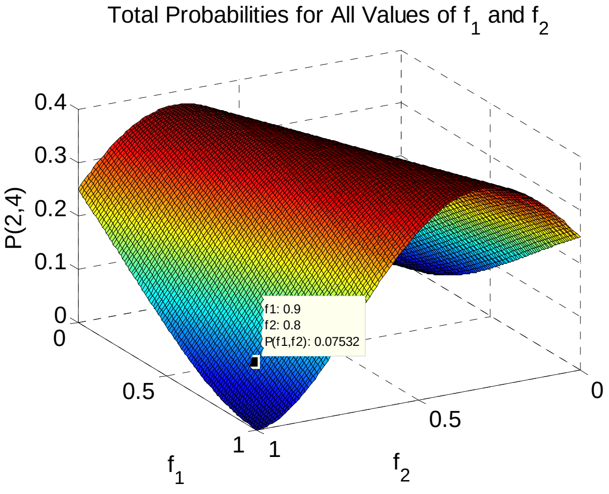


Figure 9. Plot of Total Probabilities for All Values of f_1 and f_2 with $x = 2$ and $n = 4$

Several observations can be made of Figure 9. First, the values f_1 and f_2 are continuous and bounded ($0 \leq f_i \leq 1$). Thus, the plot is a true surface and not discrete values. Second, the values of f_1 and f_2 that have the highest total probability (depicted as red in Figure 9) form a skewed line. This trend is more easily seen when the total probability plot is viewed "straight down", as shown in Figure 10. Such a 2-dimensional plot is called a "heat map", as the 3rd dimension is depicted purely as a gradient color (typically from blue to red). Recall that this example assumes we have "missed" 2 out of 4 failure events ($x = 2$ and $n = 4$). Thus the probability that both prognostic algorithms have f_i values of 50% ($f_1 = 0.5$ and $f_2 = 0.5$) should be high. As shown most readily in the heat map of Figure 10, this is indeed the case. However, since it is not presumed known how the values of x and n break down in relation to each failure mode, there are other scenarios that are just as probable. For example, the algorithm associated with failure mode m_1 may have a better (higher) value of f_1 that is compensated by a worse (lower) value of f_2. This tradeoff is evidenced by the straight banding of colors shown in Figure 10. If the failure rates (or MTBFs) of the two failure modes were equal, this line would be at a 45° angle to the f-axes. In this example, however, failure mode m_2 has a higher failure rate (or, equivalently, a lower MTBF) than failure mode m_1. Thus, a change in the value of f_2 will have a more pronounced effect on the total probability than a change in the value of f_1. This is illustrated by the skewing of the bands of the heat map to the $f_2 = 0.5$ line (or "to the vertical").

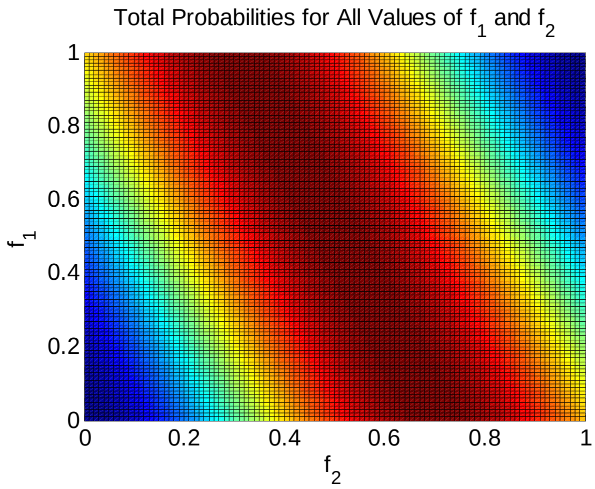


Figure 10. Heat Map of Total Probabilities for All Values of f_1 and f_2 with $x = 2$ and $n = 4$

Continuing with the generalization of the approach outlined for the single failure mode case, the total probability surface shown in Figure 9 is the marginalization of the joint distribution of the values of f_1 and f_2. Thus, if the surface is normalized such that the total volume under the surface is 1.0, the resulting surface will be the joint probability distribution function (pdf) of the values f_1 and f_2. Note that this joint pdf surface is the same shape as that of Figure 9 with the only difference being the scaling of the z-axis.

Having calculated the joint pdf, determining the confidence that the f_i values of the two prognostic algorithms are in given ranges is simply a matter of integrating the joint pdf over the ranges of interest. For example, to determine the probability that $0.45 \leq f_1 \leq 0.55$ and $0.4 \leq f_2 \leq 0.6$, the following double integral would be evaluated:

$$\int_{0.4}^{0.6}\int_{0.45}^{0.55} p(f_1, f_2)df_1df_2 \,, \tag{11}$$

where $p(f_1, f_2)$ is the joint pdf. For the case of $x = 2$ and $n = 4$, the integration will yield the following:

$$\int_{0.4}^{0.6}\int_{0.45}^{0.55} p(f_1, f_2)df_1df_2 = 2.87\% \,. \tag{12}$$

This confidence value is rather small, but recall that the total number of replacements (n) in this example is just four. If the number of replacements is increased to $n = 100$ and the number of missed failures kept at 50% ($x = 50$), the resulting pdf is shown in Figure 11. Compared to Figure 9, this pdf has much sharper roll-offs on either side of the "high-probability" line. The resulting confidence is also higher:

$$\int_{0.4}^{0.6}\int_{0.45}^{0.55} p(f_1, f_2)df_1df_2 = 8.46\% \ \,. \tag{13}$$

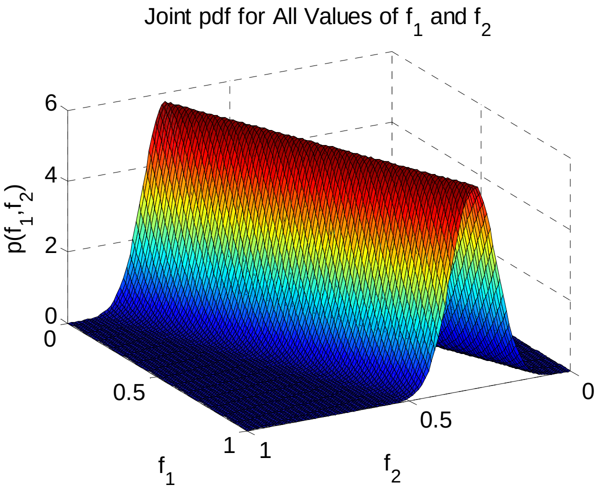


Figure 11. Joint pdf of f_1 and f_2 with Fixed $x = 50$ and $n = 100$

Even with the sharper roll-off from a higher number of replacements (n), the confidence for this example is still only 8.46%. This is primarily due to the large number of combinations of failure modes that will yield around 50% prediction (i.e., $f_1 \approx 0.5$ and $f_2 \approx 0.5$). A perhaps more realistic example, shown in Figure 12, represents five missed failures ($x = 5$) out of 100 replacements ($n = 100$). Calculating the confidence that $0.8 \leq f_1 \leq 0.99$ and $0.9 \leq f_2 \leq 0.99$ yields a value of 64.31%:

$$\int_{0.9}^{0.99}\int_{0.8}^{0.99} p(f_1, f_2)df_1df_2 = 64.31\% \,. \tag{14}$$

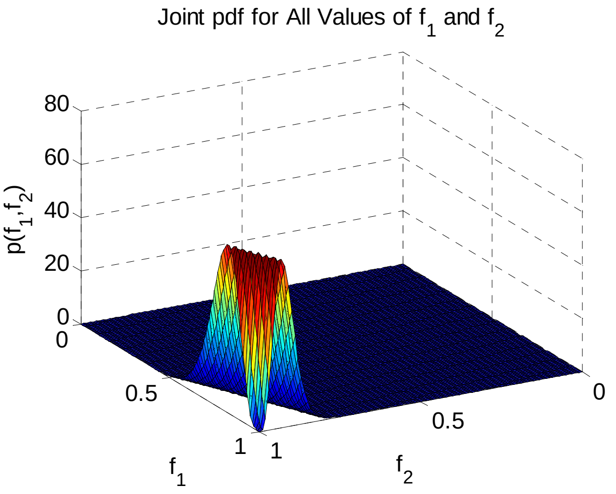


Figure 12. Joint pdf of f_1 and f_2 with Fixed $x = 5$ and $n = 100$

4.2. Generalization to *k* Failure Modes

The generalization from two failure modes to an arbitrary number of failure modes is straightforward. However, the process quickly turns into an exercise in proper indexing. The two primary points to keep in mind are the following. First, all combinations of $P(x_i, n_i)$ must be accounted for and weighted based on their frequency and relative failure rate. Second, for each scenario (i.e., set of values $\{x_i\}$ and $\{n_i\}$), the following must be true:

- $\Sigma_{i=1}^{k} x_i = x$, (15)
- $\Sigma_{i=1}^{k} n_i = n$, and (16)
- $n_i \geq x_i$. (17)

Equation 15 through 17 are simple generalizations of the two-failure mode construction (Section 4.1) to k failure modes. Although the derivation is quite involved (and omitted from this paper for space), one solution is presented below.

$$P(x,n) = \sum_{x_1=0}^{x} \sum_{x_2=0}^{x-x_1} \cdots \sum_{x_{k-1}=0}^{x-\Sigma_{i=1}^{k-2} x_i} \sum_{n_1=x_1}^{x_1+s} \sum_{n_2=x_2}^{x_2+s-s_1} \cdots \sum_{n_{k-1}=x_{k-1}}^{x_{k-1}+s-\Sigma_{i=1}^{k-2} s_i} \left\{ \binom{n}{n_1, n_2, \cdots, n_k} \prod_{i=1}^{k} \left[P(x_i \mid n_i, m_i) P(m_i)^{n_i} \right] \right\} \quad (18)$$

with:

$$\begin{aligned} s &= n - x \\ s_i &= n_i - x_i \\ x_k &= x - \Sigma_{i=1}^{k-1} x_i \\ n_k &= x - \Sigma_{i=1}^{k-1} n_i \\ P(x_i \mid n_i, m_i) &= \binom{n_i}{x_i} (1 - f_i)^{x_i} f_i^{n_i - x_i} \end{aligned} \quad (19)$$

The k-dimensional joint pdf is then found by normalizing the k-dimensional integral of $P(x,n)$ to one:

$$p(f_1, f_2, \cdots, f_k) = \frac{P(x,n)}{\int_0^1 \cdots \int_0^1 \int_0^1 P(x,n) df_1 df_2 \cdots df_k} . \quad (20)$$

Finally, the confidence that $a_i \leq f_i \leq b_i$ for $i = 1,\ldots,k$ is given by:

$$\int_{a_k}^{b_k} \cdots \int_{a_2}^{b_2} \int_{a_1}^{b_1} p(f_1, f_2, \cdots, f_k) df_1 df_2 \cdots df_k \; . \quad (21)$$

4.3. Non-Prognosable Failure Modes

Finally, the case of non-prognosable failure modes is considered. A non-prognosable failure mode is simply a failure mode for which there is no prognostic algorithm in place to predict remaining useful life. Note that the lack of a prognostic algorithm need not imply that such an algorithm could not be developed, only that it isn't in place for the component being analyzed. Further, it is reasonable that many failure modes of a component will not have prognostic algorithms (due to the relative infrequency of occurrence of the failure modes or a lack of technical understanding to develop such algorithms). All of these non-prognosable failure modes can, for the purposes of this analysis, be combined into a single non-prognosable failure mode with a composite MTBF given by:

$$MTBF_{non-prognosable} = \left(\sum MTBF^{-1} \right)^{-1} , \quad (22)$$

where the summation is taken over all non-prognosable failure modes.

A non-prognosable failure mode can then be characterized as a failure mode where $f = 0$. That is, there is zero probability that the failure mode will be predicted before that failure mode occurs. Further, this value of f is not probabilistic (it is deterministic with value zero), so it should not be included as a variable in the joint pdf. The effect of this characterization is to alter the calculation of the total probability function, $P(x,n)$. Without loss of generality, let the non-prognosable failure mode be listed as the last (or k^{th}) failure mode. Thus, $f_k = 0$. Now consider the term $P(x_k|n_k,m_k)$ that occurs in the calculation of $P(x,n)$. This term is given by:

$$\begin{aligned} P(x_k \mid n_k, m_k) &= \binom{n_k}{x_k} (1 - f_k)^{x_k} f_k^{n_k - x_k} \\ &= \frac{n_k!}{(n_k - x_k)! x_k!} (1-0)^{x_k} 0^{n_k - x_k} \\ &= \frac{n_k!}{(n_k - x_k)! x_k!} 0^{n_k - x_k} . \end{aligned} \quad (23)$$

From the constraints mentioned in Section 4.2, n_k must be greater than or equal to x_k, so consider the two cases $n_k = x_k$ and $n_k > x_k$ (recalling that $0! = 0^0 = 1$):

$$P(x_k \mid n_k, m_k) = \begin{cases} 1 & , n_k = x_k \\ 0 & , n_k > x_k \end{cases}. \tag{24}$$

This is consistent with the earlier characterization that a non-prognosable failure mode misses all occurrences of that failure mode (that is, when $n_k = x_k$ the probability is one). And for any case where fewer than all of the non-prognosable failure modes are missed ($n_k > x_k$), the probability is zero.

5. Implications

5.1. Tabulation

As discussed in Section 4, the overall confidence equation (Eq. 4) can be used when considering the verification of an overall requirement for a given component (as opposed to the multiple failure mode confidence equation given as Eq. 21). The evaluation of the integral in the overall confidence equation does have a closed-form solution. This closed form of the solution, though unwieldy to write down, is generally quicker (and more accurate) to calculate than to evaluate the integral using numerical techniques. Tabulations can be pre-calculated and stored instead of performing the complex calculation every time a value is needed.

5.2. Minimum Amount of Data Required for Verification

It is often desirable to know how much data will be required to verify a requirement. Such knowledge can be useful when scheduling and allocating resources to the verification task. To show how this information can be derived from this verification technique, consider an electro-mechanical actuator with the following prognostic requirement.

The prognostic algorithm shall provide a minimum of 20 hours time-to-maintenance such that between 95% and 99% of failures of the EMA will be avoided with 90% confidence.

The more failures that occur, the more replacements must have been performed to meet the requirement. Also, as mentioned in Section 3.3, for some numbers of failures, there is no number of replacements that will satisfy the requirement. For example, the least number of replacements that can conceivably be used to verify the requirement is 168, but only if there have been four failures in those 168 replacements. Table 2 shows the minimum number of replacements required for verification for a given number of failure occurrences.

These numbers, combined with the predicted reliability failure rate, can give a minimum value for the amount of data required and the time required to verify a prognostic algorithm. Unfortunately, these values are only minimum values. A more practical approach is given next.

Table 2. Minimum Number of Replacements Required For Verification

No. of Failures	Minimum No. of Replacements Required
0 - 3	N/A
4	168
5	187
6	210
7	234
8	257
9	281
10	> 300

5.3. Confidence Growth Curves

The blue line in Figure 13 shows a typical confidence growth curve for a prognostic algorithm. The black asterisks indicate 'missed' failures (i.e., failures that occurred before the indicated TTM). All other replacements were scheduled in accordance with a prognostic algorithm time-to-maintenance prediction.

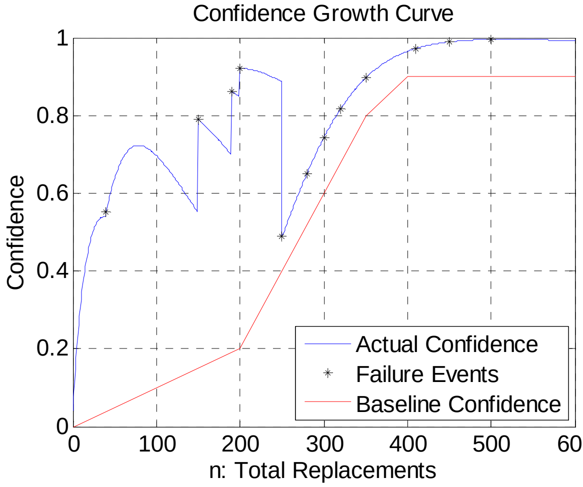


Figure 13. Confidence Growth Curve

As can be seen, the confidence starts out low and tends to increase as more data points are acquired. When a failure event occurs (and the associated un-predicted maintenance replacement), the confidence drops, particularly in the beginning when there are few data points. However, as the number of replacements increases, the effect of a failure event on the confidence curve is dampened.

The confidence growth curve, along with a baseline confidence curve, can be used to bound the time and data required for verification, as well as to provide a means of declaring a verification as failed. The baseline confidence curve is a minimum confidence threshold for the actual confidence curve. The shape of the baseline curve would be specified based on specific knowledge of the algorithm being verified. In general, though, it would tend to be pessimistic initially to allow for large swings in the confidence. An example of a baseline confidence curve is shown in Figure 13 in red.

In order for a prognostic algorithm to be verified, it would not only have to reach the desired confidence, but also do so without going below the baseline curve. If the actual confidence does dip below the baseline curve, the verification could be considered failed. An added benefit of the approach is that the baseline curve can constrain how much time is available for an algorithm to reach verification. For example, in Figure 13, the baseline confidence requires that the algorithm reaches verification (90% confidence) no later than by the 400th replacement.

The confidence growth curve can also be used to determine when to start relying on a prognostic algorithm. Often, particularly for a new prognostic technique, there can be reluctance to schedule maintenance on a part based on the prognostic prediction. In these cases, traditional maintenance concepts can initially be employed while a hypothetical confidence curve is tracked on the side. The hypothetical curve would assume that the prognostic prediction was acted upon. Similarly, if a failure occurred that the algorithm did not predict, the hypothetical confidence curve would be penalized accordingly. When and if the hypothetical confidence reaches a pre-determined threshold of acceptance, maintenance can start being scheduled based on the prognostic prediction instead of the traditional means.

6. Conclusion

This paper has addressed two of the central issues concerning verification of prognostic algorithms. First, the question of how to write a meaningful and verifiable prognostic algorithm requirement statement was considered. Through the course of the paper, it was shown that the following requirement statement template is both statistically meaningful and verifiable using available field data.

The prognostic algorithm shall provide a minimum of <TTM> hours time-to-maintenance such that between <LOWER>% and <UPPER>% of failures of component <COMPONENT> will be avoided with <CONFIDENCE>% confidence.

Second, a statistical approach to verifying such a statement was presented. The approach requires very few assumptions and can be easily pre-tabulated for a given requirement's failure threshold. Furthermore, implications of the approach can be used to bound the time and data necessary for verification as well as provide a means of building confidence in an un-tested algorithm.

Nomenclature

k	Number of failure modes
n	Total number of components replaced (failed and not failed)
n_i	Total number of components replaced (failed and not failed) due to failure mode m_i
$P(m_i)$	Relative probability of failure mode m_i
$P(x_i,n_i\|m_i)$	Probability of missing x_i out of n_i failures given the failure mode is m_i
$P(x,n)$	Probability of missing x out of n failures of any combination of failure modes,
x	Total number of components that failed before being replaced
x_i	Total number of components that failed due to failure mode m_i before being replaced

References

Azzam, H., Cook, J., & Driver, S. (2004), FUMS™ technologies for verifiable affordable prognostics health management (PHM), *IEEE Aerospace Conference Proceedings* (vol. 6, pp. 3764-3781), March 6-13.

Feldman, A., Kurtoglu, T., Narasimhan, S., Poll, S., Garcia, D., de Kleer, J., Kuhn, L. & van Gemund, A. (2010), Empirical Evaluation of Diagnostic Algorithm Performance Using a Generic Framework, *International Journal of the PHM Society* (vol. 1).

Halligan, G. R., Thumati, B. T., & S. Jagannathan (2011), Artificial immune system-based diagnostics and prognostics scheme and its experimental verification, *IEEE International Conference on Control Applications (CCA)* (pp. 958-963), 28-30 Sept. 2011.

Hao, L., Jinsong, Y., Ping, Z., & Xingshan, L. (2011), Review on Verification and Validation technology in integrated health management system, *10th International Conference on Electronic Measurement & Instruments (ICEMI)* (vol. 2, pp. 125-129), August 16-19.

Kacprzynski, G. J., Liberson, A., Palladino, A., Roemer, M. J., Hess, A. J., & Begin, M. (2004), Metrics and development tools for prognostic algorithms, *IEEE Aerospace Conference Proceedings* (vol. 6, pp. 3809-3815), March 6-13.

Leao B. P., Yoneyama, T., Rocha, G. C., & Fitzgibbon, K. T. (2008), Prognostics Performance Metrics and their Relation to Requirements, Design, Verification and Cost-Benefit, *IEEE International Conference on Prognostics and Health Management*.

Line, J. K., & Clements, N. S. (2006), Prognostics usefulness criteria, *IEEE Aerospace Conference Proceedings*, March.

Luna, J. J. (2010), Towards Defining and Allocating PHM Requirements for Military Systems, *Annual Conference of the Prognostics and Health Management Society*.

Lybeck, N., Marble, S., & Morton, B. (2007), Validating Prognostic Algorithms: A Case Study Using

Comprehensive Bearing Fault Data, *IEEE Aerospace Conference* (pp. 1-9), March 3-10.

Massam, S., & McQuillan, S. (2002), Verification of PHM capabilities: A joint customer/industrial perspective, *IEEE Aerospace Conference Proceedings* (vol. 6, pp. 2799-2813).

Mikat, H., Siddiolo, A. M., & Buderath, M. (2012), Virtual Framework for Validation and Verification of System Design Requirements to enable Condition Based Maintenance, *First European Conference of the Prognostics and Health Management Society*.

Ping, X., Wang, Z., & Vue, L. (2010), Prognostics and Health Management (PHM) System requirements and validation, *Prognostics and Health Management Conference* (pp. 1-4), January 12-14.

Roemer, M. J., Dzakowic, J., Orsagh, R. F., Byington, C. S., & Vachtsevanos, G. (2005), Validation and verification of prognostic and health management technologies, *IEEE Aerospace Conference* (pp. 3941-3947), March 5-12.

Saxena A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008), Metrics for Evaluating Performance of Prognostic Techniques, *IEEE International Conference on Prognostics and Health Management*, October 6-9.

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010), Metrics for Offline Evaluation of Prognostic Performance, *International Journal of the PHM Society* (vol. 1).

Tang, L., Orchard, M. E., Goebel, K., & Vachtsevanos, G. (2011), Novel metrics and methodologies for the verification and validation of prognostic algorithms, *IEEE Aerospace Conference* (pp. 1-8), March 5-12.

Usynin, A., Hines, J. W., & Urmanov, A. (2007), Formulation of Prognostics Requirements, *IEEE Aerospace Conference* (pp. 1-8), March 3-10.

Wheeler, K. R., Kurtoglu, T., & Poll, S. D. (2010), A Survey of Health Management User Objectives Related to Diagnostic and Prognostic Metrics, *International Journal of the PHM Society* (vol. 1).

Zhang, B., Tang, L., DeCastro, J., & Goebel, K. (2010), A verification methodology for prognostic algorithms, *IEEE AUTOTESTCON* (pp. 1-8), September 13-16.

Zhou, Y., Bo, J., Jie, Z., & Mingwei, G. (2013), Performance Metrics Assessment Method on Aircraft Prognostics and Health Management, *Fifth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)* (pp. 799-802), January 16-17.

BIOGRAPHIES

N. Scott Clements is a Systems Engineer at Lockheed Martin Aeronautics Company. He is currently researching physics of failure fault models and associated PHM techniques. He received his bachelor's degree from Mississippi State University in 1996 and his master's and doctoral degrees from Georgia Institute of Technology in 1998 and 2003, respectively. His research interests include PHM, data mining, verification techniques, and fault tolerant control systems.

David S. Bodden is a Senior Fellow in Flight Control/Vehicle Management Systems at Lockheed Martin Aeronautics Company. He received his bachelor's degree in Aerospace Engineering from Texas A&M University in 1976, and his master's degree in Engineering Mechanics from Virginia Tech in 1984. His technical background includes structural dynamics, air vehicle conceptual design, and multi-variable/intelligent control synthesis techniques. His current research area is application of PHM technology to improving system reliability.

Uncertainty in Impact Identification Applied to a Commercial Wind Turbine Blade

Raymond M Bond[1], Douglas E Adams[2]

[1,2] *Purdue University, West Lafayette, IN, 47906, USA*

rmbond@purdue.edu

deadams@purdue.edu

Abstract

This work evaluates the uncertainty of impact force and location estimates using an entropy-based impact identification algorithm applied to a commercial wind turbine blade. The effects of sensor placement, measurement directions and distance between impacts and sensor locations are studied. Results show that impacts to a 35m long wind turbine blade can be accurately located using a single tri-axial accelerometer regardless of sensor location. Uncertainties in impact force estimates are consistent across sensor locations. When omitting acceleration information in the spanwise direction, the bias and variance of force estimates is consistent, but when a single channel of acceleration data is used, both increase somewhat. Impact force identification error was found to be uncorrelated with the distance between the impact and sensor location. The entropy of the estimated force time history, an indicator of the impulsivity of the estimate, was found to be a good indicator of the quality of force estimate. The bias and variance of impact force estimation error was found to be directly correlated with the entropy of the impact force estimate. When considering validation test data from all possible sensor configurations, the entropy of the recreated force estimates was a better indicator of the force magnitude prediction interval than was the specific sensor configuration. By classifying impact force estimates based upon entropy values, impact force prediction intervals were more precisely determined than when all validation impact data were considered at once.

1. Introduction

Impact damage is a significant concern for most large composite structures because this type of damage is often below the surface and not evident from visual inspection. Composite damage mechanisms such as delaminations, substructure disbonds and core crushing can substantially reduce the strength of the structure without providing a clear visual indication. Inspection for this type of damage is often very time consuming and requires multiple inspection techniques to accurately identify the location and extent of these numerous damage mechanisms (Hayman, Wedel-Heinen, & Brondsted, 2008). Inspection of large rotor blades is particularly expensive and challenging, due to the size and inaccessibility of these blades. The inspection burden could be significantly alleviated by identifying the location and magnitude of applied impact loads. However, in order to make an informed maintenance decision based on these types of impact estimates, the associated uncertainty must be well understood. To this end, this work applies an entropy-based impact identification technique to a commercial wind turbine blade, and then evaluates the performance and uncertainty of impact location and force estimates.

Damaging impact loads are a concern for wind turbine blades both while in operation and during transport (Cripps, 2011; Veritas, 2006). Some examples of impact loads in operation are hail, bird strikes, or ice shedding from other blades. One study found that 7% of unforeseen malfunctions in 1.5MW wind turbines operating in Germany have been attributed to rotor blade problems, with an average down time of four days per failure(Hahn, Durstewitz, & Rohrig, 2007). Unforeseen repairs on wind turbines are especially costly, as these repairs are around 500% more expensive than regularly scheduled maintenance(Adams, White, Rumsey, & Farrar, 2011). An impact load estimation technique such as the one presented here has the potential to provide maintainers the information they need to limit the progression of damage by way of prompt repairs, schedule maintenance in advance, and track the loading history of blades to identify problematic trends.

Raymond Bond et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Impact identification methods have been widely studied (see, for example, (H. Inoue & Reid, 2001; Inoue, Kishimoto, Shibuya, & Koizumi, 1992; Hu, Matsumoto, Nishi, & Fukunaga, 2007; Stites, 2007; Yoder & Adams, 2008; Wang & Chiù, 2003)). These techniques are generally categorized as model-based techniques, based on an underlying model of the system, and artificial neural network based techniques, which are based on representative response training data and computational algorithms. Although neural network based techniques can be effective at locating impacts using a large array of sensors, model-based techniques are better suited to load estimation, even in sparse sensing configurations. For instance, other work from our research group (Budde, Yoder, Adams, Meckl, & Koester, 2009; Budde, 2010; Stites, Escobar, White, Adams, & Triplett, 2007; Stites, 2007; Yoder & Adams, 2008) has shown the ability to estimate impact load and position using a single sensor on filament-wound rocket motor casings and helicopter blades. This work builds on these previous efforts by developing an impact identification algorithm capable of monitoring very large and/or non-uniform structures with a single sensor.

2. Theory

This impact identification algorithm consists of two major steps: (1) estimating a set of potential impact forces assuming each of the possible input degrees of freedom, and (2) determining which of these force estimates most likely corresponds to the actual forcing location. The first step of this process is the same as that presented in (Yoder & Adams, 2008; Stites, 2007; Stites et al., 2007; Budde et al., 2009); the distinction between this algorithm and these other works is the method used to determine the likely impact location.

Estimated impact loads at each potential input degree of freedom are found by formulating and solving an overdetermined inverse problem based on experimentally estimated frequency response functions and measured responses. Given N_i input degrees of freedom and N_o output degrees of freedom, the response, $\{X(j\omega)\}$, can be expressed in terms of the frequency response function matrix, $[H(j\omega)]$, and forcing function, $\{F(j\omega)\}$, as follows:

$$\{X(j\omega)\}_{(N_0x1)} = [H(j\omega)]_{(N_oxN_i)}\{F(j\omega)\}_{(N_ix1)} \qquad (1)$$

For practical implementations, the number of response channels would be significantly fewer than the number of input degrees of freedom, that is, $N_0 \ll N_i$. With this constraint on the system configuration, the inverse problem of solving for $F(j\omega)$ based on $H(j\omega)$, $X(j\omega)$ and the relationship given in (1) is underdetermined with an infinite number of solutions. To reliably estimate the impact forces based on measured data, an overdetermined inverse problem is ideal in order to minimize the effects of measurement noise and error in the estimated frequency response functions. By assuming that the forcing function acts at a single degree of freedom, k, (1) can be re-written as follows:

$$\Big\{X(j\omega)\Big\}_{(N_0x1)} = \Big\{H_k(j\omega)\Big\}_{(N_ox1)} F_k(j\omega) \qquad (2)$$

With the force-response relationship in this form, the force can be determined given the frequency response function and any (non-zero) number of response channels. This inverse problem is overdetermined when more than one response channel is available. The linear least squares estimate of the forcing function at a particular frequency, $F_k(j\omega)$, is found by pre-multiplying 2 by the pseudoinverse of the frequency response function matrix at that frequency, $\{H_k(j\omega)\}^+$. Other numerical methods could also be used to solve for the least squares solution, but the pseudoinverse approach is advantageous for real-time impact monitoring because the computationally intensive portion of the solution procedure, calculating the pseudoinverse, is done prior to monitoring the structure for impacts, leaving only matrix multiplication to be done in real time.

Because the actual location of impact is unknown, the impact force must be estimated at each of the possible input degrees of freedom, and then the force estimates are analyzed to determine which of these force estimates most likely corresponds to the actual impact location. To determine which force estimate corresponds to the true location, the estimated force time histories for each location are analyzed to find which best matches the assumption of an impulsive impact force. The impulsivity of the recreated force time histories is quantified by evaluating the entropy of the impact force time histories.

Entropy in the context of information theory, is a measure of the average quantity of information contained in each event, in this case, in each sample of signal. The total information of a signal is defined as the minimum number of bits required to completely describe the signal. A purely random signal has the highest possible entry, and a completely uniform signal has zero entropy. For the correct impact location, the estimated force time history will closely resemble the actual impact force, which is very concentrated and ordered. In comparison, the estimated force time history at other locations will be much less ordered, more dispersed, and more random. Therefore, the estimated force time history with the lowest entropy most likely corresponds to the actual location of impact.

The mathematical definition of entropy is based on a set of N possible outcomes, x_i, with probabilities $p(x_i)$. In this case, the probability distribution used is a categorical distribution with the N possible outcomes corresponding to the signal amplitude falling into one of N possible ranges. The entropy, h, is computed as follows:

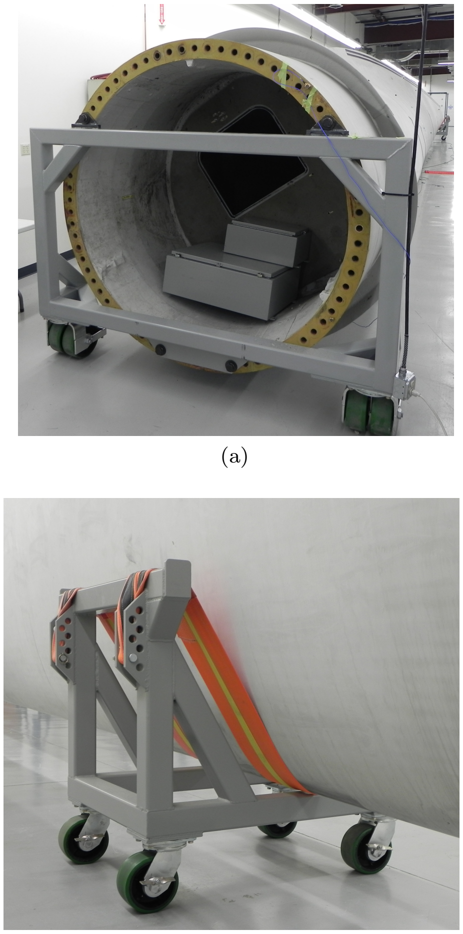
$$h = \sum_{i}^{N} -p(x_i)log_2(p(x_i)) \qquad (3)$$

The maximum value that the entropy may take is $log_2(N)$, corresponding to a uniform random distribution, and the lowest value is zero, corresponding to a constant signal. In this application, N was chosen to be 200, but the impact identification algorithm was found to be fairly insensitive to the choice of N.

The most important use of the entropy value is selecting the most likely impact location, but the minimum entropy value is also useful in evaluating how well the impact force estimate meets the assumed impulsive shape. The lower the minimum entropy value is, the better the force estimate matches expectations of a simple impulsive load. The relationship between the minimum entropy value and the quality of the force estimate will be evaluated with the experimental results of this study.

3. Experimental Setup

This impact identification technique was tested on a commercial wind turbine blade from a 1.6MW turbine. The blade was damaged in a lightning strike, and was delaminated at the tip with a portion of the tip missing. The blade was fastened at the root of the blade to a steel fixture, and supported towards the end of the blade with nylon straps. Figure 1 shows the blade and boundary conditions.


(a)

(b)

Figure 1. Photographs of test specimen, showing (a) attachment at the blade root, and (b) second blade support

Five tri-axial accelerometers were mounted to the blade to test the influence of sensor placement on the accuracy of impact identification. Accelerometer 1 is a PCB 356T18, an ICP triaxial accelerometer with nominal sensitivity of 1000mV/g. Accelerometers 2-5 are Silicon Designs 2460-050 DC coupled peizoresistive triaxial accelerometers with nominal sensitivities of 80mV/g. A grid of 130 impact locations was marked on the section of the blade between the root and the support. The vertical spacing between points was approximately 0.36m, and the horizontal spacing was roughly 0.91m. The sensor and impact locations are shown in Figure 2.

To create the frequency response function model of the blade, modal impact testing was carried out using a 5.5kg modal sledge hammer, model PCB 086D50. Peak force amplitude for these impacts ranged from 542.2lbf to 2469.3lbf, with a mean value of 1205.4lbf and a standard deviation of 371.5lbf. The bandwidth of excitation, as measured by the first frequency where the amplitude of the force spectrum drops to one tenth the maximum amplitude, ranged from 101.5Hz to 281.5Hz, with a mean bandwidth of 174Hz, and a standard deviation of 31.5Hz. Testing was conducted with ten impacts per point, sampled at 2560Hz for a duration of 2 seconds per impact. Frequency response functions were estimated with the H1 estimator.

To test impact identification accuracy, a validation data set was collected with two impacts per point. The impact identification algorithm was applied to response data, and the estimated location and impact magnitude were compared to the known values to evaluate performance.

4. Results

In order to test the performance of the impact identification algorithm on the blade, the response data from each validation impact was passed through the algorithm, and the estimated location and maximum force level was recorded. Two key metrics will be used to evaluate the

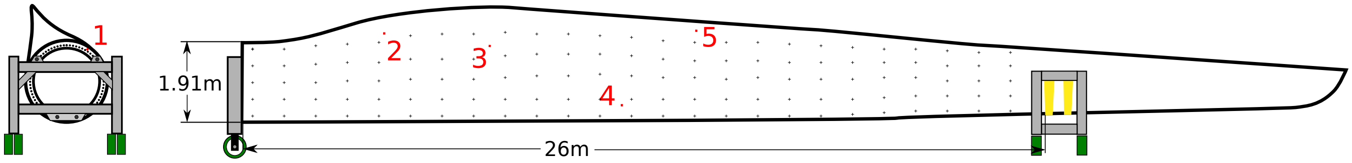


Figure 2. Illustration of the blade with approximate dimensions, sensor locations (enumerated in red), and impact locations (marked in black)

accuracy of the estimate: 1) the location identification accuracy, that is, how many of the validation impacts were correctly located, and 2) the magnitude of the peak force error, that is, the absolute value of the difference between the estimated and measured peak force relative to the measured peak force value.

To evaluate how sensor configurations affected the impact identification accuracy, the data from each of the five accelerometers was used individually to perform these validation simulations. Although the data for these sensors was collected simultaneously, only one sensor is used at a time in these validation tests. Accuracy is evaluated when data from all three measurement directions are used, when data from two of the three measurement directions are used, and when data from a single measurement direction are used.

The results of the validation simulation using all three response channels per sensor are summarized in Table 1. Regardless of the sensor location, 100% of impacts were accuracy located. The accuracy of the impact force magnitude estimates was also fairly consistent between sensor locations. The peak force identification error was biased towards underestimating the peak magnitude of the impact force by an average of 0.68%. The fifth sensor, which was placed the furthest towards the blade tip and closest to the trailing edge, performed the best of the tested locations. The force estimates using the fifth sensor had a median error of 3.3%, with 75% of the impact forces estimated within 5.6% of the true peak force value, and a maximum error of 21.2%. The sensor with the lowest force accuracy was the fourth sensor, which was located closer to the root of the blade and close to the leading edge of the blade. The force estimation error for the fourth sensor had a median value of 4% and a maximum error of 35.8%.

Table 1. Impact Identification Performance Using Each Triaxial Accelerometer

	Force Estimation Error (%)				
Sensor	Quartile			Mean	Max
	1	2	3		
1	1.7	4.1	6.4	4.8	27.9
2	1.7	3.8	6.4	4.9	30.7
3	1.8	4.2	6.6	4.8	25.6
4	2.0	4.0	6.8	5.0	35.8
5	1.6	3.3	5.6	4.2	21.2

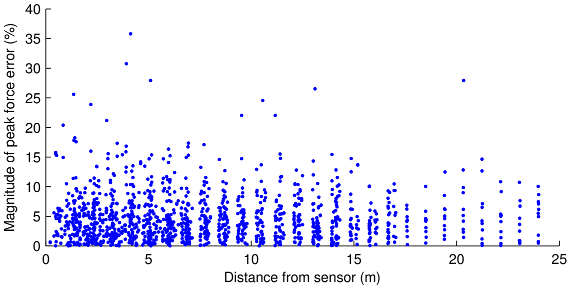


Figure 3. Force estimation error vs. distance from sensors for all combinations of sensor and impact location

From these results, the force accuracy shows no significant dependence on the distance from the sensor. Figure 3 is a scatter plot of the force error plotted against distance from the sensor, showing the results of validation tests using each of the available sensors. This plot illustrates the independence of the force accuracy on the distance from the sensor, even for very large distances. Most of the largest force estimation errors that were observed were at locations relatively close to the sensor, but this fact is probably in part due to the larger number of points that are an intermediate distance from the sensor than those very distant. Other impact force identification techniques have shown a linear increase in error with distance from the sensor (Seydel & Chang, 2001), so the fact that distance and accuracy are largely uncorrelated in this instance is significant.

The same type of validation test was repeated with only two of the three response directions used, with the response data in the axial direction ignored. Using this subset of the data produces results very similar to those when all three response channels per sensor are used. The results are detailed in Table 2. Most of the mean errors are marginally higher than when using all three channels, but the median errors and maximum errors are mostly lower. Based on these results, a bi-axial accelerometer could be used just as effectively as a tri-axial accelerometer, even on very large structures.

Table 2. Impact Identification Performance Omitting Data in the Axial Direction

Sensor	Force Estimation Error (%)				
	Quartile			Mean	Max
	1	2	3		
1	1.9	3.9	6.6	4.8	28.1
2	1.6	3.8	6.4	4.8	30.3
3	1.8	4.2	6.6	4.9	25.4
4	2.0	4.0	6.7	4.9	35.7
5	1.6	3.3	5.5	4.2	21.1

The validation simulation results show that entropy of the estimated force time histories is an effective measure to discriminate between the force at the actual impact location and the other erroneous force estimates. When using two or three response channels, every impact was correctly located, so the entropy value corresponding to the impact location was always the least. To better evaluate how effective the recreated force entropy is in discriminating between correct and incorrect locations, the recreated force entropy is compared between the actual impact locations and the other incorrect impact locations. The results from all of the three channel validation response simulations were considered, and histograms of the recreated force entropy values for correct and incorrect locations are shown in Figure 4. For this comparison and the following entropy discussion, the signals were discretized to 200 amplitude values. Therefore, a purely random signal would have $log_2(200) = 7.64$ bits of entropy. This comparison of entropy value distributions shows that the recreated force entropy is a very effective discriminator between the correct and incorrect locations. There is very little overlap between the two distributions, the entropy of the incorrect locations is tightly distributed, and the values of the correct location entropy are much lower than those from the corresponding incorrect locations. When entropy values from one impact were considered, the value corresponding to the correct location was always more than 1.5 times the interquartile range of the other entropies, with some values more than 10 times the interquartile range below the other entropy values. This measure indicates that for this set of data, not only is the entropy for the correct location always lowest, it is always a clear outlier of the distribution.

Entropy of the recreated force time histories effectively locates impacts because the value characterizes how well the force estimate meets the assumption of an impulsive load. Therefore, noise and error in the force estimate that alters the shape of the recreated force signal would generally contribute to an increase in the entropy of the force estimate. To evaluate the extent that the entropy of the recreated force time history is related to error in the force estimate, the force estimates were split into seven categories according to entropy value. Boxplots of the magnitude of force estimation error were plotted for each of these entropy ranges in Figure 5, along with a histogram showing the frequency of estimates within each of these entropy ranges. Statistical measures corresponding to each of these entropy ranges are detailed in Table 3.

Both the average bias and variance of the force estimation error are monotonically increasing with the entropy value of the estimated force. Both the mean error and standard deviation for the force estimates with entropy greater than four are more than three times the corresponding values for estimates with entropy less than 2.5. This result is important because with an understanding of how the recreated force entropy and force error are related, the uncertainty in a force estimate can be characterized based on the entropy value for that estimate.

To further investigate the quantification of impact load uncertainty based on estimated force entropy, empirical cumulative distributions of the magnitude of impact force estimation error were investigated. These distributions, shown in Figure 6, indicate the increasing uncertainty and higher force estimation error for higher entropy forces. Another important feature of these distributions is that the distribution based on all force estimates is a poor indicator of the uncertainty of force estimates with high or low entropy values. Categorizing force estimates based on recreated force entropy better characterizes the uncertainty in that force estimate.

When considering all force estimates, 95% of validation tests showed a peak force estimation error of less than 12.6%. In contrast, 95% of estimates with entropy of less than 2.5 bits were accurate within 5.5%, while the 95th percentile level was 22% for force estimates with more than 4 bits of entropy. Therefore, the uncertainty for force estimates in the lowest entropy range was significantly overstated by the distribution of all estimates, and the uncertainty for force estimates with the highest entropy was significantly understated by the distribution of all estimates.

5. Conclusions

The entropy-based impact identification technique applied here was able to identify the location and magnitude of impact loads applied to a commercial wind turbine blade using a single sensor regardless of sensor location. Impact force identification accuracy was independent of the proximity to the sensor, enabling even very large structures like this one to be monitored with very few sensors.

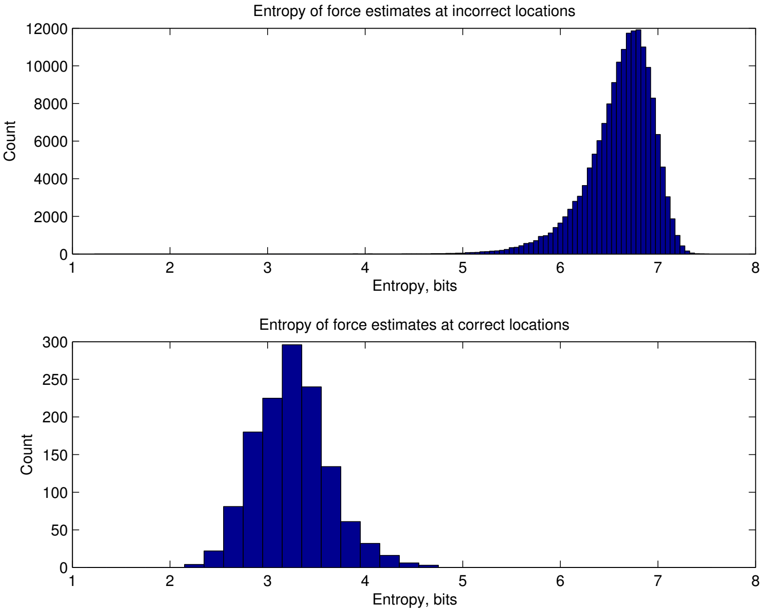


Figure 4. Histograms showing the distribution of entropy values for force estimates corresponding to incorrect locations (top) and correct locations (bottom)

Table 3. Peak force estimation error statistics corresponding to estimated force entropy

Entropy Range (bits)	Count	Magnitude of Peak Force Error (%)				
		Quartile			Mean	Standard Deviation
		1	2	3		
0 - 2.5	21	0.38	1.54	3.06	2.04	1.93
2.5 - 2.8	114	1.56	2.80	4.76	3.51	2.88
2.8 - 3.1	307	1.38	3.22	5.30	3.97	3.32
3.1 - 3.4	443	1.89	4.09	6.22	4.72	3.64
3.4 - 3.7	275	2.04	4.27	7.13	5.28	4.86
3.7 - 4	93	3.40	5.19	7.56	6.43	5.05
>4	47	3.03	4.3	8.87	6.97	6.19

The measure of recreated force entropy discriminates between force estimates from correct and incorrect locations very well, with the entropy at the correct location always being a statistical outlier. The value of the minimum recreated force entropy was shown to be a good indication of the uncertainty of that estimate. When categorizing the impact force estimates based on entropy values, the bias and variance of the peak force estimation errors monotonically increased with increasing entropy values. Comparing the 95$^{\text{th}}$ percentile force estimation accuracy levels between these entropy ranges showed that the uncertainty in force accuracy was more precisely identified when force estimates were categorized by entropy.

Identifying impact loads on large composite structures could significantly lower the associated inspection and repair costs by enabling condition based maintenance rather than scheduled wide area inspections and unscheduled repairs when damage progresses unexpectedly. This impact identification technique allows for minimal sensing configurations, and the method of characterizing the uncertainty of these estimates allows these condition based maintenance decisions to be well informed.

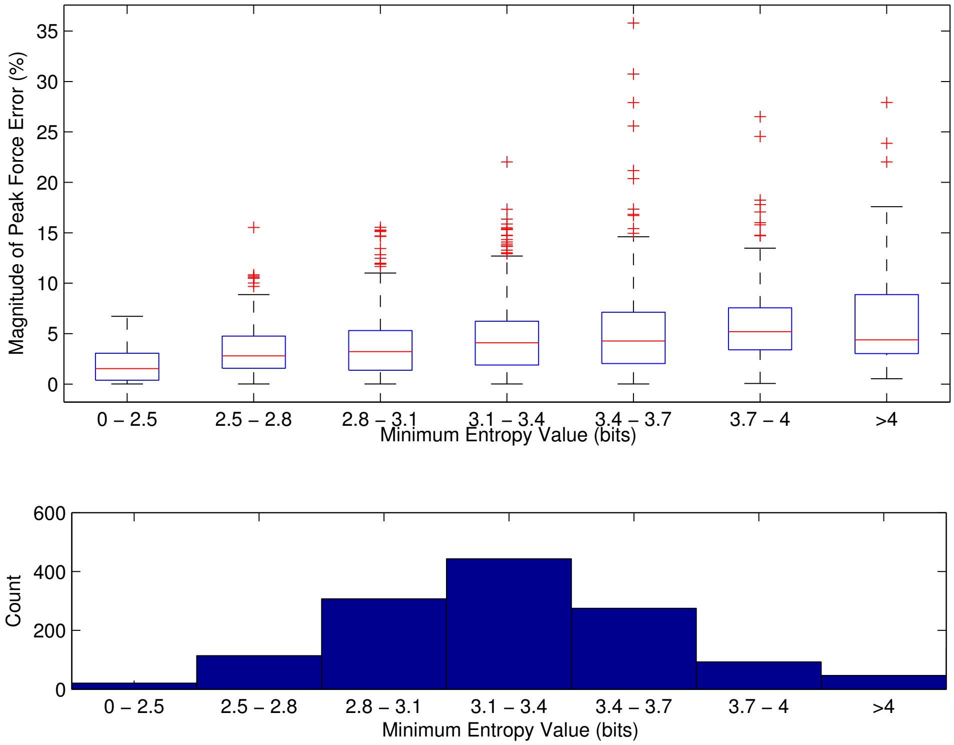


Figure 5. Relationship between impact force identification error and entropy of the force estimate

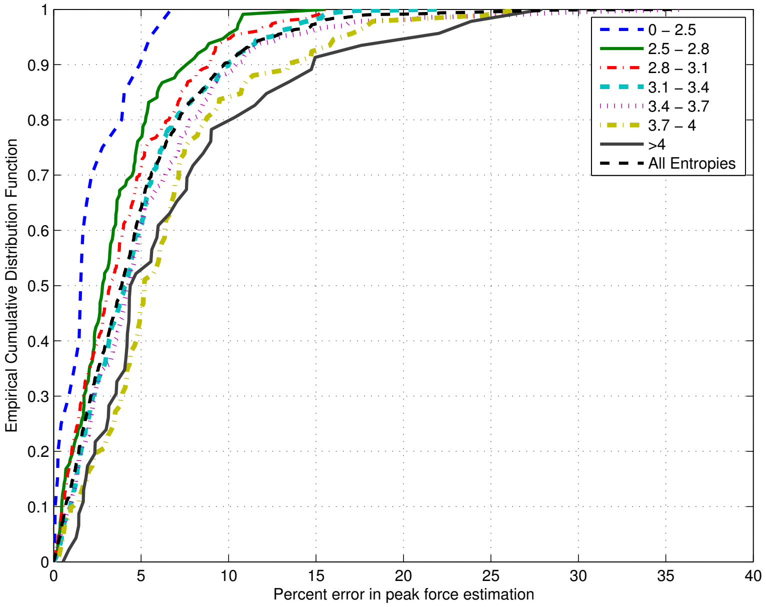


Figure 6. Empirical cumulative distributions of peak force estimation error for force estimates of varying entropy

Acknowledgment

The author would like to acknowledge support from Sandia National Laboratories through a graduate fellowship in the Sandia Campus Executive Fellowship Program.

References

Adams, D., White, J., Rumsey, M., & Farrar, C. (2011). Structural health monitoring of wind turbines: method and application to a HAWT. *Wind Energy*, *14*(4), 603–623.

Budde, C. (2010). *Impact force identification for composite helicopter blades using minimal sensing* (Unpublished master's thesis). Purdue University.

Budde, C., Yoder, N., Adams, D., Meckl, P., & Koester, D. (2009). Impact detection for fiberglass composite rotor blade..

Cripps, D. (2011). The future of blade repair. *Reinforced Plastics*, *55*(1), 28–32.

Hahn, B., Durstewitz, M., & Rohrig, K. (2007). Reliability of wind turbines. In *Wind energy* (pp. 329–332). Springer.

Hayman, B., Wedel-Heinen, J., & Brondsted, P. (2008). Materials challenges in present and future wind energy. *Warrendale: Materials Research Society*.

H. Inoue, J. H., & Reid, S. (2001). Review of inverse analysis for indirect measurement of impact force. *Applied Mechanics Reviews*, *54*, 503.

Hu, N., Matsumoto, S., Nishi, R., & Fukunaga, H. (2007). Identification of impact forces on composite structures using an inverse approach. *Structural Engineering and Mechanics*, *27*(4), 409–424.

Inoue, H., Kishimoto, K., Shibuya, T., & Koizumi, T. (1992). Estimation of impact load by inverse analysis: Optimal transfer function for inverse analysis. *JSME International Journal. Ser. 1, Solid Mechanics, Strength of Materials*, *35*(4), 420–427.

Seydel, R., & Chang, F. (2001). Impact identification of stiffened composite panels: Ii. implementation studies. *Smart Materials and Structures*, *Vol 10*, 370-379.

Stites, N. (2007). *Minimal-sensing passive and semiactive load and damage identification techniques for structural components* (Unpublished master's thesis). Purdue University.

Stites, N., Escobar, C., White, J., Adams, D., & Triplett, M. (2007). Quasi-active, minimal-sensing load and damage identification and quantification techniques for filament-wound rocket motor casings. In K. Tribikram (Ed.), (Vol. 6532, p. 65321E). SPIE. doi: 10.1117/12.715789

Veritas, D. (2006). Design and Manufacture of wind turbine blades. *Offshore and Onshore Wind Turbines*, *1*.

Wang, B., & Chiù, C. (2003). Determination of unknown impact force acting on a simply supported beam. *Mechanical Systems and Signal Processing*, *17*(3), 683–704.

Yoder, N., & Adams, D. (2008). Multidimensional sensing for impact load and damage evaluation in a carbon filament wound canister. *Smart Materials and Structures*, *Vol. 66 (7)*, 756-763.

A Model-Based Approach for Predicting the Remaining Driving Range in Electric Vehicles

Javier A. Oliva[1], Christoph Weihrauch[1], Torsten Bertram[1]

[1] *Institute of Control Theory and Systems Engineering, Technische Universität Dortmund, Germany*
javier.oliva@tu-dortmund.de
christoph.weihrauch@tu-dortmund.de
torsten.bertram@tu-dortmund.de

Abstract

The limited driving range has been pointed out as one of the main technical factors affecting the acceptance of electric vehicles. Offering the driver accurate information about the remaining driving range (RDR) reduces the range anxiety and increases the acceptance of the driver. The integration of electric vehicles into future transportation systems demands advanced driving assistance systems that offer reliable information regarding the RDR. Unfortunately the RDR is, due to many sources of uncertainty, difficult to predict. The driving style, the road conditions or the traffic situation are some of these uncertain factors. A model-based approach for predicting the RDR by combining unscented filtering and Markov chains is introduced in this paper. Detailed models are implemented for representing the electric vehicle and its energy storage system. The RDR prediction is validated by a set of simulation based experiments for different driving scenarios. Whereas traditional approaches consider the RDR as a deterministic quantity, to our knowledge, this approach is the first to represent the RDR by a probability density function. We aim to provide initial steps towards a solution for generating reliable information regarding the RDR which can be used by driving assistance systems in electric vehicles.

1. Introduction

Electric vehicles have emerged as a promising solution for reducing the oil dependence in transportation systems. Nevertheless, their integration into modern transportation systems is largely limited by the higher cost and the long charging times, on the one hand, and by the low driving range, on the other hand. The limited driving range has been considered as one of the major factors that affect the acceptance of electric vehicles. However, it has been shown (Franke, Neumann, Bühler, Cocron, & Krems, 2012) that reliable information regarding the remaining driving range (RDR) may help to overcome the range anxiety, i.e., the fear that the range of the vehicle is insufficient to reach the desired destination. Unfortunately the RDR is not easy predictable. Many stochastic factors such as the driving style, the traffic situation, the road conditions or the weather largely influence the RDR. It is therefore necessary to take these factors into account in order to meaningfully predict the RDR. To the best of our knowledge, few studies have addressed the RDR prediction. An approach that predicts the driving load of an electric vehicle based on driving pattern identification has been introduced by Yu, Tseng, and McGee (2012). To this aim, a library of identified driving patterns is used. Other approaches address the RDR prediction from a technological point of view. Conradi and Hanssen (2011) introduced an approach that combines a web server, a digital map and a mobile application. The mobile device sends the position of the vehicle and the current state of charge (SOC) of the battery to the web server, which first estimates the energy consumption along all possible routes and then, based on the SOC, calculates the maximum driving range. The main drawback presented in these approaches is that the RDR is treated as a deterministic quantity.

We introduce an approach that predicts the RDR under a stochastic framework. The basic theoretical foundation of this work is based on the work introduced by Daigle and Goebel (2011), where a model-based approach is applied to predict the remaining useful life (RUL) of pneumatic valves. The remainder of this paper is organized as follows. Section 2 introduces the proposed RDR prediction methodology. In section 3 the model of the electric vehicle is presented. Section 4 briefly discusses the algorithm for the state estimation. Section 5 explains the steps needed to predict the RDR. Section 6 presents simulation results for validating the proposed approach. Finally, section 7 concludes the findings of this paper and provides an outlook on the future work.

Javier A. Oliva et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

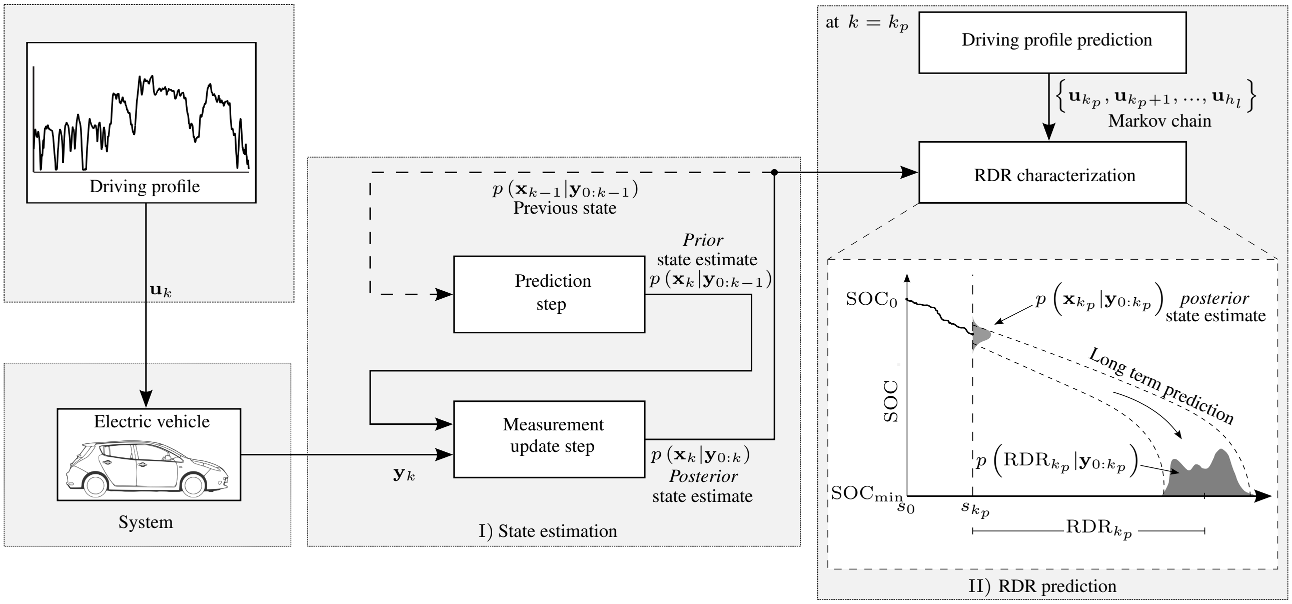

Driving profile
$\mathbf{u}_k$
Electric vehicle
System
$\mathbf{y}_k$
$p\left(\mathbf{x}_{k-1}|\mathbf{y}_{0:k-1}\right)$
Previous state
Prior
state estimate
$p\left(\mathbf{x}_{k}|\mathbf{y}_{0:k-1}\right)$
Prediction step
Measurement update step
$p\left(\mathbf{x}_{k}|\mathbf{y}_{0:k}\right)$
Posterior
state estimate
I) State estimation
at $k = k_p$
Driving profile prediction
$\left\{\mathbf{u}_{k_p}, \mathbf{u}_{k_p+1}, ..., \mathbf{u}_{h_l}\right\}$
Markov chain
RDR characterization
SOC_0
SOC
$\mathrm{SOC}_{\min}$
s_0
s_{k_p}
$p\left(\mathbf{x}_{k_p}|\mathbf{y}_{0:k_p}\right)$ posterior
state estimate
Long term prediction
$p\left(\mathrm{RDR}_{k_p}|\mathbf{y}_{0:k_p}\right)$
RDR_{k_p}
II) RDR prediction

Figure 1. RDR prediction architecture.

2. RDR Prediction Methodology

This section formulates the RDR prediction problem in electric vehicles and introduces the proposed prediction architecture.

2.1. Problem Statement

The RDR is defined as the distance that an electric vehicle can drive, with the energy stored in its battery, before recharging is required. Analogous to the Remaining Useful Life (RUL) calculation problem (Daigle, Saxena, & Goebel, 2012), the RDR prediction problem is formally formulated by considering the electric vehicle as a nonlinear system represented, in a discrete-time form, by

$$\begin{aligned}\mathbf{x}_k &= \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k, \mathbf{w}_k\right)\\ \mathbf{y}_k &= \mathbf{h}\left(\mathbf{x}_{k}, \mathbf{u}_k, \mathbf{n}_k, \mathbf{w}_k\right),\end{aligned} \tag{1}$$

where $\mathbf{x}_k$ is the state vector, $\mathbf{w}_k$ is the parameter vector, $\mathbf{v}_k$ is the process noise vector, $\mathbf{u}_k$ is the input vector, $\mathbf{y}_k$ is the output vector and $\mathbf{n}_k$ is the measurement noise vector. $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ represent the state and output function respectively. The RDR prediction problem is concerned with predicting the power demand of the electric vehicle, at a given time k_p, and identifying the distance between the position s_{k_p} and the location at which the electric vehicle must be recharged. By defining a threshold in the form

$$T(\cdot) = \left\{\begin{array}{l} 1 \\ 0 \end{array}\right. \tag{2}$$

it is possible to mathematically determine the recharging point and therefore the RDR. The challenge lies in determining the variables on which $T(\cdot)$ depends. This work considers the battery state of charge (SOC) to be the indicator that determines the threshold condition. Accordingly, the threshold is expressed as $T(\mathrm{SOC})$. Thus, $T(\mathrm{SOC}) = 1$ if $\mathrm{SOC}_{\min}$ is reached and $T(\mathrm{SOC}) = 0$ otherwise. The $\mathrm{SOC}_{\min}$ is usually dictated by the battery management system (BMS) of the electric vehicle in order to protect the battery cells from a possible total charge depletion.

2.2. Prediction Architecture

The RDR is a random variable that is influenced by many sources of uncertainty. This causes the RDR to be difficult to predict. For example, the lack of knowledge about the state variables, such as the SOC, the noise presented in the measurements or the ignorance regarding the future power demand, are some of the factors that largely contribute to the uncertainty of the RDR. Therefore, properly predicting the RDR requires accounting for these sources of uncertainty. To this aim we adopt a model-based methodology, as shown in Fig. 1. The approach proceeds basically in two phases, namely the *state estimation* (I) and the RDR *prediction* (II).

In the first phase, the states are recursively estimated. The *posterior* state estimate is computed in two steps. First, a prediction is made to obtain a *prior* state estimate $p\left(\mathbf{x}_k|\mathbf{y}_{0:k-1}\right)$. In the second step, as new measurements $\mathbf{y}_k$ become available, the predicted states are updated to compute $p\left(\mathbf{x}_k|\mathbf{y}_{0:k}\right)$. This estimate establishes the starting point for the RDR prediction phase.

In the second phase, at given time k_p, the RDR is predicted in two steps. First, future values of the driving profile[1] are predicted by means of Markov chains. In this way the driving profile is generated as a sequence of random variables $\left\{\mathbf{u}_{k_p}, \mathbf{u}_{k_p+1}, ..., \mathbf{u}_{h_l}\right\}$ which representatively capture driving patterns that occur in real-world driving situations. The index h_l denotes the horizon length of the driving profile prediction. In the second step, the uncertainty represented by the posterior state estimate $p\left(\mathbf{x}_{k_p}|\mathbf{y}_{0:k_p}\right)$ is propagated through the predicted driving profile until the $\mathrm{SOC}_{\min}$ is reached.

To carry out such an *uncertainty propagation* a sample-based approach in terms of a Monte Carlo simulation is employed. In this approach the probability density function $p\left(\mathbf{x}_{k_p}|\mathbf{y}_{0:k_p}\right)$ is approximated by a set of samples. Each sample is independently propagated through the predicted driving profile until the $\mathrm{SOC}_{\min}$ is reached. Once this happens, the RDR of all samples is identified and used to approximate the posterior $p\left(\mathrm{RDR}_{k_p}|\mathbf{y}_{0:k_p}\right)$.

3. Electric Vehicle Modeling

From a physical standpoint, an electric vehicle can be modeled by a forward-facing (dynamic) or by a backward-facing (quasi-static) approach (Guzzella & Sciarretta, 2005). In the forward-facing approach the vehicle is controlled to follow a desired speed. This approach considers the physical properties of each component of the powertrain and the dynamic interaction between them. Although this approach accurately describes the behavior of the electric vehicle, it requires high computational effort to solve the differential equations of the model.

The backward-facing approach overcomes this issue by assuming that the vehicle reaches the reference speed. With an imposed speed profile, the model calculates the forces acting on the wheels and processes them backwards through the powertrain. The calculation of the power demand depends only on algebraic equations, which decreases the computational burden of the model.

Nevertheless, the battery of the electric vehicle cannot be modeled using this approach since, as already mentioned, the SOC represents the indicator that determines the threshold condition of the prediction algorithm. Since this value is determined in the state estimation step, it cannot be represented by a quasi-static model. For this reason a dynamic model describes the behavior of the battery. The electric vehicle is modeled by combining these two approaches, as shown in Fig. 2.

In the following two sections both parts of the model are explained in detail. For the sake of better understanding, we omit expressing the variables of the quasi-static model as time dependent, since this model is described by a set of algebraic equations. The differential equations of the dynamic model are expressed in a discrete time form, since both, the state estimation and the RDR prediction modules, require a discrete-time representation of the battery model.

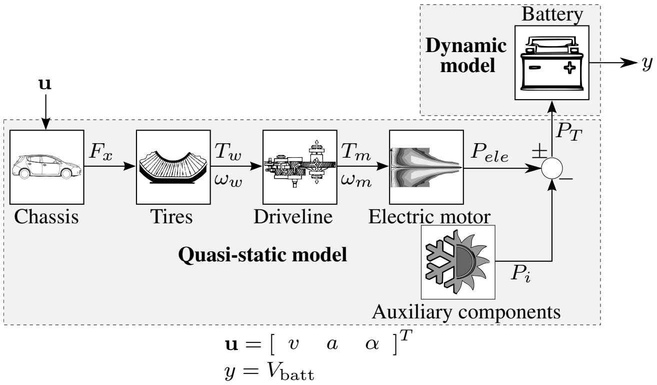


Figure 2. Combined quasi-static/dynamic model of the electric vehicle.

3.1. Quasi-static Model

An electric vehicle is composed by many components which, for simplification purposes, can be considered to move uniformly. Thus, the electric vehicle can be represented as a single lumped mass. As shown in Fig. 3, the force F_x required by the vehicle is given by

$$F_x = F_{air} + F_g + F_r + F_i. \tag{3}$$

The forces affecting the motion of the electric vehicle are:

- $F_{air} = \frac{1}{2}\rho_{air} c_w A v^2$ is the aerodynamic drag force,
- $F_g = mg\sin(\alpha)$ is the hill climbing force,
- $F_r = mgK_r$ is the rolling resistance,
- $F_i = ma$ is the force needed to accelerate/decelerate the electric vehicle,

where ρ_{air} is the density of air, c_w is the aerodynamic drag coefficient, A and m are the frontal area and the mass of the vehicle, g is the gravitational acceleration, K_r is the rolling resistance coefficient, α is the inclination (slope) of the road segment and v is the speed of the vehicle.

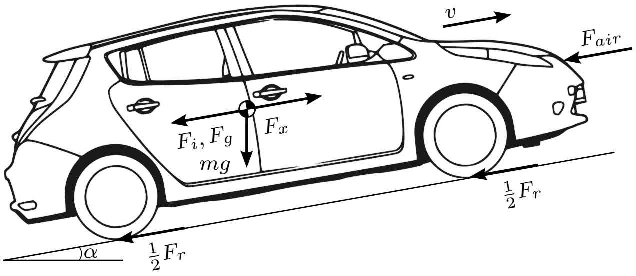


Figure 3. Forces acting during the motion of a vehicle.

[1] The driving profile is characterized by the speed (v) and acceleration (a) of the vehicle and by the slope (α) of the road.

The mechanical power P_{mec} demanded by the electric motor is easily calculated by means of a polynomial power requirement model (Kim, Lee, & Shin, 2013) as follows

$$\begin{aligned} P_{mec} = F_x v &= \frac{1}{2}\rho_{air} c_w A v^3 + mg \sin(\alpha) v + \\ &+ mgK_r v + mav. \end{aligned} \tag{4}$$

This model accurately calculates the mechanical power demand of a vehicle with a very low computational cost. The electrical power demand P_{ele} of the electric motor is then computed by

$$P_{ele} = \frac{P_{mec}}{\eta(\omega_m, T_m)}, \tag{5}$$

where η represents the electric motor's efficiency, $\omega_m = \frac{v i_d}{r_{tire}}$ is the rotational speed of the rotor and $T_m = \frac{F_x r_{tire}}{i_d}$ is the torque demand of the motor. Here r_{tire} and i_d are the tire's radius and the gear ratio of the driveline respectively.

Modern electric vehicles are able to recover a certain amount of the kinetic energy by means of regenerative braking. Such systems operate the electric motor in generator mode for delivering back the recovered energy to the battery. It is worth mentioning that η depends on whether the electric drive operates in motor or in generator mode. Accordingly,

$$\eta = \begin{cases} \eta_m(\omega_m, T_m) \le 1, & \text{motor mode} \\ \eta_g(\omega_m, T_m) > 1, & \text{generator mode.} \end{cases} \tag{6}$$

As shown in Fig. 2, the total electrical power P_T of the electric vehicle is composed of the electrical power demanded by the electric motor P_{ele} and by the sum of the power P_i consumed by each of the auxiliary components

$$P_T = P_{ele} + \sum_{i=1}^{n} P_i. \tag{7}$$

For the sake of simplicity, the power consumed by each of the auxiliary components is assumed to be constant.

3.1.1. Input Variables of the Quasi-static Model

To properly employ Eq.(4) in the RDR prediction algorithm, we need to differentiate between input variables and parameters. The input variables of the quasi-static model must be easily acquirable and should be highly dynamic, so that they cannot be considered as constant. Table 1 summarizes the dynamics and the dependency of the quasi-static model parameters. The parameters g and ρ_{air}, even though they can be easily determined, depend on the altitude and rarely change drastically during a trip. Also m, c_w and A are easily observed. They also change slowly since they depend on the vehicle design. The friction coefficient K_r, despite its high dynamic, cannot be easily determined. For this reason it is considered as a constant under the assumption that the road conditions do not change during the trip.

Table 1. Dynamics and dependency of the quasi-static model parameters.

Parameter	**Dynamics**	**Dependency**
a $(\mathrm{m/s^2})$	Very high	Driver, road, traffic
v $(\mathrm{m/s})$	High	Driver, road, traffic
m (kg)	Very low	Vehicle design
g $(\mathrm{m/s^2})$	Very low	Altitude
K_r	High	Road
α $(°)$	High	Road
ρ_{air} $(\mathrm{kg/m^3})$	Low	Altitude
c_w	Very low	Vehicle design
A $(\mathrm{m^2})$	Very low	Vehicle design

Our approach considers a, v and α as the input variables for the quasi-static model, since they meet the requirements previously mentioned. Accordingly, the input vector is given by

$$\mathbf{u} = \begin{bmatrix} v & a & \alpha \end{bmatrix}^T. \tag{8}$$

3.2. Battery Model

Our approach employs the model of a Li-ion cell shown in Fig. 4. The model combines the Kinetic Battery Model (KiBaM) (Manwell & McGowan, 1994) for capturing the nonlinear effects in the battery capacity, such as the recovery and the rate capacity effect, with a second order equivalent circuit based model which captures the dynamic response of the Li-ion cell. Furthermore, the combined model demands low computational effort, which makes it suitable for real-time applications.

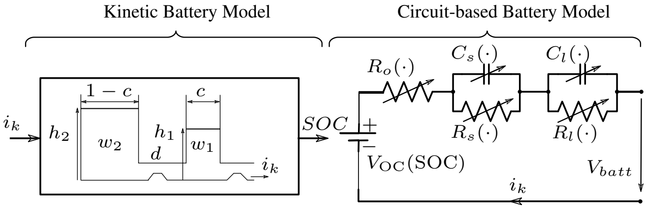


Figure 4. Combined battery model.

Even though the KiBaM was initially developed for lead acid batteries, it has been shown to be suitable for modeling the capacity behavior of Li-ion cells (Jongerden & Haverkort, 2009).

The Kinetic Battery Model abstracts the chemical processes of the battery discharge to its kinetic properties. The model assumes that the total charge of the battery is distributed with a capacity ratio $0 < c < 1$ between two charge wells. The first well contains the available charge and delivers it directly to the load. The second well supplies charge only to the first well by means of the parameter d. The rate of charge that flows from the second to the first well depends on both d and on the height difference between the wells $(h_2 - h_1)$. If the first well is empty, then the battery is considered to be fully discharged.

By applying load to the battery, the charge in the first well is reduced, which leads to an increment in the height difference between both wells. After removing the load, certain amount of charge flows from the second well to the first well until the height of both wells is the same. In this way the recovery effect is taken into account by the model. The rate capacity effect is also considered in this model. For high discharge currents, the charge in the first well is delivered faster to the load in comparison to the charge that flows from the second well. In this scenario there is an amount of charge that remains unused. The consideration of this effect is especially important for applications in electric vehicles, since the unused charge might eventually increase the driving range.

The KiBaM yields two difference equations which describe the change of capacity in both wells in dependence of the load i_k, the conductance d and the capacity ratio c:

$$w_{1,k+1} = a_1 w_{1,k} + a_2 w_{2,k} + b_1 i_k, \tag{9}$$

$$w_{2,k+1} = a_3 w_{1,k} + a_4 w_{2,k} + b_2 i_k, \tag{10}$$

where

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} = \mathrm{e}^{\begin{pmatrix} -\frac{d}{c} & \frac{d}{1-c} \\ \frac{d}{c} & -\frac{d}{1-c} \end{pmatrix}\Delta t}$$

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \int_0^{\Delta t} \mathrm{e}^{\begin{pmatrix} -\frac{d}{c} & \frac{d}{1-c} \\ \frac{d}{c} & -\frac{d}{1-c} \end{pmatrix}\vartheta} \mathrm{d}\vartheta \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

The term Δt is the sampling time used in the discretization of the model. The battery SOC is given by

$$\mathrm{SOC}_k = \frac{w_{1,k}}{c C_n 3600}, \tag{11}$$

where C_n is the nominal capacity of the battery. The right-hand-side equivalent circuit of Fig. 4 is compounded of three parts, namely, the open circuit voltage V_{OC}, a resistance R_o and two RC networks.

The voltage V_{OC} changes at different SOC levels, and is given by the following empirical equation

$$\begin{aligned} V_{\mathrm{OC}}(\mathrm{SOC}) \quad = \quad & a_1 - \frac{a_2}{\mathrm{SOC}} - a_3 \mathrm{SOC} + a_4 \ln(\mathrm{SOC}) + \\ & + a_5 \mathrm{e}^{\left(\frac{-a_6}{1-\mathrm{SOC}}\right)} + a_7 \mathrm{SOC}^2 + a_8 \mathrm{SOC}^4 + \\ & + a_9 \ln(1 - \mathrm{SOC}) + a_{10} \sin(a_{11} \mathrm{SOC}). \end{aligned} \tag{12}$$

The ohmic resistance R_o captures the I-R drop, i.e., the instantaneous voltage drop due to a step load current event. The $R_s C_s$ and $R_l C_l$ networks capture the voltage drops due to the electrochemical and the concentration polarization, respectively. In Fig. 4 the dependency of these parameters on the temperature and on the SOC is represented by the term $(\cdot)$. This part of the model yields two difference equations which describe the transient response of the battery:

$$v_{s,k+1} = \mathrm{e}^{-\frac{\Delta t}{R_s C_s}} v_{s,k} + \left(-R_s \mathrm{e}^{-\frac{\Delta t}{R_s C_s}} + R_s\right) i_k, \tag{13}$$

$$v_{l,k+1} = \mathrm{e}^{-\frac{\Delta t}{R_l C_l}} v_{l,k} + \left(-R_l \mathrm{e}^{-\frac{\Delta t}{R_l C_l}} + R_l\right) i_k. \tag{14}$$

Accordingly, the state vector of the battery model is given by

$$\mathbf{x}_k = \begin{bmatrix} w_{1,k} & w_{2,k} & v_{s,k} & v_{l,k} \end{bmatrix}^T. \tag{15}$$

The output y_k of the system, represented by the terminal voltage $V_{batt,k}$, is then computed as follows

$$y_k = V_{batt,k}(\mathrm{SOC}) = V_{\mathrm{OC}}(\mathrm{SOC}) + R_o i_k + v_{l,k} + v_{s,k}. \tag{16}$$

As presented in the previous section, the quasi-static model computes the total electrical power demand P_T. Nevertheless, the battery model requires the load current i_k as the input variable. Therefore, it is necessary to express i_k in terms of P_T. The load current i_k can be obtained from the definition of electrical power $P = IV$. Considering $P = P_T$ and $V = V_{batt}$ the terminal voltage can be expressed as

$$V_{batt} = \frac{P_T}{i}. \tag{17}$$

By substituting Eq. (17) into Eq. (16) and solving for i, the current at time k is given by

$$i_k = -\frac{C - \sqrt{C^2 - 4 P_T(\mathbf{u}_k) R_o}}{2 R_o}, \tag{18}$$

with

$$C = \left(V_{\mathrm{OC}}(\mathrm{SOC}) + v_{s,k} + v_{l,k}\right).$$

$P_T(\mathbf{u}_k)$ expresses the dependence of the total electrical power demand on the input vector given by Eq. (8). The solution with the positive part in the square root term of Eq. (18) is neglected, since its consideration would cause some current to be supplied by the battery when $P_T = 0$, which in practice is not possible.

4. State Estimation

At the beginning of the estimation and prediction steps the system states, and especially the initial SOC, are unknown. To obtain an accurate prediction of the RDR, the prediction module needs an initial starting point that is as accurate as possible. For that reason the state estimation has to converge to the true value before the prediction is carried out. In the prediction framework shown in Fig. 1 the task of the estimation step is to compute $p(\mathbf{x}_k|\mathbf{y}_{0:k})$, i.e., to represent the most up-to-date knowledge of the state variables at given time k

based on the history of measurements of the system.

For state estimation in nonlinear systems Bayesian tracking algorithms such as the particle filter (PF) (Rigatos, 2009), the extended Kalman filter (EKF) or the unscented Kalman filter (UKF) (Julier & Uhlmann, 2004) are the most established ones. This study uses the UKF as suggested by Daigle et al. (2012), because of the smaller number of sampling points and reduced computational complexity compared to the PF. The next section briefly describes the framework of the unscented Kalman filter that was implemented in this work.

4.1. Unscented Kalman Filter

The UKF applies the so-called Unscented Transform (UT) to approximate the distribution of the state variables. The UT considers each variable x as a random variable with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_x$ and computes the mean $\bar{\mathbf{y}}$ and covariance $\mathbf{P}_y$ of the output variable. This computation is carried out by choosing a set of deterministically weighted points $\mathbf{S}_i = \{w_i, \boldsymbol{\mathcal{X}}_i\}$, which are sampled from the distribution of x and are known as sigma points (Julier & Uhlmann, 2004). The sigma points are then propagated through a nonlinear function $\boldsymbol{\mathcal{Y}}_i = \mathbf{g}(\boldsymbol{\mathcal{X}}_i)$ that relates both the sigma points and the transformed sigma points. The posterior mean and covariance of the output variable can be recovered by

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2L} w_i \boldsymbol{\mathcal{Y}}_i \tag{19}$$

$$\mathbf{P}_y \approx \sum_{i=0}^{2L} w_i \left(\boldsymbol{\mathcal{Y}}_i - \bar{\mathbf{y}}\right)\left(\boldsymbol{\mathcal{Y}}_i - \bar{\mathbf{y}}\right)^T . \tag{20}$$

Many methods have been developed for selecting sigma points and these methods along with the choice of their parameters play an important role for the accuracy of the state estimation (Daigle & Goebel, 2010). In this study, the symmetric unscented transform is used. Here the set of 2L+1 sigma points are selected as:

$$\begin{aligned}
&\boldsymbol{\mathcal{X}}_0 = \bar{\mathbf{x}} && i = 0\\
&\boldsymbol{\mathcal{X}}_i = \bar{\mathbf{x}} + \left(\gamma\sqrt{(L+\lambda)\,\mathbf{P}_x}\right)_i && i = 1,...,L\\
&\boldsymbol{\mathcal{X}}_i = \bar{\mathbf{x}} - \left(\gamma\sqrt{(L+\lambda)\,\mathbf{P}_x}\right)_i && i = L+1,...,2L,
\end{aligned} \tag{21}$$

with the weights given by

$$\begin{aligned}
&w_0^{(m)} = \tfrac{\lambda}{L+\lambda} && i = 0\\
&w_0^{(c)} = \tfrac{\lambda}{L+\lambda} + (1-\alpha^2+\beta) && i = 0\\
&w_i^{(c)} = \tfrac{\lambda}{2(L+\lambda)} && i = 1,...,2L,
\end{aligned} \tag{22}$$

where L refers to the number of states in the state vector and $\left(\sqrt{(L+\lambda)\,\mathbf{P}_x}\right)_i$ is the i^{th} column of the square root of the weighted covariance matrix. The parameters λ, α, β and γ serve for scaling the sigma points in the state space and are chosen according to the (heuristic) recommendations of Julier and Uhlmann (2004). The algorithm 1 summarizes the main steps for state estimation using the unscented Kalman Filter.

Algorithm 1 Unscented Kalman Filter for State Estimation

```
Initialize:
$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0], \mathbf{P}_0 = E\left[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T\right]$
For $k = 1..., \infty$
1. Calcualte sigma points:
$\boldsymbol{\mathcal{X}}_{k-1} = \left[\ \hat{\mathbf{x}}_{k-1} \quad \hat{\mathbf{x}}_{k-1} \pm \gamma\sqrt{\mathbf{P}_{x_{k-1}}}\ \right]$
2. State prediction:
 a. Propagate the sigma points through the system model:
    $\boldsymbol{\mathcal{X}}_{k|k-1} = \mathbf{f}\left(\boldsymbol{\mathcal{X}}_{k-1}, \mathbf{u}_{k-1}\right)$
 b. Calculate the propagated mean and covariance:
    $\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} w_i^{(m)} \boldsymbol{\mathcal{X}}_{i,k|k-1}$
    $\mathbf{P}_{x_k}^- = \sum_{i=0}^{2L} w_i^{(c)} \left(\boldsymbol{\mathcal{X}}_{i,k|k-1} - \hat{\mathbf{x}}_k^-\right)\left(\boldsymbol{\mathcal{X}}_{i,k|k-1} - \hat{\mathbf{x}}_k^-\right)^T + \mathbf{R}_v$
 c. Sigma point propagation through the output model:
    $\boldsymbol{\mathcal{Y}}_{k|k-1} = \mathbf{h}\left(\boldsymbol{\mathcal{X}}_{k-1}\right)$
 d. Calculate the propagated mean:
    $\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} w_i^{(m)} \boldsymbol{\mathcal{Y}}_{i,k|k-1}$
3. Measurement update:
 a. Calculate the estimated covariance:
    $\mathbf{P}_{y_k} = \sum_{i=0}^{2L} w_i^{(c)} \left(\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\mathbf{y}}_k^-\right)\left(\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\mathbf{y}}_k^-\right)^T + \mathbf{R}_n$
    $\mathbf{P}_{x_k y_k} = \sum_{i=0}^{2L} w_i^{(c)} \left(\boldsymbol{\mathcal{X}}_{i,k|k-1} - \hat{\mathbf{x}}_k^-\right)\left(\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\mathbf{y}}_k^-\right)^T$
 b. Calculate the Kalman gain K:
    $\mathbf{K}_k = \mathbf{P}_{x_k y_k} \mathbf{P}_{y_k}^{-1}$
 c. Update the state estimation and covariance:
    $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{y}_k - \hat{\mathbf{y}}_k^-\right)$
    $\mathbf{P}_{x_k} = \mathbf{P}_{x_k}^- - \mathbf{K}_k \mathbf{P}_{y_k} \mathbf{K}_k^T$
```

The UKF estimate of $\mathbf{x}_k = \left[\ w_{1,k} \quad w_{2,k} \quad v_{s,k} \quad v_{l,k}\ \right]^T$ is used to calculate the output $y_k = V_{\text{batt},k}(\text{SOC})$ by using Eq. (16). The output voltage y_k depends on the SOC which is not part of the state vector. Therefore the SOC has to be calculated from the states in the algebraic Eq. (11). As further explained in the following section, if a RDR prediction is desired at given time k_p, the prediction module uses the last estimation $\hat{\mathbf{x}}_{k_p}$, from which a new set of sigma points is generated. The set of new sigma points is then used as the initial condition for the prediction step.

5. RDR Prediction

A prediction starts at given time k_p. Here, the posterior estimate $p\left(\mathbf{x}_{kp}|\mathbf{y}_{0:kp}\right)$ serves as the starting point for the prediction. As already mentioned in section 2, a sample-based approach for predicting the RDR is used. Therefore, the set of sigma points $\mathbf{S}_{kp}^i = \{\boldsymbol{\mathcal{X}}_{kp}^i, w_{kp}^i\}$, calculated by the UKF in the state estimation step, is used and propagated forward in time through simulation. All sigma points are propagated independently by following the predicted driving profile until $T^i_{kp} = 1$, i.e., until each sigma point reaches the $\mathrm{SOC}_{\min}$. The posterior mean $\overline{\mathrm{RDR}}_{k_p}$ and its covariance $\mathbf{P}_{\mathrm{RDR}_{k_p}}$ can be recovered, in a similar fashion as in Eq. (19) and Eq. (20), by

$$\overline{\mathrm{RDR}}_{k_p} \approx \sum_{i=0}^{2L} w^i_{k_p} \mathrm{RDR}^i_{k_p}, \tag{23}$$

$$\mathbf{P}_{\mathrm{RDR}_{k_p}} \approx \sum_{i=0}^{2L} w^i_{k_p} (\mathrm{RDR}^i_{k_p} - \overline{\mathrm{RDR}}_{k_p})(\mathrm{RDR}^i_{k_p} - \overline{\mathrm{RDR}}_{k_p})^T \tag{24}$$

As stated above, the propagation of all sigma points requires an hypothesized future driving profile. This work employs a stochastic approach based on Markov chains to predict the driving profile. The predictions are generated in such a way, that characteristic driving patterns of real-world driving situations are captured.

5.1. Driving Profile Prediction

Driving profiles can be modeled as a discrete-time Markov chain (T. Lee & Filipi, 2011). In this work two Markov chains are used. First, future values of speed and acceleration are generated by a 2D chain. Second, the slope profile is predicted by means of a 1D Markov chain independent of the speed and the acceleration. To apply a Markov chain the input space is quantized, for the speed/acceleration pair and for the slope, in such a way that each input variable takes a finite number of values. The inputs are given by $\{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_{h_l}\}$, with $\mathbf{u}_k = \begin{bmatrix} \mathbf{u}^{va}_k & u^{\alpha}_k \end{bmatrix}^T$. Here $\mathbf{u}^{av}_k = \begin{bmatrix} v_k & a_k \end{bmatrix}$ and $u^{\alpha}_k = \alpha_k$ represent parts of the input space given by the speed/acceleration pair and by the slope respectively, with h_l as the horizon length of the predicted profiles.

The Markov chain assumes that the transition probability from $\mathbf{u}_k$ to $\mathbf{u}_{k+1}$ only depends on the current state and not on the history of previous states.

The transition probabilities among the states are grouped in a transition probability matrix (TPM) $\mathbf{\Phi}$ such that

$$p_{ij} = \mathbf{\Phi}\left(\mathbf{u}_{k+1} = j | \mathbf{u}_k = i\right), \tag{25}$$

where p_{ij} is the ij^{th} element of $\mathbf{\Phi}$.

In this paper two transition probability matrices are used, namely $\mathbf{\Phi}^{va}$ and $\mathbf{\Phi}^{\alpha}$. The transition probabilities of $\mathbf{\Phi}^{va}$ are estimated from historical driving data and from standard driving cycles. The resolution of $\mathbf{\Phi}^{va}$ for the speed is 1 km/h in the interval $[0, 130]\mathrm{km/h}$ and for the acceleration it is 0.2 $\mathrm{m/s}^2$ in the interval $[-3, 3]\mathrm{m/s}^2$. $\mathbf{\Phi}^{\alpha}$ is estimated from real road height profiles and has a resolution of $0.5°$ in the interval $[-10, 10]°$. These resolutions offer a good trade-off between computational effort and accuracy in the prediction.

For estimating both TPMs the maximum likelihood estimation method (T. C. Lee, Judge, & Zellner, 1970) is applied. The transition probability p_{ij} is computed by

$$p_{ij} = \frac{n_{ij}}{\sum\limits_{j=1}^{s} n_{ij}} = \frac{n_{ij}}{n_i}, \tag{26}$$

where n_{ij} is the number of times a transition from $\mathbf{u}_i$ to $\mathbf{u}_j$ has occurred, and n_i is the total number of occurrences of $\mathbf{u}_i$. Algorithm 2 summarizes the steps required for the prediction of a driving profile.

Algorithm 2 Driving Profile Prediction

```
Require: $\mathbf{\Phi}^{va}, \mathbf{\Phi}^{\alpha}, v_{k_p}, a_{k_p}, \alpha_{k_p}, h_l$
Ensure: $\{\mathbf{u}_k, \mathbf{u}_{k+1}, ..., \mathbf{u}_{h_l}\}$
  Initialize:
  $i \leftarrow 0$
  $k \leftarrow k_p$
  $v_i \leftarrow v_{k_p}$, $a_i \leftarrow a_{k_p}$, $\alpha_i \leftarrow \alpha_{k_p}$
  for $l = 1$ to $h_l$ do
    Randomly draw $\mathbf{u}^{av}_j = \begin{bmatrix} v_j & a_j \end{bmatrix}$ for the next state according to $\mathbf{\Phi}^{va}\left(\mathbf{u}^{va}_{k+1} = \mathbf{u}^{va}_j | \mathbf{u}^{va}_k = \mathbf{u}^{va}_i\right)$
    $v_{k+1} \leftarrow v_j$
    $a_{k+1} \leftarrow a_j$
    $\mathbf{u}^{av}_{k+1} \leftarrow \begin{bmatrix} v_{k+1} & a_{k+1} \end{bmatrix}$
    Randomly draw $u^{\alpha}_j = \alpha_j$ for the next state according to $\mathbf{\Phi}^{\alpha}\left(\mathbf{u}^{\alpha}_{k+1} = u^{\alpha}_j | \mathbf{u}^{\alpha}_k = u^{\alpha}_i\right)$
    $\alpha_{k+1} \leftarrow \alpha_j$
    $u^{\alpha}_{k+1} \leftarrow \alpha_{k+1}$
    Create the driving profile
    $\mathbf{u}_{k+1} \leftarrow \begin{bmatrix} \mathbf{u}^{av}_{k+1} & u^{\alpha}_{k+1} \end{bmatrix}^T$
    $i \leftarrow j$
    $k \leftarrow k + 1$
  end for
```

The prediction is initialized with the current values v_{k_p}, a_{k_p} and α_{k_p}. Both Markov chains are generated separately in an iterative manner, until the desired length h_l of the profile is reached.

5.2. RDR Characterization

Until now the prediction of the RDR, as computed by Eq. (23), requires propagating the set of sigma points through a single predicted driving profile. Nevertheless, this procedure only accounts for the uncertainty related to the state estimation. Therefore, the uncertainty presented in the driving profile prediction is not taken into account. This issue is approached by generating multiple hypotheses about the future driving profile and by propagating the set of sigma points through each of them. In this way the uncertainty of the driving profile is accounted making the prediction of the RDR more meaningful.

After all sigma points, along all predicted driving profiles, reach the $\text{SOC}_{\min}$, i.e., $T^{ij}_{kp} = 1$, the posterior $p\left(\text{RDR}_{k_p}|\mathbf{y}_{0:k_p}\right)$ is approximated by means of the kernel density estimation (Bowman & Azzalini, 1997)

$$p\left(\text{RDR}_{k_p}|\mathbf{y}_{0:k_p}\right) \approx \frac{1}{N_u h}\sum_{j=1}^{N_u} K\left(\frac{\text{RDR}_{k_p} - \overline{\text{RDR}}^{j}_{k_p}}{h}\right), \quad (27)$$

where N_u is the number of predicted driving profiles, $K(\cdot)$ is the kernel, which in this work is Gaussian, and h is a smoothing factor known as bandwidth. It must be noted that the estimate kernel density estimate is based only on the set of mean posteriors $\left\{\overline{\text{RDR}}^{j}_{k_p}\right\}_{j=1}^{N_u}$ calculated with Eq. (23), as shown in Fig. 5. This implies that the covariances $\left\{\mathbf{P}^{j}_{\text{RDR}_{k_p}}\right\}_{j=1}^{N_u}$ are not taken into account. In our case this is justified since the uncertainty added by driving profile prediction is considered to be large in comparison to the uncertainty related to the state estimation. For the sake of better understanding, Fig. 5 shows the propagation of only one sigma point through all predicted driving profiles.

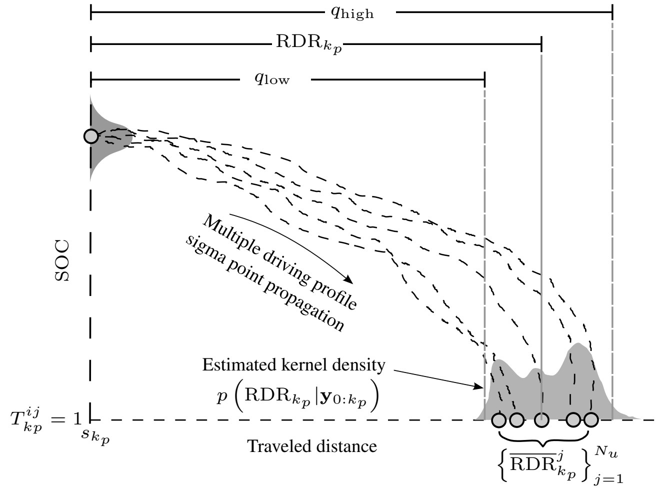


Figure 5. Propagation of one sigma point through multiple driving profiles.

Since $p\left(\text{RDR}_{k_p}|\mathbf{y}_{0:k_p}\right)$ is usually non Gaussian, we rely on the median for estimating the RDR and on quantiles, here represented by q_{low} and q_{high}, as the measure of spread (Hoaglin, Moesteller, & Tukey, 1983).

6. Results and Discussions

This section demonstrates and validates the proposed approach for predicting the RDR. The performance and the accuracy of the proposed approach for different driving scenarios, namely the city, rural areas and the highway, is investigated. Accordingly, the standard driving cycles shown in Fig. 6 are used in the simulations.

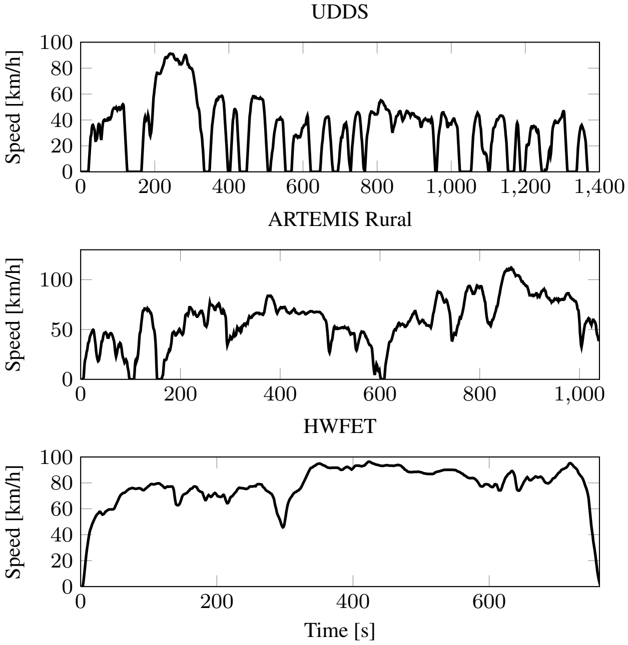


Figure 6. Standard drive cycles used in the simulation.

The parameters of both, the quasi-static and the battery model, are presented in table 2. The parameter of the quasi-static model are obtained from manufacturer's data sheets of the Nissan Leaf (Hayes, Oliveira, Vaughan, & Egan, 2011). The rolling resistance coefficient K_r is chosen to represent a dry road. It must be noted that the parameters of the battery model, correspond to those of one cell. These parameters are identified with the help of experimental data from a Li-ion cell. Since the parameters of the equivalent-circuit based model depend on both, the temperature and the battery SOC, they can not be considered as constant. Therefore, lookup tables are used to store them. To properly simulate the behavior of the entire battery pack, the cell capacity and the nominal voltage are scaled up to 24 kWh and 403.2 V respectively. For all experiments a temperature of 25 °C is assumed.

Table 2. Parameters of the quasi-static and the battery model.

Quasi-static Model		Battery Model	
Parameter	Value	Parameter	Value
A	2.29 m^2	C_n	2.15 Ah
c_w	0.28	V_{nom}	4.2 V
m	1520 kg	V_{lim}	2.8 V
K_r	0.7	d	1.4×10^{-5}
$T_{m,\max}$	280 Nm	c	0.96
$P_{ele,\max}$	80 kW		
r_{tire}	0.3 m		
ρ_{air}	1.226 kg/m^3		
g	9.81 m/s^2		

Each experiment is carried out as follows. First, for each driving scenario a sufficiently large drive cycle is created, so that the electric vehicle can drive until it reaches the $\mathrm{SOC}_{\min}$. During the simulation both the current and the terminal voltage of the battery are measured, which are then used by the UKF for estimating the states of the battery. The RDR is predicted every 1000 seconds for the city scenario and every 500 seconds for both, the simulation in rural areas and on the highway. The reason for this is that the time needed for simulating each scenario largely depends on the speed range of the drive cycle. The higher the speed of the drive cycle, the shorter the time it takes to finish the simulation and therefore the lower the amount of RDR predictions that can be done.

For the sake of demonstration, Figs. 7, 8 and 9 depict the RDR prediction process at prediction time $k_p = 2$, i.e., after 1000 seconds for the city scenario and after 500 seconds for the rural and highway scenarios. In all cases, the initial battery SOC is approximately 0.9. Multiple simulations are carried out simultaneously based on the predicted driving profiles. We have set $\mathrm{SOC}_{min} = 0.1$ for all experiments. As it can be seen, the SOC curves evolve tighter during the simulation of the highway scenario than in the first two scenarios. This causes the uncertainty related to the predicted driving profiles to be lower and therefore the estimated RDR kernel density is narrower.

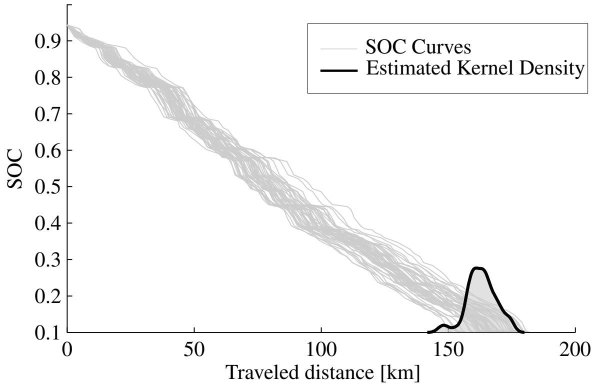


Figure 7. Predicted RDR represented by a kernel density in the city scenario.

6.1. Performance of the RDR Prediction

To evaluate the performance of the RDR prediction, the relative accuracy (RA) and the alpha-lambda ($\alpha - \lambda$) metrics are employed (Saexena, Celaya, Saha, Saha, & Goebel, 2009).

The relative accuracy measures the error in the RDR prediction relative to the true RDR. The RA is given by

$$\mathrm{RA}_{k_p} = 100 \left(1 - \frac{\left| \mathrm{RDR}^*_{k_p} - \mathrm{RDR}_{k_p} \right|}{\mathrm{RDR}^*_{k_p}} \right), \qquad (28)$$

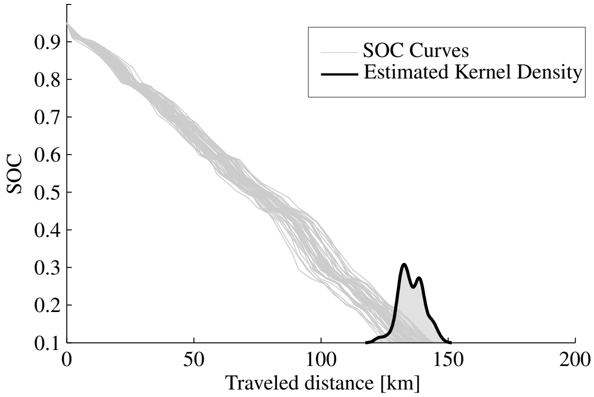


Figure 8. Predicted RDR represented by a kernel density in the rural scenario.

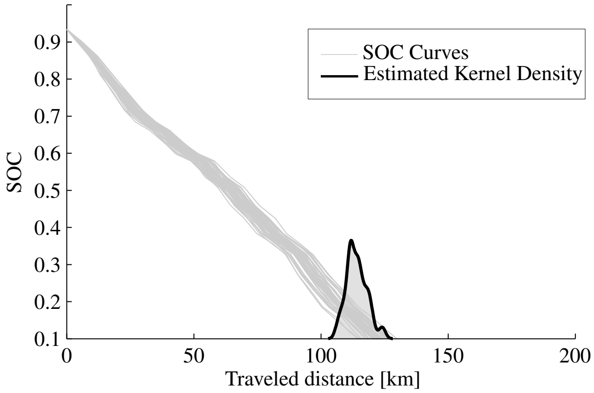


Figure 9. Predicted RDR represented by a kernel density in the highway scenario.

where $\mathrm{RDR}^*_{k_p}$ represents the ground truth RDR at time k_p and RDR_{k_p} is the predicted RDR at that time. The $\alpha - \lambda$ metric serves to evaluate whether the predicted RDR's lie withing specified bounds. These bounds are usually calculated as a fraction of the true RDR. We set a value $\alpha = 0.15$. Table 3 summarizes the calculated RA for each driving cycle at different prediction times.

As it can be seen in Figs. 10, 11 and 12, the proposed approach performs similar for each driving scenario. The simulation for the city scenario was done based on the UDDS drive cycle. The predicted RDR falls near the true RDR for the entire simulation. Also quantiles q_5 and q_{95} fall within the bounds of the α metric for most part of the simulation. The second scenario, namely the rural areas, is simulated base on the ARTEMIS rural drive cycle (Andre, 2004). The results also show an acceptable accuracy. Nevertheless, in this case the RDR is overestimated for most part of the simulation and the area between q_5 and q_{95} doesn't fall withing the bounds. The reason for this is, that the transition probability matrix used for the prediction of the driving profiles in this scenario combines information of both, the city and the highway. For this reason, the predicted profiles don't always properly

Table 3. RDR prediction performance.

k_p	RA_{UDDS}	RA_{RURAL}	RA_{HWFET}
1	96.67	88.90	98.39
3	99.17	90.81	97.01
5	99.48	91.89	97.59
7	98.64	92.22	92.65
9	98.69	91.63	96.25
11	96.48	92.40	86.01
13	98.11	99.70	42.68
15	85.47	97.41	−−
17	91.92	53.47	−−
19	73.17	−−	−−
21	90.56	−−	−−

represent this driving scenario. The third scenario is represented by the HWFET driving cycle. As it can be observed in Fig. 12, the predicted RDR's during the simulation are very close to the true values. Moreover, the uncertainty band given by the quantiles lies very tightly withing the α bounds. The deviation presented towards the end for both the rural and the highway scenario is due to the acceleration phase at the beginning of each prediction, since it always starts from the standing position. This makes the predicted energy consumption at the beginning of the prediction to be larger in comparison to the real one, where such acceleration phase is not present.

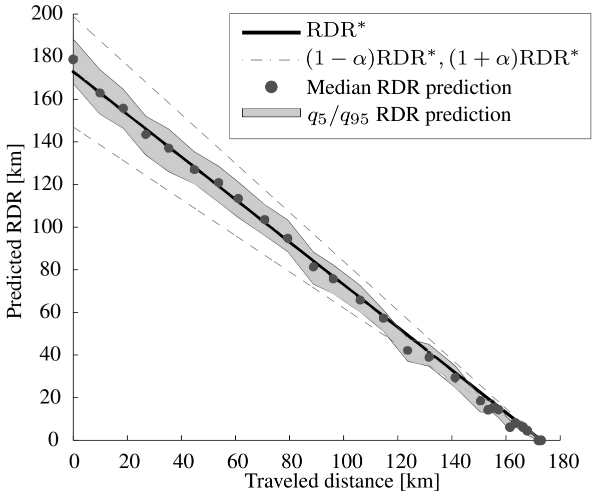


Figure 10. RDR predictions for the UDDS drive cycle.

7. Conclusions and Future Work

This work presents a model-based approach for predicting the RDR in electric vehicles. The proposed approach proceeds in two steps, namely the state estimation and the prediction step. Detailed models for determining the power demand of electric vehicles and for describing the dynamic behavior of the battery are also presented. Our approach takes into account the sources of uncertainty that influence the RDR. First, a Bayesian tracking algorithm, namely the unscented Kalman filter was implemented to estimate the SOC of the battery. In the second step, this estimate is used as the starting point for the RDR prediction. Here, the set of sigma points, obtained from the state estimation, is propagated forward in time by letting them follow multiple predicted driving profiles. A stochastic approach based on Markov chains for predicting the driving profiles is employed. The RDR is then computed as a probability density function approximated by the distribution of the propagated sigma points. The proposed approach is demonstrated and validated by means of a series of simulation experiments. The experiments allowed to predict the RDR under different driving situations. The obtained results have shown that the RDR can be accurately predicted for given scenarios with our approach. Nevertheless, the efficiency of the algorithm still largely depends on the number of evaluated driving profiles. An aspect we aim to investigate in the future is therefore to describe the driving profile parametrically. In this way the uncertainty, i.e., the probability distribution of the parameters can be incorporated in

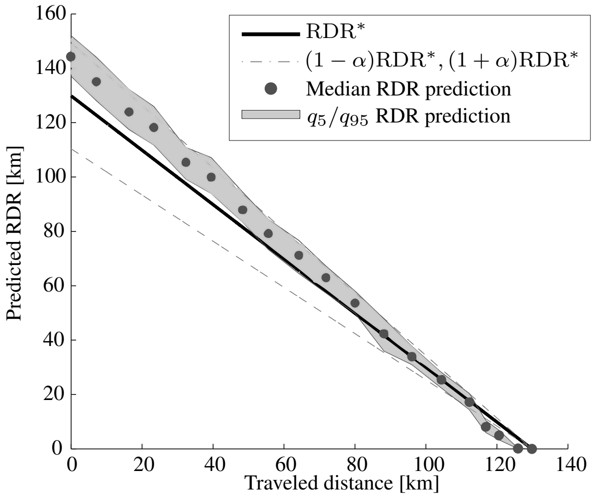


Figure 11. RDR predictions for the ARTEMIS rural drive cycle.

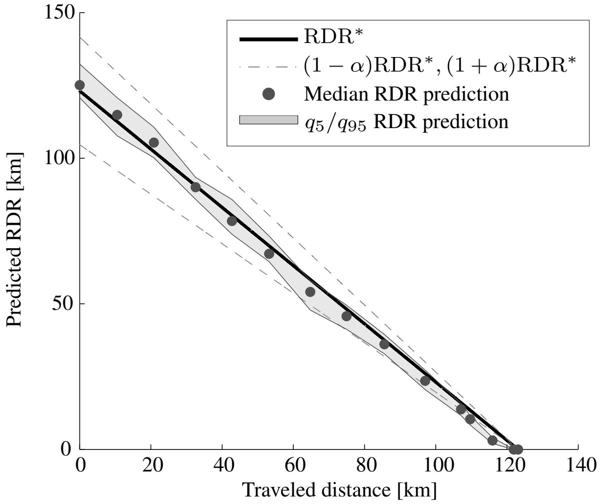


Figure 12. RDR predictions for the HWFET drive cycle.

the unscented transform. Other aspect that we plan to investigate is the adaption of the driving profile prediction to the driving style, which may increase the accuracy of the RDR prediction. To this aim it would be necessary to update $\mathbf{\Phi}^{va}$ according to new observed driving data.

Acknowledgment

The funding for this work was provided by the federal state of North Rhine-Westphalia (NRW) in frame of the project "Technology and test platform for a competence center for interoperable electromobility, infrastructure and networks" (TIE-IN).

References

Andre, M. (2004). The ARTEMIS European driving cycles for measuring car pollutant emissions. In *Science of the total environment.*

Bowman, A. W., & Azzalini, A. (Eds.). (1997). *Applied smoothing techniques for data analysis.* New York: Oxford University Press Inc.

Conradi, P., & Hanssen, S. (2011). Dynamic cruising range prediction for electric vehicles. In *Advanced microsystems for automotive applications 2011* (p. 269-277). Springer-Verlag Berlin Heidelberg.

Daigle, M., & Goebel, K. (2010). Improving computational efficiency of prediciton in model-based prognostics using the unscented transform. In *Annual conference of the prognostics and health management society 2010.*

Daigle, M., & Goebel, K. (2011). A model-based prognostics approach applied to pneumatic valves. In *International journal of prognostics and health management.*

Daigle, M., Saxena, A., & Goebel, K. (2012). An efficient deterministic approach to model-based prediction uncertainty estimation. In *Annual conference of the prognostics and health management society 2012.*

Franke, T., Neumann, I., Bühler, F., Cocron, P., & Krems, J. (2012). Experiencing range in an electric vehicle - understanding psychological barriers. In *Applied psychology: An international review* (Vol. 61, p. 368-391).

Guzzella, L., & Sciarretta, A. (Eds.). (2005). *Vehicle propulsion systems: Introduction to modeling and optimization.* Springer Verlag, Heildelberg.

Hayes, J., Oliveira, R. de, Vaughan, S., & Egan, M. (2011). Simplified electric vehicle power train models and range estimation. In *Vehicle power and propulsion conference (VPPC), 2011 IEEE* (p. 1-5).

Hoaglin, D. C., Moesteller, F., & Tukey, J. C. (Eds.). (1983). *Understanding robust and exploratory data analysis.* Wiley.

Jongerden, M., & Haverkort, B. (2009). Which battery model to use? In *Software, IET* (Vol. 15, p. 445-457).

Julier, S., & Uhlmann, J. (2004). Unscented filtering and nonlinear estimation. In *Proceedings of the IEEE.*

Kim, E., Lee, J. L., & Shin, K. G. (2013). Real-time prediction of battery power requirements for electric vehicles. In *ACM/IEEE 4th international conference on cyber-physical systems (ICCPS 13).*

Lee, T., & Filipi, Z. (2011). Representative real-world driving cycles in Midwestern US. In *Les rencontres scientifiques d'lFP energies nouvelles - RHEVE 2011.*

Lee, T. C., Judge, G. G., & Zellner, A. (Eds.). (1970). *Estimating the parameters of the markov probability model from aggregate time series data.* North-Holland, 2nd edition.

Manwell, J. F., & McGowan, J. G. (1994). Extension fo the kinetic battery model for wind-hybrid power systems. In *Proceedings of EWEC.*

Rigatos, G. (2009). Particle filtering for state estimation in nonlinear industrial systems. In *Instrumentation and measurement, IEEE transactions on.*

Saexena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009). On applying the prognostics performance metrics. In *Annual conference of the prognostics and health management society 2009.*

Yu, H., Tseng, F., & McGee, R. (2012). Driving pattern identification for EV range estimation. In *Electric vehicle conference (IEVC), 2012 IEEE international.*

Biographies

Javier A. Oliva received his B.S. degree in Mechanical Engineering from the University Landivar in Guatemala in 2006 and his M.S. degree in Automation and Robotics from the Technische Universität Dortmund in 2010. His research interests include probabilistic methods, diagnosis and prognostics applied to electric vehicles. He is currently working as researcher at the Institute of Control Theory and Systems Engineering from the TU Dortmund in the area of driver assistance systems for electric vehicles.

Christoph Weihrauch graduated in 2012 from the Ruhr-Universität Bochum with the Master's degree in Electrical Engineering. He is currently working as researcher at the Institute of Control Theory and Systems Engineering from the TU Dortmund in the area of charging systems for electric vehicles.

Torsten Bertram is Professor at the Technische Universität Dortmund and he directs the Institute of Control Theory and Systems Engineering. He has carried out applied research in the areas of drive systems, service robotics and development methodology.

A Modeling Framework for Prognostic Decision Making and its Application to UAV Mission Planning

Edward Balaban[1], Juan J. Alonso[2]

[1] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
edward.balaban@nasa.gov

[2] *Stanford University, Stanford, CA, 94305, USA*
jjalonso@stanford.edu

Abstract

The goal of prognostic decision making (PDM) is to utilize information on anticipated system health changes in selecting future actions. One of the key challenges in PDM is finding a sufficiently expressive yet compact mathematical representation of the system for use with decision optimization algorithms. In this paper we describe a general modeling approach for a class of PDM problems with non-linear system degradation processes and uncertainties in state estimation, action effects, and future operating conditions. The approach is based on continuous Partially Observable Markov Decision Processes (POMDPs) used in conjunction with 'black box' system simulations. The proposed modeling framework can be cast into simpler representations, depending on which sources of uncertainty are being included. The approach is illustrated with a mission planning case study for an unmanned aerial vehicle (UAV). In the case study a PDM system is tasked with optimizing the vehicle route after an in-flight component fault is detected. A stochastic algorithm (based on particle filtering) is used for decision optimization, with a second, deterministic algorithm providing a performance evaluation baseline. Both algorithms utilize a UAV physics simulator for generating predictions of future vehicle states. Performance benchmarking is done on a set of mission scenarios of increasing complexity.

1. Introduction

Decision-making in complex aerospace applications (our area of interest) encompasses the selection of actions at numerous levels of system abstraction. At the lower levels decision making can mean selecting controller gain values in a subsystem or defining the allowable movement range for a control device. At the mission level decision making can involve modifications to the vehicle route. At the highest levels, decision making can extend to allocating assets for a mission from a fleet of vehicles or even reorganizing a logistics chain.

Prognostic Decision Making (PDM) can be defined as the process of selecting system actions informed by predictions of the future system health state. While in many respects PDM is similar to other planning and decision optimization problems (such as those from the fields of optimal control or path planning), there are two key reasons, in our opinion, to consider it as a distinct problem type. First, incorporating information, however imperfect, about the evolution of the system health parameters may help increase decision quality and reduce the state estimation uncertainty. Doing that may prove to be a non-trivial task, as health degradation processes in aircraft or spacecraft components often have complex dependencies on operating conditions, environmental factors, and the degradation processes occurring in the other parts of the system. Second, selecting actions appropriately may, in turn, help improve subsequent prognostic estimates. In nominal operations, incorporation of prognostics into the decision-making could help optimize vehicle performance and minimize maintenance costs. In situations where the vehicle is experiencing in-flight malfunctions, having the additional source of information provided by prognostic methods could be crucial to ensuring a safe mission outcome.

In order to make the prognostic decision making problem tractable for an aerospace application, it may need to be subdivided into smaller sub-problems, with the information exchange organized for overall decision coherency (Balaban & Alonso, 2012). The goal of the work described in this paper is to further develop modeling techniques and algorithms for solving the types of sub-problems that require representation of uncertainties in state estimation (including in payoffs/rewards), action outcomes, and future operating conditions. Our modeling approach is based on continuous Partially Observable Markov Decision Processes (Bertsekas,

Edward Balaban et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1995). As a generalization of the more traditional state-space representations and Markov Decision Processes (Bellman, 1957b), the same POMDP framework can be used to describe problems with some or all of the aforementioned sources of uncertainty absent.

A mission replanning case study for an unmanned aerial vehicle is used as an illustrative example. It is extended from a mission replanning study for a ground vehicle (Balaban & Alonso, 2012). While the general ideas for the two studies are similar, additional challenges present themselves when implementing PDM for a flight vehicle. With motion now conducted in three dimensions (six degrees of freedom), the task of estimating a future vehicle state is inherently more complex. In addition, the system needs to produce its action recommendations as quickly as possible since, unlike a ground vehicle, an airplane cannot simply stop, enter a safe mode, and wait for the decision-making process to complete. The longer the process takes, the worse the fault condition may become and the further from the optimal flight path the vehicle may end up.

Exact solutions of problems formulated as POMDPs are generally not achievable (Pineau, Gordon, & Thrun, 2006), therefore some type of an approximate algorithm needs to be utilized. In this work an algorithm based on particle filtering (Gordon, Salmond, & Smith, 1993) is implemented, with a second, deterministic algorithm used for validation and performance assessment purposes. The latter algorithm is based on backtracking search (Knuth, 1968).

In the near term, the case study described in this work is meant to pave the way for PDM flight demonstrations on the Edge 540 electric UAV (Hogge, Quach, Vazquez, & Hill, 2011). The Swift electric UAV (Denney, Pai, & Habli, 2012), being developed at NASA Ames Research Center, is under consideration as an advanced follow-up test platform. The Swift is significantly larger than the Edge 540 (13.0 meter vs 2.45 meter wingspan) and can fly longer missions (7-8 hours aloft projected vs. approximately 0.3 hours for the Edge 540). The longer flight endurance is expected to allow for a greater variety of gradually developing fault modes to be introduced into the test scenarios.

The rest of the paper is organized as follows. Related research efforts are briefly reviewed in the next section. Section 3 describes our modeling methodology. The mission replanning case study is defined in Section 4. The prototype vehicle health management architecture used to implement the case study is described in Section 5. A UAV physics simulator (Section 6) is one of the components of this architecture. The simulator includes prognostic degradation models for some of the vehicle components. Section 7 presents the two decision-making algorithms (deterministic and stochastic), which are tested on a set of simulated missions of varying complexity (Section 8). Section 9 provides a summary of the work performed and outlines our plans for future research.

2. Related Work

This section highlights some of the research efforts that, we believe, form a representative sample of the current state of the art in PDM. Most of the PDM research to-date has been done for low-level component control, with a few projects using prognostics in solving higher-level planning and scheduling problems.

Pereira, Galvao, and Yoneyama (2010) developed a Model Predictive Control (MPC) system that distributes control effort among several redundant actuators based on prognostic health information. Damage accumulation is assumed to be linearly dependent on the exerted control effort. The approach is tested in simulation on a tank level regulation problem. Brown and Vachtsevanos (2011) also incorporate prognostic information into a model-predictive controller and apply it to optimizing performance (in simulation) of an electro-mechanical actuator. The work includes recommendations for error analysis and for estimation of uncertainty bounds in long-term Remaining Useful Life (RUL) predictions. Bogdanov, Chiu, Gokdere, and Vian (2006) investigate coupling of a prognostic lifetime model for servo motors with a family of LQR controllers.

Edwards, Orchard, Tang, Goebel, and Vachtsevanos (2010) propose a set of metrics to quantify the impact of input uncertainty on non-linear prognostic systems. The metrics are incorporated into a feedback correction loop in order to demonstrate RUL extension for a non-linear, non-Gaussian system (a helicopter gear plate experiencing a fatigue crack fault). An algorithm based on particle filtering is used for uncertainty estimation. The work done by Tang, Hettler, Zhang, and Decastro (2011) contains elements of prognostics-enhanced control, but also extends into prognostic path planning for an unmanned ground vehicle. RUL estimates were used either as a constraint or as an additional element in the cost function of the path-planning algorithm, with the algorithm based on Field D* search (Ferguson & Stentz, 2006). Methods for estimating and managing prediction uncertainty were also developed.

In our work we aim to build on the above efforts by exploring the benefits and the challenges of performing prognostic decision making in a more general framework, where the system model is treated as a 'black box'. Our approach to doing this is described next.

3. Modeling Approach

The mathematical modeling approach adapted for this work is meant to be general and applicable to a wide variety of systems with degradation processes. A POMDP formulation was chosen as the basis since it allows for transition costs, rewards, and action outcomes to be expressed in probabilistic terms. The main elements of a model are described below:

State space:	$S \subseteq \mathbb{R}^n$
Action space:	$A \subseteq \mathbb{R}^m$
Observations:	$Z \subseteq \mathbb{R}^p$
Transition function:	$T(s,a,s') = pdf(s' \vert s,a) : S \times A \times S \to [0,\infty)$
Observation function:	$O(z',a,s') = pdf(z' \vert s',a) : S \times A \times Z \to [0,\infty)$
Belief state:	$b(s) = pdf(s)$
Belief space:	B - the set of all belief states
Initial belief:	b_0
Belief update:	$b^{az}(s') \propto O(z',a,s') \int\limits_S T(s,a,s')b(s)ds$
Policy:	$\pi(a,b) = pdf(a \vert b) : A \times B \to [0,\infty)$, Π is the set of all policies
Costs:	$\mathcal{C} = \{c_1(s,a), ..., c_{\vert\mathcal{C}\vert}(s,a)\} \subseteq \mathbb{R}^{\vert\mathcal{C}\vert}$
Cost functions:	$c_i(s,a,c) = pdf(c \vert s,a) : S \times A \times \mathbb{R} \to [0,\infty)$, where c is a specific real value, $i \in \{1,\ldots,\vert\mathcal{C}\vert\}$
Rewards:	$\mathcal{R} = \{r_1(s), ..., r_{\vert\mathcal{R}\vert}(s)\} \subseteq \mathbb{R}^{\vert\mathcal{R}\vert}$
Reward functions:	$r_i(s,r) = pdf(r \vert s) : S \times \mathbb{R} \to [0,\infty)$, where r is a specific real value, $i \in \{1,\ldots,\vert\mathcal{R}\vert\}$
Features:	$\mathcal{X} = \{x_1(s),\ldots,x_{\vert\mathcal{X}\vert}(s)\} : S \to \mathbb{R}^{\vert\mathcal{X}\vert}$
Constraints:	$\mathcal{G} = \{g_1(s),\ldots,g_{\vert\mathcal{G}\vert}(s)\} : S \to \mathbb{B}^{\vert\mathcal{G}\vert}$
Objective function:	$J^{\pi}(b_0) : \Pi \times B \to \mathbb{R}$

System states in S are not defined explicitly, but rather obtained through a 'black box' simulator, such as the one described in Section 6. The simulator takes an estimate of the current state and the desired action as inputs and generates a state estimate for the next time step as the output. Compared to discretized state representations, this approach helps to avoid the 'curse of dimensionality' (Bellman, 1957a), resulting from the exponential growth in the size of the state space with the number of state vector components. It also mitigates discretization errors. While a discretized formulation does not require a system simulator, we believe that employing such a formulation would quickly become impractical for all but the simplest PDM problems.

The concept of a continuously valued action space A allows us, in conjunction with the simulator, to represent not only the system actions, but the loads and environmental conditions imposed onto the system as well. Observations Z most often come from sensor readings, but can also be manual measurements of a quantity such as the length of a crack. The transition probability function $T(s,a,s')$ allows us to model the uncertainty in estimating the future loading conditions and their effects on the system state (s denotes the current state, a the action taken in the current state, and s' is the next state). The observation function $O(z',a,s')$ describes the probability of seeing an observation z' if a state s' is achieved as a result of an action a.

Since in real life it is often not possible to know the exact state the system is in, the belief state $b(s)$ is defined as the probability density of being in a particular state s. A policy $\pi(a,b)$ is then defined as the probability density of taking a specific action a given a belief state b. In practical terms, a policy allows us to define what actions the system should take given the belief distribution over possible states.

A cost is defined probabilistically as a value dependent on the current system state and the action taken in it. A cost can be the amount of energy needed to transition to the next system state or the corresponding measure of component wear. Rewards, on the other hand, depend only on the system state achieved. Reward functions define the desirability of a state as a probability density over the real domain. Features are also functions defined over state variables that enable reasoning over the various properties of a state. Unlike rewards, feature values are not meant to be used in an additive manner directly and are thus defined as point estimates.

Constraints are defined as inequalities of the form $g_i(s) \geq 0$ on the components of the state of the system. Strict equality constraints can be defined as well, although they are less useful in the current context. Constraints can be used to define regions of nominal and off-nominal system behavior, as will be done shortly for system health.

Finally, features, costs, and rewards are combined into an objective function J^{π} that, given an initial belief state b_0 and a policy π, can compute a single evaluation metric of that policy. Several objective functions can be defined for multi-criteria reasoning. Given the predominantly non-negative transition costs in a typical PDM application, a separate discount factor (often used in value and objective functions to guarantee convergence and encourage shorter solutions) is likely unnecessary. Negative costs correspond to self-healing or commanded recovery events and, while certainly valid, should not dominate the non-negative costs (e.g. energy consumption or component wear), given a properly designed objective function. Terminal states are defined through the constraints in $\mathcal{G}$, including (but not limited to) constraints on vehicle health parameters or position.

Now that the modeling framework has been described, the key system health management concepts can be defined within it:

- **Health parameters**: $H = \{h_1, \ldots, h_H\}$ is a subset of state vector elements that describes the health state of the system.
- **Fault:** A subset of constraints $\mathcal{G}_{\textit{fault}} \in \mathcal{G}$ defines significant deviations from the expected nominal behavior (with respect to vehicle health). A fault occurs if $\exists i, g_i(s) = false, g_i \in \mathcal{G}_{\textit{fault}}$. At least some of the health parameters H are expected to be included in the constraints of $\mathcal{G}_{\textit{fault}}$.
- **Failure:** Another subset $\mathcal{G}_{\textit{failure}} \in \mathcal{G}$ defines states where the system loses its functional capability with respect to a health parameter $h \in H$.
- **System failure:** System failure is defined through a boolean function $\mathcal{F} : B \to \mathbb{B}$, which indicates when the entire system is believed to be effectively non-functional ($\mathcal{F}$ is defined via the $\mathcal{G}_{\textit{failure}}$ set). System failure can be indicated when a single $g_i \in \mathcal{G}_{\textit{failure}}$ is believed to be violated, for example, or a larger subset of them. More generally, the function could also be defined as $\mathcal{F} : B \to [0, 1]$, mapping the belief space to a probability of system failure.
- **End of Life (EoL)**: End of Life in this set of definitions is synonymous with system failure. Time of EoL is defined similarly to (Daigle, Bregon, & Roychoudhury, 2012) as $t_{EoL} \triangleq \inf\{t \in \mathbb{R} : (t \geq t_p) \wedge (\mathcal{F}(b) = true)\}$, where t_p is the time of prediction (or, alternatively, the present time).
- **Remaining Useful Life (RUL)**: $RUL = t_{EoL} - t_p$

The *diagnostic* problem then becomes the process of determining the current belief state, b_t. The *prognostic* problem can be stated as the process of determining, at time t, the belief state $b_{(t+\Delta t)}$, given the current policy π (with $\Delta t > 0$). The *prognostic decision making* problem is then the process of finding (or approximating) π^*, such that

$$\pi^* = \arg\max_{\pi \in \Pi} J^{\pi}(b_t),$$

where $J^{\pi}(b_t)$ is an objective function computed using a policy π given a state of belief b_t at time t. Time (or an alternative index) can be discretized.

Given this general framework for describing the system, its actions, and the state of system health, we can recast it depending upon the problem at hand. While in some circumstances representation and quantification of uncertainty is necessary to increase the decision accuracy, in others the downsides of having a random process present in the reasoning system outweigh the benefits. When in real-time use aboard an air vehicle, for example, running the simulation in the deterministic mode (supplying the expected value of the next state) may often be preferable. For the purposes of development and verification, it may also be beneficial to start with a deterministic state transition system. Enabling uncertainty in the effects of an action would turn the system into a continuous Markov Decision Process (MDP) (Bertsekas, 1995). Adding uncertainty in state estimation would turn it into a continuous POMDP.

While exact solutions for problems posed as deterministic state-space representations can be produced using a variety of search techniques, as the amount of uncertainty increases computing exact solutions becomes more and more challenging. In certain cases (linear system model, quadratic objective function) continuous MDP problems can be solved exactly using Differential Dynamic Programming (Jacobson & Mayne, 1970). In other cases approximate methods are required. The two algorithms described in Section 7, backtracking search and Particle Filter, can be used with both deterministic and stochastic next-state representations. The former will produce an exact solution if the states are defined deterministically, while the latter will generate approximate solutions in both cases.

4. Mission Replanning Case Study

Our case study is based on a UAV mission scenario where the vehicle is tasked with visiting a set of waypoints and returning back to the point of origin. Such a mission profile is typical of reconnaissance or geophysical survey missions, for example. For tasks of this type the order of visiting the waypoints is often less important than which waypoints were actually reached. Further scenario details are listed below.

Given:

- The waypoint set is defined as $wp = \{wp_i\}_{i=1}^{N}$, where a waypoint $wp_i = \{(x_i, y_i, z_i), \mathcal{G}_i, r_i\}$ is specified by its Cartesian coordinate vector (x_i, y_i, z_i), waypoint-specific constraints subset $\mathcal{G}_i$ (e.g. on airspeed or bank angle), and a reward value r_i.
- A path $p = (wp_j)_{j=1}^{M}$ is defined as an ordered subset of wp (i.e. $M \leq N$). P is the set of all possible paths.
- The aircraft starts its mission at waypoint wp_1 (home runway) and is required to return there for a mission to be considered a full or a partial success.
- An initial path p_0 is specified (p_0 is not necessarily optimized).
- $h_h \in [0, 1]$ is the normalized system health index (1 is full health and 0 indicates failure). The constraint on system health is defined as $g_h(s) = h_h \geq 0$.
- $h_e \in [0, 1]$ is the normalized remaining energy index (1 is full charge and 0 indicates depleted energy). The constraint on energy is defined as $g_e(s) = h_e \geq 0$.
- A healthy vehicle is able to complete p_0 within the energy and component health constraints (before either reaches 0).
- Energy and health transition costs between a pair of waypoints a and b are defined as $c_e(wp_a, wp_b)$

and $c_h(wp_a, wp_b)$, respectively (the costs are history-dependent). $\mathcal{C}_e(p)$ and $\mathcal{C}_h(p)$ are the corresponding cumulative costs for a path. Similarly, the cumulative path reward is denoted by $\mathcal{R}(p)$.

- A fault f occurs at a time t_f, that makes it impossible to complete p_0 before EoL. The fault magnitude changes with time, loading conditions, and environmental factors.

Find:

$$p^* = \arg\max_{p \in P} \mathcal{R}(p),$$

where p^* is a path that maximizes cumulative reward.

A waypoint wp_i is considered to be reached if the vehicle achieves a position $(x_i \pm \epsilon_x, y_i \pm \epsilon_y, z_i \pm \epsilon_z)$, with all of the applicable constraints satisfied. Here ϵ_x, ϵ_y, and ϵ_z are the predefined maximum position error values.

5. PHM Reasoning Architecture

The reasoning architecture used in this work consists of the four main components: the diagnoser (DX, for diagnostics), the decision maker (DM), the vehicle simulator (VS), and the vehicle itself (Figure 1).

The execution process is initiated with an input route p_0 and an initial fault set F_0 supplied to the decision maker. This can be done at the beginning of a mission (in that case the fault set may be empty) or if a fault is diagnosed in flight. A fault set F is one or more fault descriptors. A fault descriptor f_i consists of a fault type d and a fault magnitude m. Fault magnitude m is generally normalized to the interval $[0, 1]$.

DM utilizes VS to evaluate and, optionally, optimize the input route (according to the criteria in Section 4). An initial optimization can be performed if a successful completion of the input route is deemed unlikely. Alternative routes sent to VS consist of an ordered set of waypoints (with vectors Γ of specific values for waypoint parameters). DM also informs VS of the relevant fault modes. VS simulates the candidate path, p_c, and returns the reward and cost estimates for it. Once DM finalizes the route recommendation p^*, it is sent to the vehicle for execution.

As the mission proceeds, DX continues to monitor observations Z (e.g. sensor readings) generated by the vehicle to detect any new fault conditions. The current prototype implementation, meant to primarily test the PDM algorithms, includes only a minimal diagnostic functionality. Future versions will incorporate a fully-featured diagnoser, such as HyDE (Narasimhan & Brownston, 2007) or QED (Daigle & Roychoudhury, 2010). One of our long-term goals is to unify diagnostic and decision making processes around a single, comprehensive system model. For the near future, however, if the diagnoser utilized is model-based (as is the case for both HyDE and QED), a separate diagnostic model would need to be constructed. A new fault condition triggers a reevaluation of the vehicle route by DM. Another event type that triggers a reevaluation is a significant deviation of the predicted component degradation rates from the rates observed as the flight progresses further. This mechanism is implemented by storing the predicted component degradation curves for the proposed route (p^*) before it is sent to the vehicle for execution.

6. Vehicle Simulator

Given an estimate of the current vehicle state and the desired action, the simulator is used to generate an estimate of the next state in its innermost loop. The state vector includes the aircraft position, velocity, acceleration, orientation, lift, drag, thrust, battery voltage, battery charge remaining, component temperatures, and other data. When an entire path p is provided, the simulator can generate reward and cost estimates $\mathcal{R}(p)$, $\mathcal{C}_h(p)$, and $\mathcal{C}_e(p)$. For that, the route is divided into segments where the key elements of vehicle dynamics can be considered constant (i.e. acceleration, bank angle, angle of climb/descent, etc). The segments are further divided into time steps. The size of a time step, dt, can be specified as one of the simulation parameters (it is adjusted automatically for segments where higher precision in position control is required, such as during take-offs and landings). As the simulator is primarily intended to provide estimates of transition costs and rewards for DM in a computationally efficient manner, it does not employ a closed loop controller to achieve precise trajectory following.

The aerodynamic forces are calculated in a right-handed, velocity-oriented, local-level frame of reference, with the origin at the center of mass of the aircraft. The position of the center of mass $\{x(t), y(t), z(t)\}$ is computed in an inertial right-handed Cartesian frame of reference, with the origin at the take-off/touch-down end of the runway. There are currently three elements of the vehicle model with non-linear, action-dependent degradation aspects: battery voltage, battery temperature, and motor temperature (subsections 6.5 and 6.6). The simulator is implemented in MATLAB (*MATLAB version 7.11.0.584 (2010b)*, 2010). The rest of the section provides further details of the simulator, describing the key equations and assumptions.

6.1. Lift and drag

Lift L and drag D are calculated as

$$L = \frac{1}{2}\rho_\infty v_\infty^2 S C_L, \tag{1}$$

$$D = \frac{1}{2}\rho_\infty v_\infty^2 S C_D, \tag{2}$$

$$C_D = C_{D,p} + C_{D,i}, \tag{3}$$

where L is lift, ρ_∞ is the freestream density, v_∞ is the freestream velocity, $C_{D,i}$is the coefficient of induced drag,

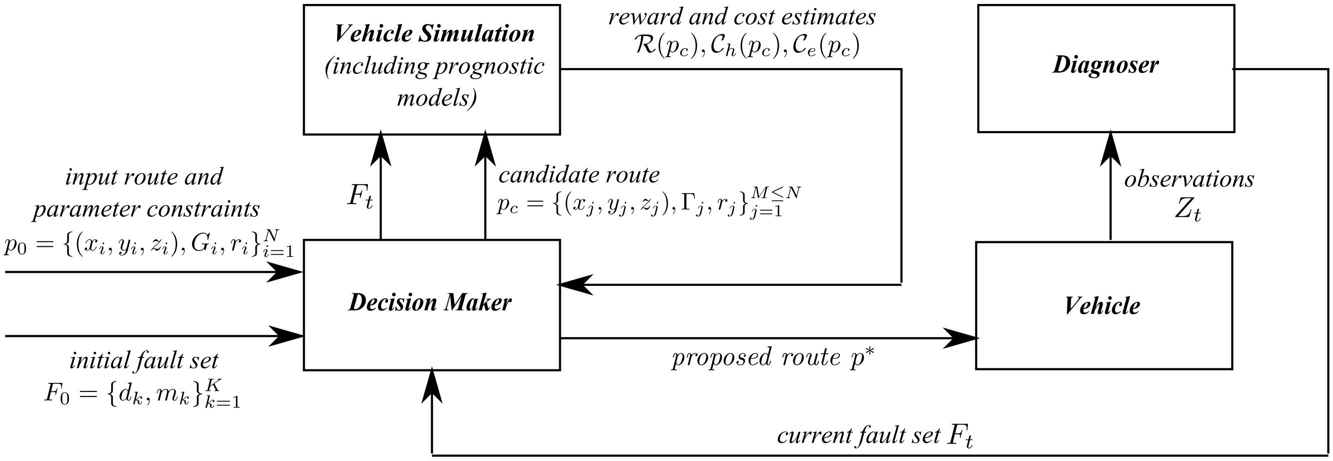

Vehicle Simulation (including prognostic models)
reward and cost estimates
$\mathcal{R}(p_c), \mathcal{C}_h(p_c), \mathcal{C}_e(p_c)$
Diagnoser
input route and parameter constraints
$p_0 = \{(x_i, y_i, z_i), G_i, r_i\}_{i=1}^{N}$
F_t
candidate route
$p_c = \{(x_j, y_j, z_j), \Gamma_j, r_j\}_{j=1}^{M \leq N}$
observations
Z_t
Decision Maker
Vehicle
initial fault set
$F_0 = \{d_k, m_k\}_{k=1}^{K}$
proposed route p^*
current fault set F_t

Figure 1. Reasoning architecture

and $C_{D,p}$ is the coefficient of profile drag. Using Prandtl's Lifting Line theory:

$$C_{D,i} = \frac{C_l^2}{\pi e AR}, \tag{4}$$

where AR is the planform aspect ratio and e is the span efficiency factor. From (3) and (4):

$$C_D = C_{D,p} + \frac{C_l^2}{\pi e AR}, \tag{5}$$

$C_{D,p}(\alpha, Re)$ is obtained from experimental wind tunnel data for a similar airfoil (Miller, 2008; Lewis, 1984), with α denoting the angle of attack (in radians) and Re the Reynold's number. The thin airfoil assumption is made, thus

$$C_l = 2\pi\alpha. \tag{6}$$

6.2. Equations of motion for straight, accelerated flight

Force balance equations in x and z directions give

$$\ddot{x} = \frac{1}{m}(T - D) - g \sin\gamma, \tag{7}$$

$$\ddot{z} = \frac{1}{m}L - g \cos\gamma, \tag{8}$$

where γ is the flight path angle, g is the acceleration of gravity, and m is the mass of the aircraft.

6.3. Equations of motion for turning flight

Assume a constant airspeed (i.e. $dv/dt = 0$, v here equivalent to the freestream velocity), balanced flight (no skidding or slipping) during turns. From the equation of equilibrium in the horizontal direction:

$$F_c = L \sin\phi \cos\gamma = \frac{v^2}{r} m, \tag{9}$$

where ϕ is the bank angle, r is the turn radius, and F_c is the centrifugal force. Then

$$r = \frac{v^2}{L \sin\phi \cos\gamma} m. \tag{10}$$

From the vertical equation of equilibrium, the lift required is now

$$L = mg \frac{\cos\gamma}{\cos\phi}, \tag{11}$$

Considering the trajectory of a generalized climbing (or descending) turn to be a helix, the x, y, z coordinates are parametrized as follows:

$$x(t) = r \cos(t), \tag{12}$$

$$y(t) = r \sin(t), \tag{13}$$

$$z(t) = 2r \sin(\gamma) t. \tag{14}$$

6.4. Power and current

The power P required to generate the desired thrust T is expressed as

$$P = \frac{1}{\eta_p} T v \text{ and} \tag{15}$$

$$P = \eta_m E \frac{dq}{dt}, \tag{16}$$

where q is the battery charge required, E is the battery voltage, and η_p and η_m are the propeller and motor efficiency coefficients, respectively. Charge q can then be expressed as

$$dq = \frac{1}{\eta_t \eta_m \eta_p} \frac{Tv}{E} \, dt, \tag{17}$$

The electrical transmission efficiency coefficient is denoted by η_t. Propeller efficiency for a specific airspeed is currently approximated as a quadratic function, with the maximum efficiency achieved at cruise airspeed v_c is

$$\eta_p = \eta_{p,max}\left(-\left(\frac{v}{v_c}\right)^2 + 2\frac{v}{v_c}\right). \quad (18)$$

In the future versions of the simulator equation 18 will be replaced with experimentally-derived curves for the propellers used.

6.5. Battery charge and voltage at the terminals

The battery model is adapted from the work described by Daigle, Saxena, and Goebel (2012), which, in turn, is based on battery models by Barsali and Ceraolo (2002), M. Chen and Rincon-Mora (2006), and Saha, Quach, and Goebel (2012). Three main processes are captured in the model: the *ohmic drop* (also known as the *I-R drop*), the *parasitic resistance* (accounting for self-discharge), and the *concentration polarization.* Out of the three, the concentration polarization resistance is the primary contributor to the non-linearity of the battery output voltage as a function of its state of charge. The state of charge (SoC) is defined as

$$SoC = 1 - \frac{q_{max} - q_b}{C_{max}}, \quad (19)$$

where q_b is the current charge in the battery (assumed to be held by capacitance C_b), q_{max} is the maximum possible charge, and C_{max} is the maximum possible capacity. The concentration polarization resistance is expressed as:

$$R_{CP} = R_{CP0} + R_{CP1}\exp\left(R_{CP2}(1 - SoC)\right), \quad (20)$$

where R_{CP0}, R_{CP1}, and R_{CP2} are empirical parameters. The resistance (and, consequently, the voltage drop at the battery terminals) increases exponentially as SoC decreases (Saha et al., 2012). End of discharge is considered to have occurred when the voltage drops below a predefined threshold. Further details of the model can be found in (Daigle, Saxena, & Goebel, 2012).

6.6. Battery and motor temperatures

Being able to predict battery temperatures is important for the following reasons: (a) temperature influences the internal battery resistance (although this effect is, at present, not included into the model) and (b) excessive temperatures can lead to premature capacity degradation and, beyond a certain point, to thermal runaway and battery failure (Y. Chen, Song, & Evans, 1996). In an electric motor overheating can also result in failure due to stator winding insulation damage or rotor magnet delamination (Milanfar & Lang, 1996). Battery and motor temperatures are currently estimated with the following simple model:

$$dT = \frac{1}{C_t}(RI^2 + h(T_a - T))dt, \quad (21)$$

where R is the electrical resistance of the component, C_t is the thermal inertia coefficient, h is the thermal transfer coefficient, I is the current, T is the component temperature, and T_a is the ambient temperature.

The thermal transfer coefficient h is estimated using the assumption that the motor and battery cooling systems are designed to keep these components operating at some nominal temperatures T_o in straight and level cruise flight, with I_c current being drawn from the batteries. Given these conditions, $dT \approx 0$, thus

$$h = \frac{RI_c^2}{T_o - T_s}, \quad (22)$$

where T_s is the standard ambient temperature for which the cooling system was calibrated. C_t is estimated empirically.

7. ALGORITHMS

For the case study described in Section 4, the model is recast as a constraint-satisfaction problem, with states under consideration concentrated in multi-dimensional clusters representing the waypoint coordinate vicinities, acceptable airspeed and bank angle ranges, possible system health conditions, etc.

7.1. Backtracking Search

Backtracking Search (BT) is a recursive, depth-first algorithm that, in our case, attempts to sequentially build up a path p from a set of waypoints $\{wp_i\}_{i=1}^N$ until one or more of the constraints in $\mathcal{G}$ are violated (Algorithm 1). When that happens, the algorithm will back up to the last known nominal state and attempt to build the rest of the path through a different set of waypoints. The algorithm keeps track of the best objective function value found, J^*, and the path that produced it, p^*, returning them after the search is complete (for this algorithm $J = \mathcal{R}$). If the system simulator is used in the deterministic mode, then the algorithm is guaranteed to find the optimal solution (or solutions). For the purposes of benchmarking, the deterministic mode is used, with specific values assigned to waypoint parameters. With this the worst case computational complexity is $O(N!)$, where N is the number of waypoints. Unlike an exhaustive search algorithm, however, BT is capable of skipping regions of search space where constraints are violated, which, in practice, results in a significant execution speed up.

7.2. Particle Filter

Particle Filter algorithms (also known as Sequential Monte Carlo algorithms) are a family of non-Gaussian/non-linear methods for approximating posterior distributions in partially observable, controllable Markov Chains (where time is discretized). While often used for system state estimation and prediction (Orchard, 2007), the general approach is suitable for other appropriately formulated problems and in our case is applied to vehicle path selection.

Algorithm 1 BT

```
inputs: $p, p^*, J^*, \{wp_i\}_{i=1}^N$
outputs: $p^*, J^*$
if $|p| > N$ then
    return
end if
$\{b, \mathcal{R}, \mathcal{C}_h, \mathcal{C}_p\} \leftarrow$ simulate($p_{test}$)
if $\mathcal{F}(b) = false$ then
    $J \leftarrow R$,
    if $J > J^*$ then
        $J^* \leftarrow J$
        $p^* \leftarrow p$
    else
        return
    end if
end if
                        ▷ recurse through the remaining waypoints
for $i \leftarrow 1, N$ do
    if $wp_i \notin p$ then
        BT($b, \{p, wp_i\}, p^*, \{wp_i\}_{i=1}^N$)
    end if
end for
```

The PF algorithm (Algorithm 2) is initialized with a set of k particles, each particle p_i containing the starting waypoint wp_1 and assigned the weight of $w_i = 1/k$. During each of the iterations of the algorithm (and for each particle), the path associated with a particle is sampled randomly out of the set of unvisited waypoints up to the maximum length of N. Each sample is tested in the simulator and the particle weight updated proportionally to the objective function value (now incorporating costs in addition to rewards). Unless system failure is believed to be likely for even the shortest path extensions, the particle path is extended by one waypoint (the first one in the randomized remaining waypoints set τ).

The number of algorithm iterations, D, is equal to N for the deterministic simulator mode and can be set to $D > N$ otherwise, to help prevent potentially promising particles from being ruled out too early. The highest weight particle is identified and stored after each iteration, to enable interruptibility. Particle weights are then normalized and the particles are resampled. The overall computational complexity of the algorithm is $O(N^2)$.

8. Experiments

For the experiments described, the physics simulator (Section 6) was used with the parameters for the Edge 540 electric UAV (Figure 2). The Edge 540 is 2.44 m long, with a 2.54 m wingspan (Hogge et al., 2011). It is powered by two electric motors connected in series through a single drive shaft. The motors drive a 0.66 m two-bladed propeller. The current for the motors is supplied by a set of four Li-Poly rechargeable batteries, which can store a total of 43200 coulomb. The average flight time is approximately 20 minutes.

Algorithm 2 PF

```
inputs: $\{wp_i\}_{i=1}^N, K$
outputs: $p^*$
$p_1, \ldots, p_K \leftarrow \{wp_1\}$
$w_1, \ldots, w_K \leftarrow 1/k$
for $d \leftarrow 1, D$ do
    for $k \leftarrow 1, K$ do
        $\tau \leftarrow$ permute($\{wp_i\}_{i=1}^N - p_k$)
        $l \leftarrow -1$
        repeat
            $l \leftarrow l + 1$
            $p_{test} = \{p_k, \{wp_1, \ldots, wp_l\}\}$
            $\{b, \mathcal{R}, \mathcal{C}_h, \mathcal{C}_p\} \leftarrow$ simulate($p_{test}$)
            $w_k \leftarrow \Theta^T \cdot \{\mathcal{R}, -\mathcal{C}_h, -\mathcal{C}_p\}$,
        until $\mathcal{F}(b) = true$
        if $l \geq 1$ then
            $p_k \leftarrow \{p_k, \{wp_1\}_\tau\}$
        end if
    end for
    $j \leftarrow \arg\max_j w_j$
    $p^* \leftarrow p_j$
    $\{w_1, ..., w_K\} \leftarrow \{w_1, ..., w_K\} / \sum_{i=1}^K w_i$
    $\{p_1, ..., p_K\} \leftarrow resample(\{p_1, ..., p_K\}, \{w_1, ..., w_K\})$
end for
```

8.1. Setup

A set W of ten sequentially numbered waypoints is created, with each waypoint associated with a reward value (Table 1 and Figure 3). Test scenarios with progressively increasing numbers of waypoints are then created (the 7-waypoint scenario contains waypoints 1 through 7, the 8-waypoint scenario contains waypoints 1 through 8, and so on). As the UAV transitions between waypoints 2 and 3 (in the original, unoptimized order), a fault is injected into one of the motors, m_2. The motor loses power, however the rotor can still spin. The fault has the following consequences on the aircraft performance:

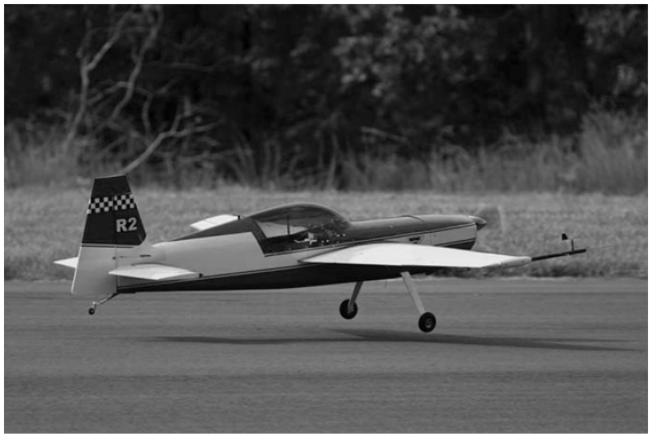


Figure 2. Edge 540 UAV (courtesy of NASA LARC)

- The available maximum thrust is reduced;
- There is now a parasitic mechanical load on the remaining motor, m_1, which not only has to provide propulsion, but also spin the rotor of m_2 (there is no disconnect mechanism);
- As a consequence, m_1 draws more current in order to execute the intended flight path;
- Increase in current results in heat build-up inside the motor housing which is not sufficiently dissipated by normal cooling mechanisms;
- The heat, if allowed to build up, will eventually raise the internal temperature of m_1 to the level where the motor windings insulation begins to melt and a short circuit can occur, irreversibly damaging the motor (assumed to be $70°C$, as measured on the motor housing);
- The increased amount of current required to drive the remaining motor also means a higher rate of battery discharge and a higher rate of heat build-up inside the battery;
- Reducing the airspeed to decrease the motor current would increase the traverse time and, below a certain threshold, result in an aerodynamic stall;

The fault is injected by changing the motor efficiency coefficient η_{m_1}(nominally 0.85, reduced to 0.4). The waypoints are selected in such a way as to make it impossible for the UAV to visit all of them in the original order before either energy depletion or vehicle health deterioration beyond the point of failure. DM is then expected to rearrange and/or reduce the set of waypoints to maximize the mission payoff (i.e. find an optimal path), while remaining within the constraints on energy consumption and vehicle health degradation:

$$p^* = \arg\max_{p \in P} \mathcal{R}(p),\ \textit{s.t.}\ \mathcal{C}_e(p) \geq 0, \mathcal{C}_h(p) \geq 0$$

The waypoint ordering matters because load profiles associated with the different routes may affect the degrading component and the energy consumption differently. Scenarios with greater numbers of waypoints offer the algorithms more choices on how to maximize mission payoff, but with the choices comes an increase in computational time.

Note that for this case study alternative strategies of handling the fault (such as switching between powered and gliding flight to let the motor cool down) are not considered.

Experiments were conducted by running the two algorithms on the same set of scenarios and recording the reward values achieved and the number of simulation calls made. BT search was executed once for each scenario, while PF was run 30 times per scenario. The results for the latter were averaged and the standard deviation of them was computed. The vehicle simulator was used in the deterministic mode, in order to enable comparison between the two algorithms.

Table 1. Waypoint parameters

	$x(m)$	$y(m)$	$z(m)$	$V(m/s)$	$\phi(deg)$	r
1	0	0	0	30	20	41
2	500	2500	450	30	10	52
3	4500	4200	900	35	20	60
4	3800	1440	550	35	10	71
5	500	-1500	750	45	25	39
6	-2300	-500	850	40	30	46
7	1100	4000	400	30	20	33
8	1800	-700	700	30	25	95
9	1200	6200	500	25	30	85
10	150	5200	600	40	15	30

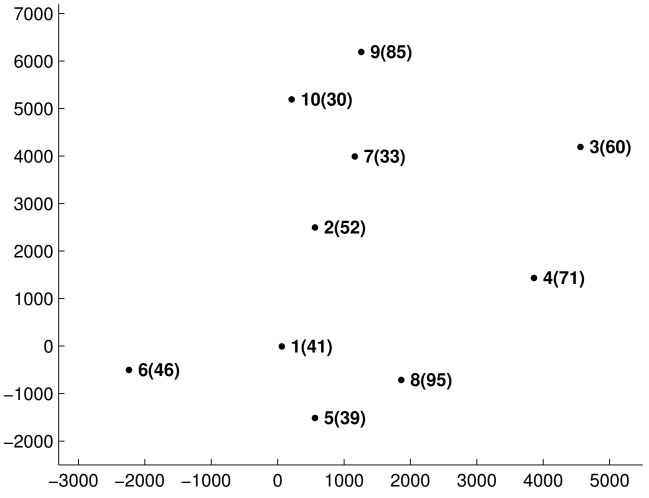


Figure 3. Waypoint locations and reward values

8.2. Results

In the result tables below (Tables 2-5), $\mathcal{R}$ is the path reward and n is the number of simulator calls made during a particular scenario. The number of simulator calls, rather than the actual algorithm runtime, is used as a platform-independent metric[*]. In the case of PF, mean and standard deviation values are provided for $\mathcal{R}$ and n. Ratios $\mathcal{R}/n$ and $\mu_\mathcal{R}/\mu_n$ are used as metrics of algorithm efficiency.

Backtracking Search

As can be seen in Table 2, BT search starts becoming impractical for the more complex scenarios, as the number of simulation calls grows exponentially. Still, the results serve as a useful evaluation baseline for PF, providing the maximum reward values achievable for a scenario.

[*] As a point of reference, a simulator call during a 10-waypoint scenario ($dt = 5$ seconds) typically took on the order of $0.005 - 0.010$ seconds to complete on a system with an Intel i7-2620M dual core CPU (2.70 GHz), with 8 Gb of RAM, running Windows 7 Professional (64-bit).

Table 2. Backtracking search results

scenario	$\mathcal{R}$	n	$\mathcal{R}/n$
7 *waypoints*	236	57	4.14
8 *waypoints*	298	241	1.23
9 *waypoints*	383	1598	0.24
10 *waypoints*	413	10117	0.04

Particle Filter

Since PF is stochastic, the algorithm was executed 30 times for each scenario and each K (number of particles) setting. The results were then averaged. Three K settings were used: 10, 20, and 30 (tables 3, 4, and 5, respectively). The average reward performance of the algorithm (and its standard deviation) steadily improve as the number of particles is increased (second column). The performance increase obviously comes at an additional computational cost (third column).

Table 3. Particle Filter results (K=10)

scenario	$\mu_\mathcal{R}(\sigma_\mathcal{R})$	$\mu_n(\sigma_n)$	$\mu_\mathcal{R}/\mu_n$
7 *waypoints*	230.2(12.8)	145.6(8.4)	1.58
8 *waypoints*	285.5(17.4)	199.2(13.8)	1.43
9 *waypoints*	368.6(21.7)	292.0(18.7)	1.26
10 *waypoints*	382.9(39.9)	331.2(20.5)	1.16

Table 4. Particle Filter results (K=20)

scenario	$\mu_\mathcal{R}(\sigma_\mathcal{R})$	$\mu_n(\sigma_n)$	$\mu_\mathcal{R}/\mu_n$
7 *waypoints*	232.4(2.5)	288.4(10.2)	0.80
8 *waypoints*	291.3(8.3)	391.2(15.8)	0.74
9 *waypoints*	372.3(14.6)	585.9(20.2)	0.63
10 *waypoints*	393.2(21.5)	670.2(29.6)	0.59

Table 5. Particle Filter results (K=30)

scenario	$\mu_\mathcal{R}(\sigma_\mathcal{R})$	$\mu_n(\sigma_n)$	$\mu_\mathcal{R}/\mu_n$
7 *waypoints*	234.0(1.1)	440.7(11.5)	0.53
8 *waypoints*	293.8(5.7)	603.8(21.6)	0.49
9 *waypoints*	375.2(10.2)	855.4(35.8)	0.44
10 *waypoints*	399.3(15.6)	1012.6(60.1)	0.39

8.3. Analysis

A comparison of algorithm performance with respect to reward values is provided in Figure 4. The dashed lines represent the reward benchmark set by BT, the dots depict the average PF rewards, and the error bars show the corresponding standard deviations. The algorithm, in general, performed well in approximating the results of the exact BT algorithm, but at a fraction of its number of simulation calls during the more complex scenarios. Even with $K = 10$ PF on average achieved over 92% of the maximum possible rewards.

Figure 5 shows a comparison with respect to the number of simulation calls. The dashed lines, again, correspond to the BT performance benchmark, the dots are the average numbers of calls for the PF algorithm, and the error bars show the standard deviations of the latter.

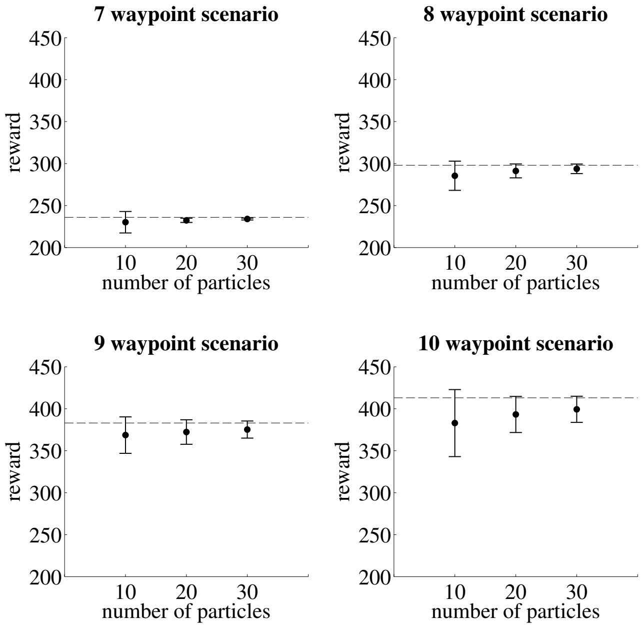


Figure 4. Reward value comparison

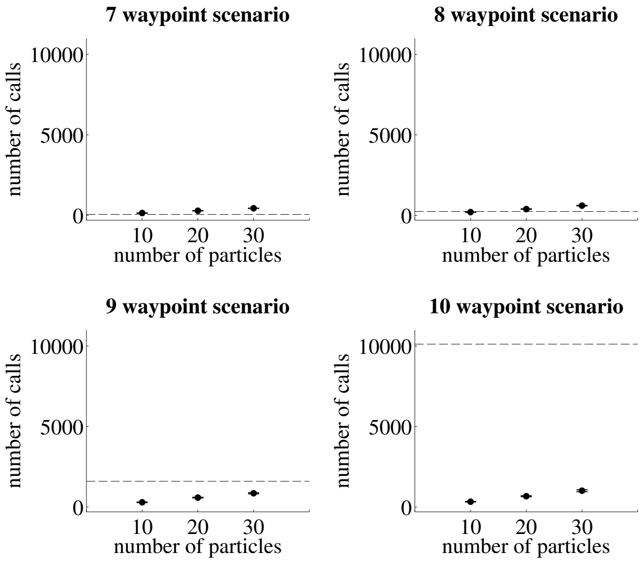


Figure 5. Number of simulation calls comparison

Even when using a fairly coarse 5-second time step in the simulator, executing BT on scenarios larger than ten waypoints became impractical. PF, on the other hand, showed the potential to be suitable for far more complex scenarios (informal tests performed on 25-waypoint scenarios still resulted in acceptable execution times, although the quality of the solutions could not be independently verified at this time).

9. Concluding Remarks

The main goal of this work is to describe a modeling framework suitable for a class of aerospace prognostic decision making (PDM) problems with complex dynamics, non-linear degradation processes, and uncertainties present in state estimation, action outcomes, and future operating conditions. Continuous Partially Observable Markov Decision Processes (POMDPs) are chosen as the foundation for the framework, with the paper providing the details on how the concepts needed for implementation of PDM are represented within it. The framework can be recast (simplified) depending on which sources of uncertainty are desired to be included. A case study serving as an illustration of the approach involves replanning a UAV mission after an in-flight fault is detected. A prototype PDM system takes into account the dependency of energy consumption and component degradation rates on the route chosen and attempts to maximize the value of the mission, while prioritizing the safe return of the aircraft. The system is demonstrated with a deterministic decision-making algorithm based on backtracking search (BT) and a stochastic algorithm based on particle filtering (PF). While likely too computationally expensive for practical applications, the BT algorithm is used to establish a performance baseline. The PF algorithm serves as an example of how a stochastic algorithm can be structured to utilize a continuous POMDP PDM model and that, potentially, such an algorithm can achieve a level of performance approaching that of an exact method, but at a fraction of its computational cost. On a set of test scenarios, the PF algorithm on average achieved over 92% of BT reward values. As the scenario complexity increased, the efficiency advantage of a stochastic algorithm became more pronounced. On the 10-waypoint scenario the PF algorithm achieved the above results while making an order of magnitude fewer calls to the vehicle simulator compared to BT.

In the near future we plan to incorporate measures of prognostic uncertainty associated with a particular policy into the policy selection process. This could be useful in situations where reduction of prognostics uncertainty over a period of time is desired or where limits on it are defined. Methods for performance evaluation of algorithms using such measures will need to be identified, as comparison against deterministic algorithms may not be meaningful. Other directions of future research include development of multi-fidelity reasoning (where system model complexity is adjusted depending upon the requirements and the circumstances) and further work on methods for PDM problem decomposition.

Acknowledgment

The authors would like to thank their colleagues at NASA Ames Research Center and Stanford University for their assistance with performing this work and preparing the manuscript for publication. Funding for this work is provided by NASA System-Wide Safety Assurance Technologies (SSAT) project (Aviation Safety Program).

Nomenclature

α	angle of attack
η_m	motor efficiency coefficient
η_p	propeller efficiency coefficient
η_t	power transmission efficiency coefficient
ϕ	bank angle
ρ_∞	freestream density
AR	planform aspect ratio
C_D	coefficient of drag (total)
C_t	thermal inertia coefficient
$C_{D,i}$	coefficient of induced drag
$C_{D,p}$	profile (airfoil) drag coefficient
D	drag
E	battery voltage
e	span efficiency factor
h	thermal transfer coefficient
I	current
L	lift
m	mass
P	power produced by the motor
q	battery charge required
R	component electrical resistance
r	radius of turn
Re	Reynolds number
S	planform area
T	thrust
T_a	ambient temperature
$T_{[c]}$	component temperature
v	airspeed
v_c	cruise airspeed
v_∞	freestream velocity
DM	Decision Maker
DX	Diagnostics (or Diagnoser)
EoL	End of Life
MDP	Markov Decision Process
PDM	Prognostic Decision Making
POMDP	Partially Observable Markov Decision Process
RUL	Remaining Useful Life
SoC	State of Charge (battery)
VS	Vehicle Simulation

References

Balaban, E., & Alonso, J. J. (2012). An Approach to Prognostic Decision Making in the Aerospace Domain. In *Annual conference of the prognostics and health management society.* Minneapolis, MN.

Barsali, S., & Ceraolo, M. (2002, March). Dynamical Models of Lead-Acid Batteries: Implementation Issues. *IEEE Transactions on Energy Conversion*, *17*(1), 16–23.

Bellman, R. (1957a). *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Bellman, R. (1957b). *A Markovian Decision Process* (Tech. Rep.). Santa Monica, CA: RAND Corporation.

Bertsekas, D. P. (1995). *Dynamic Programming and Optimal Control*. Athena Scientific.

Bogdanov, A., Chiu, S., Gokdere, L. U., & Vian, J. (2006). Stochastic Optimal Control of a Servo Motor with a Lifetime Constraint. In *Proceedings of the 45th ieee conference on decision and control.*

Brown, D. W., & Vachtsevanos, G. J. (2011). A Prognostic Health Management Based Framework for Fault-Tolerant Control. In *Annual conference of the prognostics and health management society.* Montreal, Canada.

Chen, M., & Rincon-Mora, G. (2006). Accurate Electrical Battery Model Capable of Predicting Runtime and I-V Performance. *IEEE Transactions on Energy Conversion*, *21*(2), 504–511.

Chen, Y., Song, L., & Evans, J. W. (1996). Modeling Studies On Battery Thermal Behaviour, Thermal Runaway, Thermal Management, and Energy Efficiency. In *Proceedings of the 31st intersociety energy conversion engineering conference (iecec).*

Daigle, M. J., Bregon, A., & Roychoudhury, I. (2012). A Distributed Approach to System-Level Prognostics. In *Annual conference of the prognostics and health management society.* Montreal, Canada.

Daigle, M. J., & Roychoudhury, I. (2010). Qualitative Event-Based Diagnosis: Case Study on the Second International Diagnostic Competition. In *21st international workshop on principles of diagnosis.*

Daigle, M. J., Saxena, A., & Goebel, K. (2012). An Efficient Deterministic Approach to Model-based Prediction Uncertainty Estimation. *Annual Conference of the Prognostics and Health Management Society*.

Denney, E., Pai, G., & Habli, I. (2012). Perspectives on Software Safety Case Development for Unmanned Aircraft. In *Ieee/ifip international conference on dependable systems and networks.* Boston, MA.

Edwards, D., Orchard, M. E., Tang, L., Goebel, K., & Vachtsevanos, G. (2010). Impact of Input Uncertainty on Failure Prognostic Algorithms: Extending the Remaining Useful Life of Nonlinear Systems. In *Annual conference of the prognostics and health management society.* Portland, OR.

Ferguson, D., & Stentz, A. (2006). Using Interpolation to Improve Path Planning: The Field D* Algorithm. *Journal of Field Robotics*, *23*(2), 79–101.

Gordon, N., Salmond, D., & Smith, A. (1993). Novel Approach to Nonlinear/non-Gaussian Bayesian State Estimation. *IEE Proceedings F (Radar and Signal Processing)*, *140*(2), 107–113.

Hogge, E., Quach, C., Vazquez, S., & Hill, B. (2011). *A Data System for a Rapid Evaluation Class of Subscale Aerial Vehicle, NASA/TM2011-217145* (Tech. Rep.). Hampton, VA.

Jacobson, D. H., & Mayne, D. Q. (1970). *Differential Dynamic Programming*. American Elsevier.

Knuth, D. E. (1968). *The Art of Computer Programming*. Addison-Wesley.

Lewis, K. W. (1984). *The Cumulative Effects of Roughness and Reynolds Number on NACA 0015 Airfoil Section Characteristics* (Tech. Rep.). Texas Tech University.

MATLAB version 7.11.0.584 (2010b). (2010). Natick, Massachusetts: The MathWorks Inc.

Milanfar, P., & Lang, J. (1996). Monitoring the Thermal Condition of Permanent-Magnet Synchronous Motors. *IEEE Transactions On Aerospace And Electronic Systems*, *32*(4), 1421–1429.

Miller, S. (2008). *Lift, Drag, and Moment of a NACA 0015 Airfoil* (Tech. Rep.). Ohio State University.

Narasimhan, S., & Brownston, L. (2007). HyDE - A General Framework for Stochastic and Hybrid Model-based Diagnosis. In *Proceedings of the international workshop on the principles of diagnosis.* Nashville, TN.

Orchard, M. E. (2007). *A Particle Filtering-based Framework for On-line Fault Diagnosis and Failure Prognosis* (Doctoral Thesis). Georgia Institute of Technology.

Pereira, E. B., Galvao, R. K. H., & Yoneyama, T. (2010). Model Predictive Control using Prognosis and Health Monitoring of Actuators. In *2010 ieee international symposium on industrial electronics.*

Pineau, J., Gordon, G., & Thrun, S. (2006). Anytime Point-Based Approximations for Large POMDPs. *Journal of Artificial Intelligence Research*, *27*, 335–380.

Saha, B., Quach, C., & Goebel, K. (2012). Optimizing Battery Life for Electric UAVs using a Bayesian Framework. In *2012 ieee aerospace conference.*

Tang, L., Hettler, E., Zhang, B., & Decastro, J. (2011). A Testbed for Real-Time Autonomous Vehicle PHM and Contingency Management Applications. In *Annual conference of the prognostics and health management society.*

Novelty detection in airport baggage conveyor gear-motors using Synchro-squeezing transform and Self-organizing maps

Budhaditya Hazra[1], Shilpa Pantula[2], and Sriram Narasimhan[3]

[1,2]*Department of Civil and Environmental Engineering, University of Waterloo, Waterloo, ON N2L 3G1, CANADA*

b2hazra@uwaterloo.ca
srpantul@uwaterloo.ca
sriram.narasimhan@uwaterloo.ca

ABSTRACT

A powerful continuous wavelet transform based signal processing tool named Synchro-squeezing transform (SST) has recently emerged in the context of non-stationary signal processing. Founded upon the premise of time-frequency (TF) reassignment, its basic objective is to provide a sharper representation of signals in the TF plane. Additionally, it can also extract the individual components of a non-stationary multi-component signal, which makes it attractive for rotating machinery signals. This work utilizes the decomposing power of SST transform to extract useful components from gear-motor signals in relevant sub-bands, followed by the application of standard rotating machinery condition indicators. For timely detection of faults in airport baggage conveyor gear-motors, a novelty detection technique based on the recently developed concepts of self-organizing maps (SOM) is applied on the condition indicators. This approach promises improved anomaly detection power than that can be achieved by applying condition indicators and SOM directly to the inherently complex raw-data. Data collected from the airport baggage conveyor gear-motors provides the test bed to demonstrate the efficacy of the proposed approach.

1. INTRODUCTION

Faults in gearmotors can lead to catastrophic failures in airport baggage handling system (BHS) infrastructure leading to substantial downtime, significant monetary losses and expensive replacement scheduling. Ensuring their smooth operation requires maintenance, so that any change in the condition such as deterioration or damage can be detected in a timely manner. This can be accomplished through a combination of signal processing of gear motor vibration signatures to detect faults and novelty detection to classify a healthy from a faulty state without the historical knowledge of faults. In this study, we propose a novelty detection algorithm for condition assessment of gearmotors. The algorithm utilizes the recently developed concepts of synchro-squeezing transform (Daubechies, Lu & Wu, 2011) and integrates it with the traditional condition monitoring indicators (Vecer, Kreidl & Smid, 2005) and self-organized map (SOM) (Kohonen, 1990) based novelty detector (Lee & Cho, 2005) to monitor the healthy and faulty states of the BHS gearmotors.

In order to accomplish accurate fault diagnosis, it is important that the acquired rotating machinery vibration signals have good signal to noise ratio and less complexity. The complexity of rotating machinery signals are attributed to individual contributions from different sources like gears, bearings, rotors, and motors etc. to the overall vibration response. The problem is compounded further in presence of noise and transients from faulty machine components. This can be addressed by considering the extraction of meaningful components from the mixed signals. The other alternative is to consider the signals directly as they evolve and avoid decomposing them into different components and instead make use of a spectral techniques (Jardine, Lin & Banjevic, 2006) to diagnose faults.

Traditional signal processing methods towards gear fault diagnosis comprises of non-parametric spectral analysis methods like the Fourier transform, Cepstrum analysis (Jardine et. al, 2006) and Envelope spectrum analysis (Antoni & Randall, 2011). Recent trends in fault diagnosis have witnessed a shift towards the application of time-frequency representation (TFR) methods like short-time Fourier transform (Cohen, 1996), wavelet transforms (Wang & Mcfadden, 1995) and Wigner-Ville distribution (Cohen, 1996) to accommodate the innate non-stationarity in the gear vibration signals. Time series based techniques like auto-regressive moving average models and their variants, applied using time-invariant coefficients (Zhan & Jardine, 2005); have provided attractive options in the family of

Budhaditya Hazra et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

parametric spectral approaches. Since gearbox signals comprise mainly of time-varying frequency components and amplitudes, the use of Kalman filtering based techniques for modeling time varying ARMA models is worth mentioning (Zhan & Jardine, 2005). But all of the parametric models suffer from the problem of model order selection which is again an impeding factor as far as dealing with complex and noisy rotating machinery signals is concerned.

In the alternative approach of component extraction methods, blind source separation (BSS), requiring multiple channels of data, has been applied on numerous cases of machinery fault detection (Ypma, Leshem, & Duin, 2002). The families of signal decomposition methods like empirical mode decomposition (EMD) (Lei, Lin, He, & Zuo, 2013) and synchro-squeezing transform (SST) (Daubechies et. al, 2011), applied to single channel measurements holds significant promise in this regard. EMD is a powerful and robust signal processing technique that requires only one signal measurement. However, its robustness is sometimes encumbered by its poor performance in noise and requirement of intermittency criterion. Moreover, it is essentially an empirical method lacking in rigorous mathematical construct. Synchro-squeezing transform (Daubechies et. al, 2011) is a relatively new and promising signal processing tool based on the concepts of CWT. It can decompose noisy and non-stationary signals into its components without the restrictive requirement of intermittency criterion and provides a more robust alternative to EMD for gearbox signals (Liang & Li, 2012).

Signal processing alone is not adequate to address the complete problem of condition monitoring of gear boxes. It merely generates some diagnostic patterns. These patterns need to be processed through inference tools like pattern recognition, pattern classification, novelty detection etc. (Timusk, Lipsett & Mechefske, 2008). Novelty detection (also called anomaly/outlier detection) is the process of finding an unusual behavior in machinery that has not been observed before. It is essentially a two-stage process when applied in the context of condition monitoring of machines. The first stage entails learning or training, in which the novelty detector learns by utilizing the data from a machine in normal condition. After the training stage, the detector is fed with data from the machine in a running condition to get a novelty score. If data is similar to the training data in some sense, the novelty detector shows a similair trend and the novelty score is low. Novelty score increases when there is a deviation in the operating performance of the machine or an anomaly. Higher the novelty score, higher is the level of fault in the machinery (Worden, Manson & Fieller, 2000).

A self-organizing map (SOM) is a type of artificial neural network (ANN) trained using unsupervised learning to produce a low-dimensional, discretized representation of the input space called a map (Kohonen, 1990). SOM is based on nonlinear projection of the input space to some (usually lower dimensional) output space like Principal Component Analysis. Two properties of SOM widely applicable to condition monitoring are vector projection (VP) and vector quantization (VQ). Vector projection essentially involves projecting the multidimensional data to a lower dimensional space. VQ reduces the number of samples or substitutes them with representative centroids. The accuracy of the representation of the input data in a two-dimensional map can be used as the novelty score.

This paper is based on the application of SST to identify useful signal components and then apply condition indicators (Vecer et. al, 2005) and a SOM based novelty detector to detect faulty states in a BHS gearmotor. SST (Daubechies et. al, 2011) belongs to the family of time-frequency reassignment methods that not only provide a sharp TFR but also allows extraction of the individual components (intrinsic mode functions or IMFs) of a general multicomponent non-stationary signal like EMD, yet in a much more mathematically structured manner free of restrictive requirements of intermittency. The extracted IMFs are then utilized to assess the machine health condition by subsequent application of standard condition indicators and novelty detection, in keeping with the recent trends where better diagnosis results are reported when signal processing algorithms are used in conjunction with condition indicators (Hazra & Narasimhan, 2013) and novelty detectors (Timusk, Lipsett & Mechefske, 2008).

2. Synchro-squeezing transform

Since its introduction in the context of speech signals SST has evolved into an EMD-like tool (Daubechies et. al, 2011) capable of decomposing a multi-component non-stationary signal into AM-FM components that resemble intrinsic mode functions (IMFs). It is in fact a type of time-frequency reassignment algorithm that works by reallocating the continuous wavelet transform (CWT) coefficients based on the frequency information, to obtain a sharper representation in the time-frequency plane.

To understand the basic idea of SST, it is instructive to review some of the concepts of CWT. CWT of a signal $\mathrm{s(t)}$ can be mathematically defined as an inner product:

$$\mathrm{W_s(a,b)} = \frac{1}{\mathrm{a}}\int_{\mathrm{R}} \mathrm{s(t)}\overline{\psi\left(\frac{\mathrm{t-b}}{\mathrm{a}}\right)}\mathrm{dt} \tag{1}$$

where, $\psi(\mathrm{t})$ is the mother wavelet, a and b are scale and shift parameters, respectively. In the context of synchrosqueezing, an essential requirement for the mother wavelet is that it must have a unique peak frequency (Oberlin, Meignen, & Perrier, 2012). For such a wavelet, let us denote its central frequency as ω_ψ and let ϵ_ψ be the extremal value so that ψ is supported compactly in the interval $\left[\omega_\psi - \epsilon_\psi , \omega_\psi + \epsilon_\psi\right]$. As an example (Oberlin, Meignen, & Perrier, 2012), for bump wavelet where $\psi \propto \exp\left(-\frac{1}{1-((2\pi\omega-\mu)/\sigma)^2}\right)$, $\omega_\psi = \mu$ and $\epsilon_\psi = \sigma$.

Let us consider an example (Daubechies et. al, 2011) of a purely harmonic sinusoidal signal to illustrate the working principle of SST. The wavelet transform of the signal $\mathrm{W_s(a,b)}$ should in principle be concentrated around the frequency of the signal as a line of constant magnitude. However, in practice, it is observed that the wavelet transform is always smeared out around the horizontal line corresponding to the sinusoid frequency in the T-F plane. This problem can be addressed by estimating instantaneous frequency ω(a,b) for all values of (a, b), which is given by the following formula (Daubechies et. al, 2011):

$$\omega(a,b) = \frac{-i}{W_s(a,b)}\frac{\partial}{\partial b}\, W_s(a,b) \qquad (2)$$

The primary objective for calculating the instantaneous frequency $\mathrm{\omega(a,b)}$ is that if the signal s(t) possesses an IMF like characteristic, or is of the form $\mathrm{s(t) = a(t)cos\phi(t)}$ (i.e. Hilbert transform is $Hs(\mathrm{t}){\sim}\mathrm{a(t)sin\phi(t)}$), then $\mathrm{\omega(a,b)}$ calculated using Eq. (2) is approximately equal to $\phi\,'(\mathrm{t})$. This suggests that, for asymptotic signals, synchrosqueezing will give a single line on the time-frequency plane, at the value corresponding to the instantaneous frequency of an IMF. Implementation-wise the wavelet coefficients in $\mathrm{W_s(a,b)}$ are computed only at discrete scales a_k and its synchrosqueezed counterpart $\mathrm{T_s}(\omega,\mathrm{b})$ is determined at the centers ω_c of the successive bins $[\omega_c - \frac{1}{2}\Delta\omega, \omega_c - \frac{1}{2}\Delta\omega]$, by the following formula (Daubechies et. al, 2011):

$$T_s(\omega_c,b)= \frac{1}{\Delta\omega} \sum_{a_k:|\omega(a_k,b)-\omega_c|\le\frac{\Delta\omega}{2}} W_s(a_k,b)\, a_k^{-3/2}\, \Delta a_k \qquad (3)$$

where, $\Delta\omega = \omega_c - \omega_{c-1}$ and $\Delta a_k = a_k - a_{k-1}$

The next step in the SST entails extraction of the IMFs. This involves extraction of one "ridge" from $\mathrm{T_s}$ by finding the curve c(t) with the largest energy subject to some optimization criterion (Oberlin, Meignen, & Perrier, 2012). Once the curve c(t) is known, the associated mode h can be estimated by summing the SST coefficients near that curve: at time t, according to the following equation:

$$h(t)= \int_{\omega\approx c(t)} T_s(\omega,t)d\omega \qquad (4)$$

The main problem in this approach is stability of curve extraction. Deviations of IMFs from their asymptotic behavior or contamination by noise make the extraction unstable (Oberlin, Meignen, & Perrier, 2012). In the present work, the authors follow the approach proposed by Oberlin et. al (2012). The method is based on utilizing a ridge near the frequency peak instead of using ω as suggested by some authors (Daubechies et. al, 2011). The ridge is defined by a set of coefficients as per the following equation (Oberlin et. al, 2012):

$$Y_t=\left\{a\ :\ \frac{\omega_\psi - \epsilon_\psi}{c(t)} \le a \le \frac{\omega_\psi + \epsilon_\psi}{c(t)}\right\} \qquad (5)$$

The details of the procedure are beyond the purview of this work and the readers are referred elsewhere (Oberlin et. al, 2012).

2.1. Numerical example

Let us consider a mixture of one pure sinusoid and 2 AM-FM type signals. A gearbox in its pristine state can be represented by a pure sinusoid (Hazra & Narasimhan, 2013) whose frequency matches with the meshing frequency (shaft rotation frequency times the number of gear teeth of the gear). Meshing defects in gear are manifested by the appearance of the sidebands around the meshing harmonic which can be typically represented by amplitude modulating and frequency modulating (AM-FM) signal. Thus the signal can be written as:

$$s\ (t) = \sin(2\pi 110t) + [1 + 5.5\cos(2\pi 2t)]\ \sin(\ 2\pi\mathbf{160}t + 2\sin(2\pi 10t)) + [1 + 5.5\cos(2\pi 2t)]\ \sin(\ 2\pi\mathbf{230}t + 2\sin(2\pi 10t)) + 0.7\mathrm{randn}\ (1, \mathrm{length}\ (t)). \qquad (6)$$

Fig. 1 shows the plots of the signal s (t) and its Fourier spectrum and the also the spectra of its pure sinusoidal and AM-FM components. Fig 2 shows plots of recovered IMFs. It can be observed that the synchro-squeezing transform is able to extract all the 3 components with good accuracy.

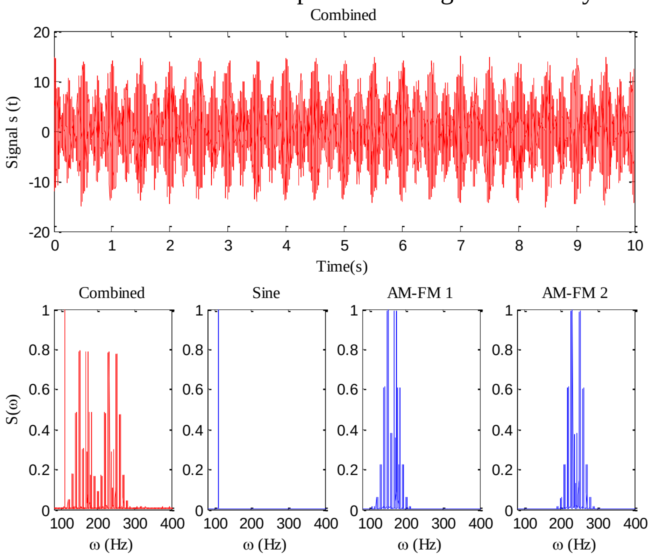


Figure 1: Signal and its components

We further consider the case of real data obtained from one of the accelerometers placed on the airport BHS gear motor (Fig. 6). The description of the BHS measurement program is mentioned in details in section. 5. Fig. 3 shows the acceleration and its synchro-squeezed time frequency representation. It is clear from Fig. 4 that SST is able to extract the key significant energy IMFs from the gearbox signal. Furthermore, it can be noted that the extracted IMFs are consistent with the time frequency representation of the signal.

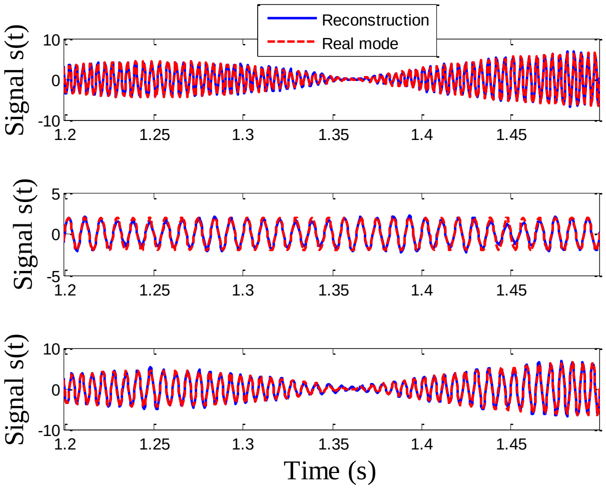


Figure 2: Recovered IMFs

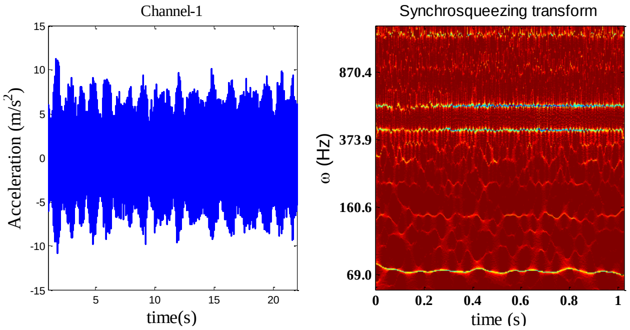


Figure 3: Acceleration data of a BHS gearmotor & its Synchro-squeezing transform

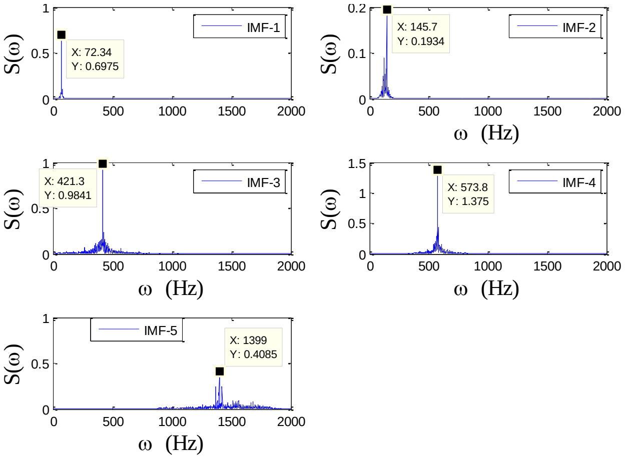


Figure 4: Recovered IMFs from the BHS data

To compare the performance of CWT and SST, we consider the same signal with added noise of SNR=20. It can be clearly observed from Fig. 5, that SST significantly reduces noise and is clearly able to delineate the AM-FM and the sinusoidal components. This property is particularly useful in dealing with noisy data. Thus, SST serves two important purposes; reduces noise and is able to decompose a non-stationary signal into its components.

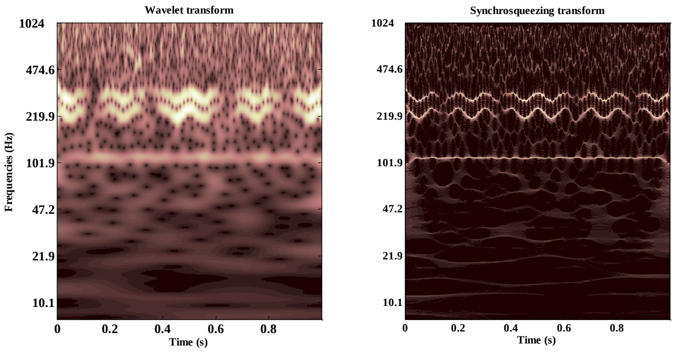


Figure 5: Comparative performance of SST and CWT in noisy data (SNR=20)

3. Novelty Detection

Self-Organizing Map (Kohonen, 1990), is used as the novelty detector in the present work. Majority of its applications are in the visualization of nonlinear relations of multidimensional data. It has also been applied in rotating machinery diagnostics (Timusk, Lipsett & Mechefske, 2008). SOM is a two-dimensional array containing neurons. A prototype vector (also called model or codebook vector), having same dimension as the input data set is associated with each neuron. This prototype vector approximates a subset of the sample vectors. During the training phase, sample vectors are assigned to the most similar prototype vector, also called best-matching unit (BMU). The algorithm trains itself in such a manner that similar input samples are mapped to the relatively close BMUs. The prototype vectors are updated iteratively during the training steps by selecting the sample randomly. The neighborhood kernel, whose radius decreases with training steps, determines the influence on the neighboring codebook vectors. Learning starts with rough learning phase having a big influence area and fast-changing codebook vectors, shifting gradually to a fine-tuning phase with small influence area and slowly adapting codebook vectors. This algorithm is referred to as sequential training or basic SOM.

SOM has also been applied to novelty detection (Lee & Cho, 2005). Given training set X, containing N normal patterns, SOM is trained to generate a set of codebook vectors $w = \{w_k | k = 1,2,\dots,K\}, K \ll N$. The codebook vector $m(x)$ of an input vector x and the Voronoi region S_k of each codebook vector w_k are defined as follows,

$$m(x) = w_k \Leftrightarrow x \in S_k,$$

if, $$\|w_k - x\|^2 < \|w_l - x\|^2, \forall l \neq k \tag{7}$$

Given a test pattern z, the Euclidean distance (quantization error) $e(z)$ between z and $m(z)$ is calculated as:

$$e(z) = \|z - m(z)\|^2 \tag{8}$$

If this is greater than a threshold value, then it is considered to be novel. To identify the threshold value, the quantization errors corresponding to the training patterns are computed.

In the present problem, SOM is trained using condition indicators (CI) estimated using the first IMF and sum of first 3 IMFs obtained from the application of SST to the acceleration data of a gearmotor in relatively new health state. The codebook vectors and quantization errors are computed to set a threshold value. Data from gearmotor in non-normal state are fed to the algorithm to compute quantization errors. The quantization errors are calculated pointwise and the average of the quantization errors over a fixed size data window is considered in this study. If the test pattern has a mean quantization error more than the threshold, then the test pattern is identified to be novel. Mean quantization error, computed using the test set from machines in non-normal conditions is used as novelty score. Higher the novelty score, higher is the level of fault in the machine.

3.1. Condition indicators (CI)

To detect the condition of the gears and bearings, 4 condition indicators (Vecer et. al, 2005) which have been widely used in the literature, are chosen, namely: variance, kurtosis, crest factor and the energy operator. Only a brief description of the performance indicators is provided here.

- Variance: The variance of a signal is defined as:

$$\text{var(s)}=\frac{1}{\text{N}}\sum_{\text{i=1}}^{\text{N}}(\text{s}(\text{t}_\text{i})\text{-}\mu)^2 \tag{9.1}$$

  where, μ is the mean of the signal $s(t)$ and N is the number of samples.
- Kurtosis: The kurtosis of a signal is the normalized fourth moment and is defined as:

$$\text{kurt(s)}=\frac{1}{\text{N}}\sum_{\text{i=1}}^{\text{N}}\frac{(\text{s}(\text{t}_\text{i})\text{-}\mu)^4}{\sigma^4} \tag{9.2}$$

  where, σ is the standard deviation of the signal
- Energy operator: The energy operator for a signal s(t) is defined as [15]:

$$\text{EOP(s)} = \Delta\text{x(i)} = s(i+1)^2 - s(i)^2 \tag{9.3}$$

  where, and $\overline{\Delta x}$ is the mean value of Δx vector.
- EOP variance and EOP kurtosis are calculated in the same manner using the formulas (7.1) and (7.2)
- Crest factor: The crest factor (CF) for a signal s(t) is defined as:

$$CF = \frac{max(x(i)) - min(x(i))}{\sigma} \tag{9.4}$$

4. PROPOSED ALGORITHM

The main steps of the proposed algorithm are as follows:

- Calculate the CWT W_s(a,b) of $s(t)$
- Calculate the instantaneous frequency $\omega(a, b)$
- Calculate the SST $\text{T}_\text{s}(\omega, \text{b})$ over the TF plane
- Extract dominant curves from c(t) from $\text{T}_\text{s}(\omega, \text{b})$
- Reconstruct the signal as a sum of components, one for each extracted dominant curve
- Apply the CI on the most dominant IMFs
- Apply SOM to the CI, treating the data from the new healthy state as the training set
- Calculate the mean quantization errors between the codebook vector and the subsequent windows of data

5. RESULTS FROM FIELD EXPERIMENTATION PROGRAM

Recently the authors have engaged in condition based maintenance program aimed at detecting faults in the Toronto Pearson airport baggage handling system (BHS) gearboxes. The main idea is to gather acceleration data from the BHS system gearboxes and develop sophisticated algorithms towards automated diagnostics and prognostics of the gearboxes. As a part of the field instrumentation and data acquisition programme, vibration data was collected for a few minutes on a particular day of every week starting from May-2013. The schematic of a typical gearmotor and the data-acquisition set-up is shown in Fig. 5. The sampling frequency was kept at 4000 Hz. The fundamental meshing frequency is approximately close to 80Hz depending upon the conveyer belt rpm which varies between 170 to 200 rpm. Data were collected from a selected gearmotor at its new and old health stages. Fourier spectra (Fig. 6) of the gearmotor data shows the presence of sidebands with significant energy in the data at the old health state compared to the new one.

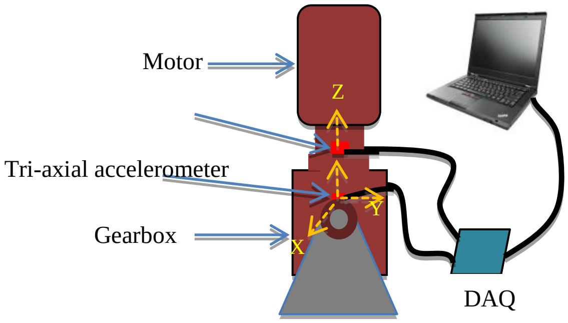


Figure 6: Schematic of the gearmotor and the instrumentation set-up

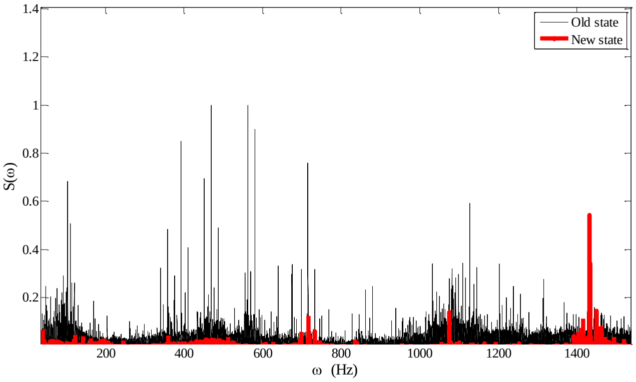


Figure 7: Fourier spectra of the gearmotor at old and new health states

The raw data from new healthy state gearmotor is concatenated to the data from the old state. The condition indicators (CI) are applied on the concatenated raw data, IMF-1 and the sum of first 3-IMFs obtained using SST. SST is applied on sub-windows of 4096 samples of data and concatenated to form the IMF vectors. The main idea of extracting first 3 IMFs and summing them up is that the first 3-IMFs contribute to most of the signal's energy content. The higher order IMFs contains contributions mostly from the noise. The condition indicators are estimated recursively considering every sample of the data. Fig. 8 shows the condition indicator for the raw data, IMF-1 and the sum of first 3 IMFs. It can be observed that there is a considerable jump in the CI values after approximately first 70000 samples of data. These samples represent the data obtained from the gearbox at the new state of health. The remaining part of CIs represents data at old health state.

The trend in the values of recursive variance, kurtosis and energy kurtosis for raw data is not as clear as it is for the corresponding CI values for IMF-1 and sum of the first 3 IMFs, indicating thereby that the combination of SST and CIs provides a better indication of comparative health states than the combination of raw data and CIs. The uncertainties associated in the estimates of the condition indicators by using raw data and the sum of IMFs 1, 2 & 3 is shown in Fig. 9. From Fig. 9, it can be observed that the variance-ratio (ratio of variances between the CIs corresponding to the old stage data and that corresponding to the new stage of data) of the condition indicators using raw data is much more compared to the same when estimated using the sum of IMFs. This clearly points towards more uncertainties associated with the use of condition indicators on the raw data.

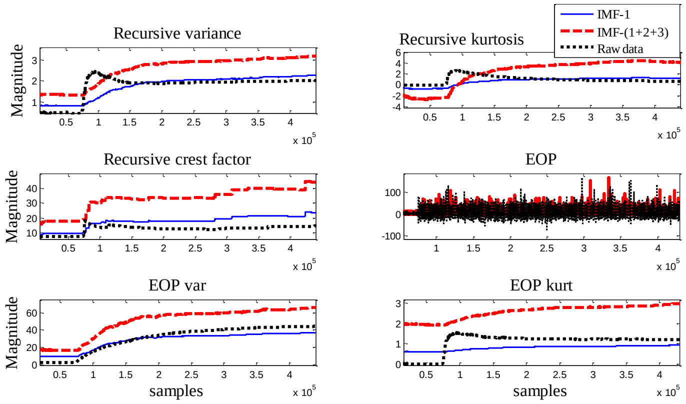


Figure 8: Recursive condition indicators applied on the raw data and IMFs

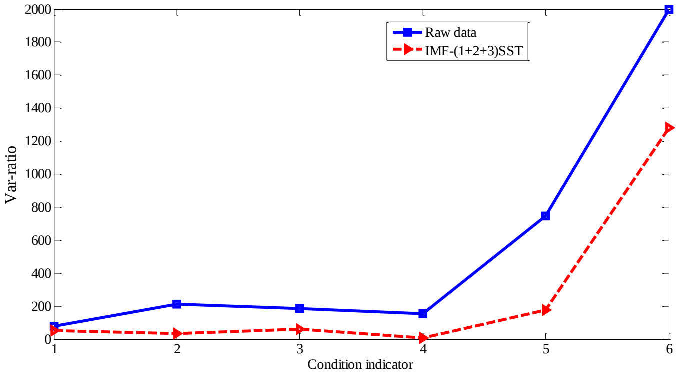


Figure 9: Uncertainties in the condition indicator estimates

Novelty detection is applied as mentioned in the previous sections. First 70,000 samples of the raw data and also the IMF-1 is treated as the training set and the mean of the quantization error as given by Eq. 8 is estimated for the successive windows of test data. Fig. 10 shows the plot of the average quantization error. It can be clearly observed that the novelty score increases for the successive windows of data. This implies that the data from the successive windows represent a more novel or an anomalous state of data compared to the first window (training window). A health state is typically indicated by approximately constant values of Q_{avg} over successive windows. Novelty detection applied on the sum of the IMFs (1, 2 & 3) obtained using SST shows best and most consistent performance. It is closely followed by the performance of the sum of 3-IMFs extracted using SST with 10% added noise and the case with IMF-1 only. The performance of novelty detection applied on raw data degrades significantly. This is clear from the trend of the Q_{avg} values for the raw data in Fig. 10, which fails to establish health states or values that are nearly constant over successive windows or 3 consecutive windows. The performance of novelty detection is worst for raw data with 10% noise. Thus, it can be concluded that the

SOM novelty indicator is clearly able to distinguish between the relatively older and newer health states. Superior performance of SST even with added levels of noise is the key result.

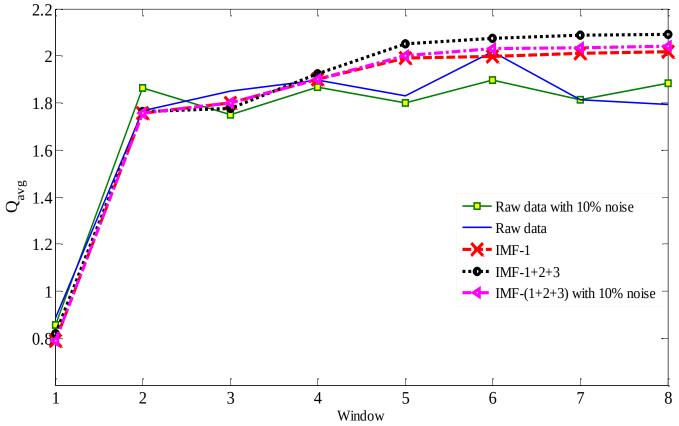


Figure 10: Average quantization error for the SOM based novelty detector

6. Conclusions

A new novelty detection algorithm towards fault diagnosis of airport baggage handling system gearmotors using a combination of synchro-squeezing transform, traditional rotating machine condition features and self-organized maps is presented. The reassignment property of synchro-squeezing transform allows for a better resolution of the signal features in the presence of noise. Subsequent application of curve extraction techniques along the ridges allowed EMD like decomposition of the signal into IMFs. Application of condition indicators recursively to the IMFs clearly shows the trend indicative of health degradation in the experimental data from the gearmotor. Application of SOM based novelty detector further delineated the new and the degraded health states of the airport baggage handling system gearmotor.

Acknowledgement

The work was supported by the MITACS accelerate program of the province of Ontario, CANADA.

References

Antoni, J. and Randall, R, B. (2011). Rolling element bearing diagnostics—A tutorial. *Mechanical Systems and Signal Processing*, 25(2), pp. 485-520.

Cohen, L. (1995). *Time-Frequency Analysis*. Prentice-Hall, Englewood Cliffs. NJ, USA.

Daubechies, I., Lu, J., and Wu, H. (2011). Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. *Applied and Computational Harmonic Analysis*, 30(2), pp. 243–261.

Hazra, B., and Narasimhan, S. (2013), Rotating machinery diagnosis using synchro-squeezing transform based feature analysis. *Proceedings of MFPT Conference*. May 13-17, Cleveland, Ohio.

Jardine, A. K. S., Lin, D., and Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition based maintenance. *Mechanical Systems and Signal Processing*, 20(7), pp. 1483-1510.

Kohonen. T. (1990), The Self-Organizing Map. *Proceedings of the IEEE*, 8(9), pp. 1464–1480.

Lee, H. J., & Cho, S. (2005). SOM-based novelty detection using novel data. *Intelligent Data Engineering and Automated Learning-IDEAL*. Springer, Berlin Heidelberg, pp. 359-366.

Lei, Y., Lin, J., He, Z., and Zuo, M, J., (2013). A review on empirical mode decomposition in fault diagnosis of rotating machinery. *Mechanical Systems and Signal Processing*, 35(1-2), pp. 108-126.

Liang, M., and Li, C. (2012). A generalized synchro-squeezing transform for enhancing signal time–frequency representation. *Signal Processing*, 92(9), pp. 2264-2274.

Oberlin, T., Meignen, S., and Perrier, V. (2012), On the Mode Synthesis in the Synchrosqueezing method. *Proceedings of EUSIPCO*, Bucharest, Aug. 27-31.

Staszewski, W., Worden, K., and Tomlinson, G. (1997). Time frequency analysis in gearbox fault detection using the Wigner-Ville distribution and pattern recognition. *Mechanical Systems and Signal Processing*, 11(5), pp. 673–692.

Timusk, M., Lipsett, M., and Mechefske, C, K., (2008). Fault detection using transient machine signals. *Mechanical Systems and Signal Processing*. 22(7), pp. 1724-1749.

Vecer, P., kreidl, M., and Smid, R. (2005). Condition indicators for gearbox condition monitoring systems. *Acta Polytechnica*, 45(6), pp. 35-43.

Wang, W. and McFadden, P. (1995). Application of orthogonal wavelets to early gear damage detection. *Mechanical Systems and Signal Processing*, 9(5), 497–507.

Worden, K., Manson, G., & Fieller, N. R. J. (2000). Damage detection using outlier analysis. *Journal of Sound and Vibration*, *229*(3), 647-667.

Ypma, A., Leshem, A., and Duin, R. (2002). Blind separation of rotating machine sources: bilinear forms and convolutive mixtures. *Neurocomputing*, 49(1-4), 349-68.

Zhan, Y. and Jardine, A. (2005). Adaptive autoregressive modeling of non-stationary vibration signals under distinct gear states Part 1: modeling. *Journal of Sound and Vibration*, 286(6), 429-450.

Budhaditya Hazra received his undergraduate degree from the National Institute of Technology, Durgapur, India, Master of Technology from the Indian Institute of Technology, Kanpur, India, and his PhD from the University of Waterloo, Canada, in 2010. He is currently a post-doctoral research associate at the University of Waterloo. His research interests are in the areas of system identification and machine diagnostics.

Shilpa Pantula received her undergraduate degree from the Indian Institute of Technology, Chennai, India in 2012. She is currently pursuing her Master of Applied Science at the University of Waterloo, Canada. Her research interests are in structural engineering and machine diagnostics.

Sriram Narasimhan received his PhD from Rice University, Houston, Texas, USA, in 2005. He is an Associate Professor in the departments of Civil and Environmental Engineering & Mechanical and Mechatronics Engineering at the University of Waterloo, Canada. His research interests are in the areas of system identification, control, and machine diagnostics. He is a member of ASCE, and serves on the editorial boards of ASCE Journal of Structural Engineering, Bridge Engineering and Structural Control and Health Monitoring.

Towards an Online Prognostic System for Predicting the Axial Shrinkage of AGR Cores

Graeme M. West[1], Christopher J. Wallace[2], and Stephen D. J. McArthur[3]

[1,2,3]*Institute for Energy and Environment, Dept. EEE, University of Strathclyde, Glasgow, G1 1XW, UK*

graeme.west@strath.ac.uk
christopher.wallace@strath.ac.uk
s.mcarthur@strath.ac.uk

ABSTRACT

In the UK, there is the desire to extend the operation of the Advanced Gas-cooled Reactor (AGR) power plants beyond their initial design lifetimes of 35 years. As part of the justification of extended operation, an increased understanding of the current and future health of the graphite reactor cores is required. One measure of the health of the AGR power plants is the axial height of the graphite core, which can be determined through measurements undertaken during statutory outages. These measurements are currently used to manually make predictions about the future height of the core, through identifying the relevant data sources, extracting the relevant parameters and generating the predictions is time-consuming and onerous. This paper explores an online prognostic approach to support these manual predictions, which provides benefits in terms of rapid, updated predictions as soon as new data becomes available. Though the approach is described with reference to a case study of the UK's AGR design of power plant, similar challenges of predicting passive structure health also exist in other designs of power plant with planned license extensions.

1. INTRODUCTION

The condition of the graphite core of an AGR power plant is a major life-limiting factor. The graphite core provides moderation of the nuclear fission process and provides the structure in which to house the uranium fuel and provides pathways for both the gas coolant and for control rods. Three underlying processes govern the degradation of the graphite. Firstly, the neutron moderation process causes dimensional change to the graphite when fission neutrons collide with the nuclei of the carbon atoms. In parallel, the strength of the graphite is altered by both the neutron irradiation, which increases the strength and radiolytic oxidation, which reduces it (Shennan, 1993). The dimensional change in the graphite provides a useful measure of the degradation of the core, and is easier to measure in-situ than the strength of the graphite. Unlike most other components of nuclear power plants (NPP) (for example PWR Reactor Vessel heads and internals and the core pressure tubes in CANDU reactors (IAEA technical report, 2008) the graphite bricks, which comprise the core of an AGR, are irreplaceable. Therefore, predictions of Remaining Useful Life (RUL) for the graphite bricks also dictate to a large extent the remaining useful life of the power plant.

Making predictions about the dimensional change of graphite in AGR stations is not new. For example, Shennan (1983) describes predictions relating to dimensional change made about the graphite in Hinkley Point B power plant in 1979 (Hinkley Point B began generating electricity in February 1976). When the power plants were built, predictions were made based on theoretical and experimental data (as no operational data was available). A lifetime of operation has produced vast quantities of data, and data capture and storage equipment has advanced significantly, first with electronic loggers replacing pen and plotter devices, and then these first generation electronic loggers in turn being replaced by newer models. However, the original plant components being monitored, such as boilers, graphite bricks and pressure vessels have not been replaced. Managing and interpreting this increased volume of data necessitates the introduction of automated analysis techniques to support a limited number of experts in the field, as traditional manual approaches do not scale well. The remainder of the paper is organized as follows – the next section discusses the drivers for prognostics in nuclear power plants, followed by a section discussing the approach adopted within AGR power plants, including the manual method of prediction currently adopted. To investigate the possible use of an online prognostic system to replace this manual approach, three case studies are then presented. The results of these case studies are discussed along with a

Graeme West et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

proposed system development for such an online system, as the next stage of this work.

2. Drivers for nuclear prognostics

In general, the nuclear industry is cautious in the adoption of new technologies for reasons of safety, preferring to utilize existing proven techniques, unless there is no viable alternative. For example, the use of wireless technologies to transmit data is used in many applications, but has yet to find significant use within the nuclear domain (though progress is being made towards this, as reported in Hashemian, 2011). The field of prognostics faces similar challenges, where there are very few examples of deployed prognostics systems in use within nuclear power plants. Varde and Pecht (2012) provide a tabulated summary of the current state of prognostics, grouped by component/system type, and provides information for both online and offline prognostics. The paper indicates that rotating plant prognostics is the most mature area, and that online prognostics is, at best, still in the R&D stages for NPP components. For general reviews of prognostics applied to rotating plant, but not constrained to nuclear applications Jardine, Lin and Banjevic (2006), and Heng et al (2009) both provide useful summaries. In other non-nuclear domains there has been significant work undertaken. For example, Goh (2006) and Kothamasu, Huang and VerDuin (2006) both provide reviews of the application of prognostics to manufacturing. Prognostics in electronic systems is another area with significant progress in prognostics, with Goebel et al (2008) describing an application to battery health and Pecht (2008) providing wider coverage of the domain. Though lagging other industries, there is the recognition that prognostics could provide significant benefits to the nuclear industry, illustrated by projects such as the US light-water reactor sustainability program (US Department of Energy, 2013). Prognostics could be applied to both passive components, such as pipework, pressure vessels and graphite moderator bricks and active components, such as turbine generators and reactor coolant pumps. The first structural degradation prognostic system for nuclear power plants is mentioned in Bond, et al (2011), but this is still at the demonstration, rather than deployment stage. Coble, Humberstone and Hines (2010) have investigated prognostics for simulated data for the new IRIS power plant, and again this is at the technology demonstration stage.

3. Approach

The usefulness of prognostic systems depends on the predictive accuracy of the model used and the availability of data to generate RUL estimates from the model. For large fleets of similar assets, models can be based largely or entirely on statistical behavior, if the assumption is made that the future components will degrade in the same manner and the operating conditions are fixed (Coble and Hines, 2008). This is termed Type 1 prognostics. If information relating to the operating environment, which influences the degradation process, then Type II may be applicable (e.g. proportional hazards models, shock models). However where there are a limited number of assets, there is also often a limited quantity of data available across the lifetime of the asset. Commonly where prognostics is applied to high criticality assets, such as a component of a nuclear reactor, it is impractical from either a safety or financial perspective to allow a component to run to failure, and therefore gather sufficient data to create these statistical models.

For this reason, prognostic models for such components often include a physical model of the state of the component, based on material, chemical or some other understanding of the physical processes causing degradation in the asset. These models typically describe a macroscopic behavior of a material (structural strength, dimensional change, etc) based on experimental data or on a microscopic model of the physical processes. Since it is not practical to model microscopic behaviors of the entire asset, the most common approach to overcoming these limitations is to use the physical model as the basis for a prognostic model, estimating macro scale behavior of the asset based on some dominant characteristic of the physical model. The prognostic estimates based purely on the use of such a model are likely to be of limited usefulness, given that the material of which the asset is constructed and the environment is located are unlikely to be identical to those in the experiments used to derive the model.

1.1 Combining Model Based and Data Driven Prognostics

An alternative approach is to use a physical model as the basis of prognostic analysis, but to fuse this with a data driven approach (Pecht and Jaai, 2010) as it becomes available. It is argued that where a general model is applicable, there may be unknown parameters within a particular asset that cause the measured data to diverge from the general model, as in Figure 1. An automated, iterative approach, whereby a revised estimate of the model is generated every time new data is available allows a prognostic model to evolve as the knowledge of the asset increases. The rapid assessment of new data may revise a prognostic model such that a pre-defined limitation will be reached sooner than anticipated based on existing models. It can be seen therefore that such an approach has clear safety benefits.

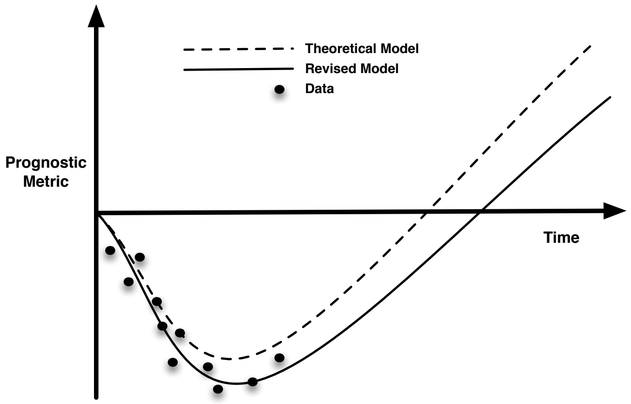


Figure 1. Generalized model of graphite dimensional change

For the graphite core used in the AGR, a model of dimensional change reported in Brocklehurst and Kelly (1993) has been adopted to estimate the evolution of dimensional change in the graphite bricks as they degrade with exposure to radiation. This model is generalized in Figure 1 and is based on previous experimental work and can provide at best a general model of the evolution of graphite under comparable parameters. A large volume of data is available describing various features of AGR cores, from inspection and monitoring activities and it is proposed that this data can supplement existing models of core evolution. Previous work by West, et. Al (2010) explored the possibility of using fuel grab load trace data, a set of monitoring measurements gathered during refuellling operations, to trend the shrinkage of the core. Though this work is related, this paper focuses solely on the use of inspection data gathered during routine outages.

2.2 An evolving prognostic system for the AGR

Based on existing data collection and condition monitoring analyses, it is possible to trend certain characteristics of the AGR core, such as brick heights and channel bore diameters, in order to calculate geometric changes in the structure of the components within the core. Using a statistical sample of different components within the reactor, a core-wide estimate of shrinkage can be derived and compared to existing predictions. Should the shrinkage be significantly smaller or larger than forecast, the prognostic model can be revised, using the most conservative data available so as to maximize safety margins. The estimation of shrinkage is used to support strategic decisions about the health of the core, and whether it can still fulfill its role as fast neutron moderator, and maintain the necessary physical structure to permit the un-impeded movement of fuel and control rods and to allow the passage of coolant gas.

3.1. Existing analysis

During statutory outages of the AGR nuclear power plants, a few fuel channels (typically 20-30) are assessed to determine the current health of the graphite bricks in these fuel channels, which in turn is extrapolated to provide an indication of the state of the whole core. This provides a short-term view of the health of the core and assessment of the data gathered during inspection needs to be assessed before the plant is returned to service. Formal documentation is produced which provides a clear statement of the current condition of each channel inspected. The raw inspection data is retained, though not in a database, but as analysis files containing both the raw data and results of analysis performed on the data.

For a longer-term view of the core, some of this data is collated and trended to provide an indication of the overall degradation in the core and to support statements that on given target dates the core is predicted to be in good health. However, gathering and filtering this data is a labor-intensive task, as suitable channels need to be identified, the associated data files located and the relevant information extracted.

There are two sources of data used to make the predictions. The first set of data is accurate measures of the internal bore of the graphite bricks throughout the height of the fuel channel. This data is gathered through specialist inspection equipment and is made available to the engineers as a raw data file. From this data three parameters are extracted which provide a representation of the shrinkage in the core:

1. A direct measurement of height
2. An average measure of brick shrinkage taken from the mid-points of each brick
3. An average measure of the full channel diameter shrinkage.

The second source of data required is to make the prediction is a measure of the cumulative irradiation the bricks have been subject to. Time is not a suitable measure, as the graphite only degrades while the plant is in use, so when predictions of RUL are made, the duty cycle of the station needs to be factored in. To ensure maximum conservatism, it can be assumed that the station is run at a constant rate, thus ensuring an under-prediction in the RUL. This cumulative irradiation measure is obtained from operational plant data, and is not recorded directly during inspection.

Values for the three measures of shrinkage are plotted against cumulative irradiation. For each set of data both a linear and a second order polynomial line are fitted to the data to provide predictions of future shrinkage. An example of such a manually produce prediction is shown in figure 2.

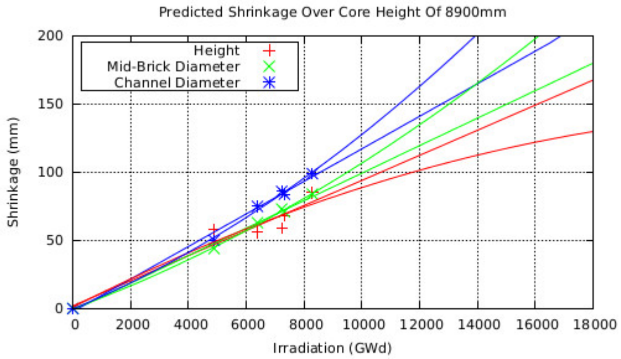


Figure 2. An example of an existing manual prediction of core shrinkage

From all the predictions, the most conservative estimate is used to inform other activities that require the use of these shrinkage predictions. These predictions are necessarily conservative and though first and second order polynomial models are simplified representations, they do provide suitably useful results.

Currently, the existing data management strategy for this inspection data creates a significant barrier for deploying this as an online system, as the primary use for the inspection data is to provide confidence that the channels inspected are in suitable condition to allow the power plant to be returned to service following an outage, and the management of the data is tailored to optimally fulfill this role. However, there is a project currently underway to improve accessibility to the raw, verified data for use by other functions, such as possible online predictions of core degradation. In order to test proposed prognostic approaches before the live station data is made available, simulated data can be used. This has the advantage that the input data and model of core degradation can be tightly controlled and the state of the core simulated at any given level of irradiation, unlike the operational data which can only provide data up to the current level of irradiation. Using simulated data can provide confidence that robust predictions of future state can be made, provided of course that the degradation model is suitably representative of the actual core degradation.

A series of case studies have been developed to demonstrate the possible online prognostic capability that could be achieved if validated inspection data were made available. It is recognized that several assumptions and simplifications have been made in these case studies, and that the resulting predictions of core shrinkage should not be taken literally, but instead as a means of demonstrating the techniques. Furthermore, the approach has been adopted to mimic the existing manual approach in the first instance, though recognizing that future work could see other prognostic approaches applied.

4. Case Studies

In order to demonstrate the framework, a case study is presented which uses simulated brick degradation data to explore the application of online prognostics. Though the most common output of a prognostic algorithm is a measure of RUL, the objective of these predictions is to estimate the shrinkage of the core at certain levels of future cumulative irradiation. MATLAB R2012b was used to implement these case studies.

4.1. Approach

Based on the manual analysis mentioned in the previous section a set of simulated data was created which would provide an approximation of the behavior of graphite degradation. It should be noted that the purpose of this data was to generate a suitable volume of data to explore the prognostic algorithms rather than being an accurate representation of the degradation of graphite. Using a suitable 2nd order polynomial, shown in Eq. (1), and approximated from the manual analysis data, a set of graphite shrinkage values was generated for a series of data ranging from no irradiation to 16,000GWd accumulative core irradiation, a reasonable estimate for a lifetime of operation of an AGR core (based on existing analysis).

$$s = -0.00000022559i^2 + 0.01110778967i \qquad (1)$$

A component of noise was added to this signal to simulate errors in measurement. This component was based on a normal distribution and a new noisy component was generated and applied each time the program was run.

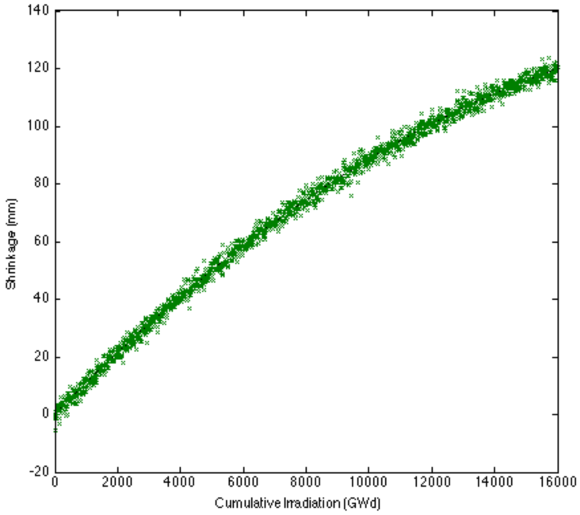


Figure 3. Simulated core shrinkage data including a noise component

Figure 3 shows the output from one such run. A second order polynomial was fitted to the noisy data and was compared to the original simulated data to provide a measure of the quality of fit that would be obtained if all the data were used to make the prediction. In this case a least squares error value of 0.99987 was obtained, reflecting a good fit to the mode from the data. In particular, the value of shrinkage at 16,000GWd was 119.9736mm from the original simulated data and 119.7853mm from the estimate derived from the noisy data, for this particular instance. In general, the difference is in the region of ±0.2% accuracy, which provides an indication of the maximum possible accuracy that could be expected of the predictions.

4.2. Case 1: Train and predict

Using the noisy data set, the first 250 data points were used to fit a 2nd order polynomial and this was used to make a prediction of the shrinkage at 16,000GWd. Figure 4 shows the raw training data used, the prediction and the actual response using an example noisy data set.

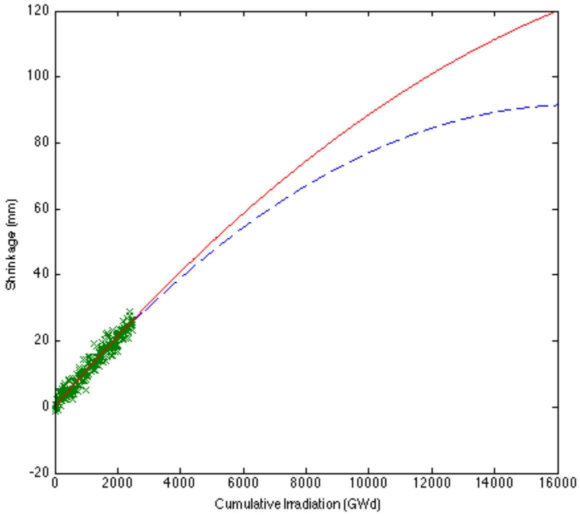


Figure 4. Predicted shrinkage response based on the first 250 measured data points shown in the dotted line. The idealized response is shown as the solid line.

The choice of the initial training set size will have an effect on the early predictions, but with each new measurement beyond the initial training set, the predictive model (2nd order polynomial) could be revised using all available data. For example, Figure 5 shows the progress of the prediction of shrinkage at 16,000GWd as more of the raw noisy data is available.

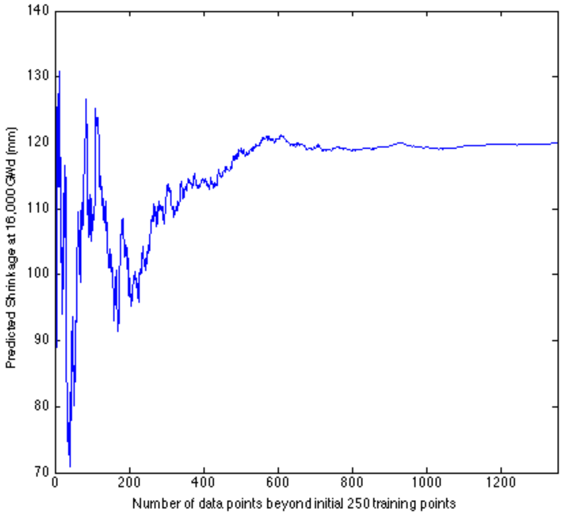


Figure 5. Progress of predicted shrinkage as more points are considered, demonstrating convergence on 120mm

In this instance, after about 600 additional data points above the initial 250 training points (850 data points total), the prediction has converged on the actual final shrinkage of 120mm. The rate of convergence will depend on the initial data set used, and figure 6 illustrates the case where 20 example noisy responses have been generated from the same underlying source signal.

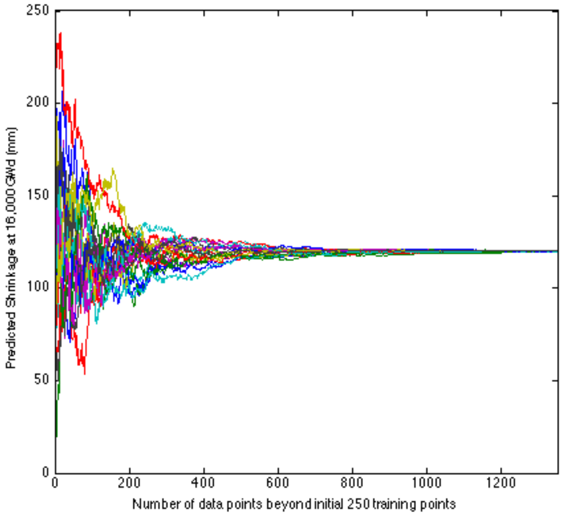


Figure 6. Illustration of responses from 20 different sets of input data

4.3. Case 2: Response to an artificial outlier

The second case investigated was to introduce an artificial outlier to investigate its response to the overall predictions made. Using the same raw data an error of +30mm was introduced to measurement point 900 and this is shown in figure 7.

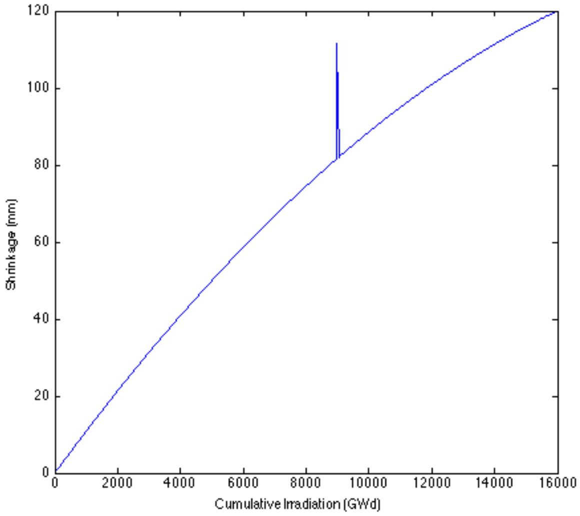


Figure 7. Raw data with an artificial outlier inserted

The corresponding response of how this affected the predicted final shrinkage at 16,000GWd was generated and as expected this did not have a significant effect on the prediction. Figure 8 shows a small perturbation caused by the introduction of the outlier at point 650 on the x-axis, but the influence of this single outlier on the overall prediction is quickly dominated by the contribution of the other data points.

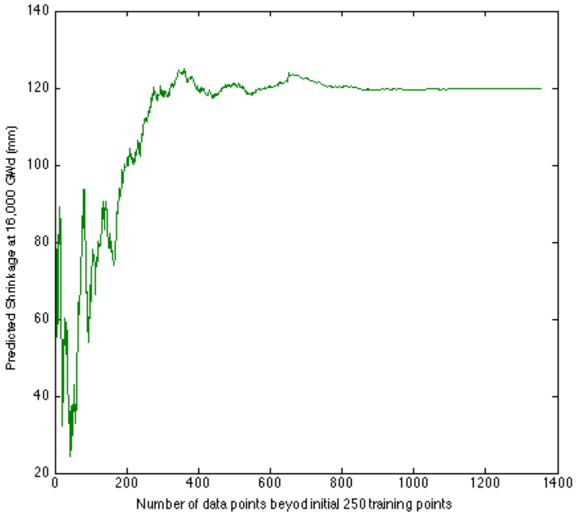


Figure 8. Progress of predicted shrinkage

4.4. Case 3: Simulation of change in underlying model

There are a number of possible reasons that the underlying degradation model may change. For example, a change in the operating temperature of the reactor may affect the underlying shrinkage rate, as might other factors such as the planned injection of methane and carbon monoxide to inhibit the rate of radiolytic oxidation. In other prognostic applications, maintenance actions could also result in a step change in the underlying degradation model.

To simulate a change in underlying model a set of data was generated based on one polynomial model for the first half of the data and switching to a second polynomial model for the second half of the data. Figure 9 shows a plot of the ideal data. The solid line represents the initial polynomial model and is extended through to 16,000GWd. The dotted line simulates a change in the underlying degradation response at 8,000 GWd and shows the new degradation path.

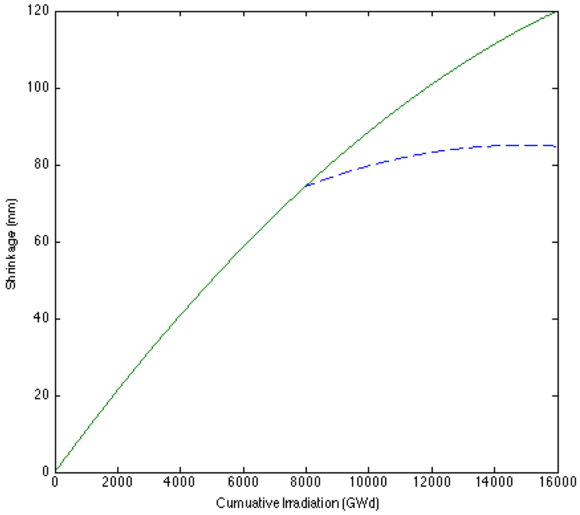


Figure 9. Input data containing a change of underlying model at 8000GWd

As with the previous cases, random noise was applied to the ideal signal, the first 250 points used as training data and a set of predictions calculated based on the available data. The results are shown in Figure 10. As before, the predictions converge towards the 120mm value from the first polynomial model then following the change of model converge towards the second value of approximately 80mm.

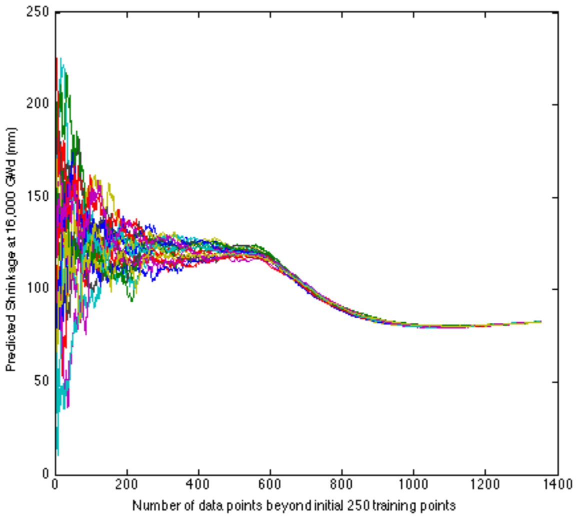


Figure 10. Progress of prediction in response to change of underlying model from 20 sample cases

5. DISCUSSION

The case studies have demonstrated the applicability of prognostics to a continuous feed of measured shrinkage data. A concern with an online prognostic system which continuously updates its prediction as new data is made available is the concern that an erroneous input measurement might significantly alter the prediction. The case study which examined this contained an extreme outlier measurement as an input and though it did affect the final prediction slightly, its affect was minimized by the volume of other measurement data. When there was a change in the underlying model, there was a lag in dealing with the change in response. As an alternative, if the point at which the underlying model changed was known, then a new predictive model could be built from that point forward. However, this new model would require a period of time to build up a suitable number of training points to be able to predict forward accurately. Another option might be to run the prediction based on the full set of available data in tandem with the data from the change point onwards then switch across to the second model once enough seed data has been gathered. It should be noted that for the graphite core, these underlying changes would be very infrequent so this change would not need to be undertaken very often.

5.1. System Development

In order to create an on-line prognostic system the following elements will be required by the system:

- Data management strategy: The prognostics system needs to be able to access the appropriate input data streams as they become available. In particular, the proper QA grade of the input needs to be assured.
- Suitable predictive algorithm
- Appropriate feedback mechanisms: as new data is fed to the system, this data can be used to either increase confidence in the predictions made, can be used to enhance the predictive model or can be used to identify an error in the data gathering. Ensuring the system is robust and able to deal with the possibility that the input data is erroneous is also important.

5.1.1. Data Management Strategy

This is critical to the successful deployment of an online prognostic system. Often data comes from existing monitoring systems, but the prognostic system may require input from a number of different monitoring systems, which were not designed to exchange data. In our case study, two sources of information are required, the cumulative damage and the measures of bore diameter. Neither of these are available as an existing consolidated data source, so this needs to be addressed.

A related issue with developing an online predictive system is the need to deal with historical data. If an online prognostic system were being implemented in a new power plant, then this could be built into the initial design. However, with the AGR power plants (and equally with a large number of other designs of plant) they were built many years ago and the data capture and storage technologies were very different. Including this legacy data is important as it does provide a good baseline reference and useful input to the predictive models.

5.1.2. Choice of predictive algorithm

There is an assumption that the chosen prognostic model is representative of the graphite degradation. In order to be deployed as an online system then some of these assumptions will need to be addressed. Li, Marsden & Fok (2004) describe in detail the relationship between the bore profile (the input to the predictive model) and irradiation induced dimensional change, and should be used as the basis for building a more complete predictive model.

5.1.3. Feedback mechanism

Both the choice of predictive algorithm and also the measure of confidence are important for the successful deployment of an online predictive system. The predictive algorithm should be capable of predicting remaining useful life as the final output of the system, but should also be able to predict interim locations along the degradation curve to allow new operational data to be compared to predictions to assess their accuracy. This provides the opportunity to update the prognostic model if the operational data and predicted data are significantly different. It also provides the opportunity to attribute a measure of confidence to the predictions that are made.

6. Conclusion

The condition of core of the AGR nuclear power plant is a major life-limiting factor and being able to make predictions about its future health is important for continues and extended plant operation. Currently these predictions are undertaken manually. This paper has explored the potential for implementing an online predictive system by providing updated predictions of core shrinkage, a measure of core health, as and when new inspection data becomes available. In the first instance, the prognostic approach closely follows the manual approach to predicting shrinkage as the authors feel that this will ease the transition towards acceptance of an online prognostic approach. However, it is recognized that there are other state of the art techniques which could have application here, and these should be investigated in future. Case studies of an erroneous measurement and a change in the underlying degradation model have been explored through the use of simulated data and a discussion provided as to how these results could be incorporated into a system which would allow online prognostics to be performed.

Acknowledgement

This work was funded by EDF Energy. The views presented by the authors do not represent the views of EDF Energy

References

Brocklehurst, J. E. & Kelly, B. T. (1993), Analysis of the dimensional changes and structural changes in polycrystalline graphite under fast neutron irradiation, *Carbon*, Vol 31 Issue 1, pp. 155-178.

Coble J. B., and Hines, J. W. (2008), Prognostic algorithm categorization with PHM Challenge application, *International Prognostic and Health Management Conference 2008,* p1-11

Coble, J. B., Humberstone, M and Hines, J. W, (2010) Adaptive monitoring, fault detection and diagnostics, and prognostics system for the IRIS nuclear plant, *International Prognostic and Health Management Conference 2010*

Goebel, K., Saha B., Saxena, A., Celaya, J. and Christophersen, J. (2008) “Prognostics in battery health man- agement,” IEEE Instrumentation & Measurement Magazine, vol. 11, no. 4, pp. 33–40

Goh, K.M. (2006), A review of Research in Manufacturing Prognostics, *IEEE Conference on Industrial Informatics*, 16-18 August 2006, Singapore, pp417-422

Hashemian, H. M., Kiger, C. J., Morton, G. W. & Shumaker, B. D. (2011), Wireless Sensor Applications in Nuclear Power Plants, *Nuclear Technology,* Vol. 173, No. 1, pp 8-16

Heng, A., Zhang, S., Tan, A. C. C and Mathew, J (2009) Rotating machinery prognostics: State of the art, challenges and opportunities, *Mechanical Systems and Signal Processing*, Volume 23, Issue 3, p. 724–739

IAEA (2008), Heavy Component Replacement in Nuclear Power Plants: Experience and Guidelines, IAEA Nuclear Energy Series No.NP-T-3.2, Vienna (2008)

Jardine, A.K.S., Lin, D and Banjevic, D. (2006), A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mechanical Systems and Signal Processing*, Volume 20, Issue 7, p. 1483-1510.

Kothamasu, R., S.H. Huang, and W.H. VerDuin (2006), “System Health Monitoring and Prognostics – A Review of Current Paradigms and Practices,” International Journal of Advanced Manufacturing Technology 28: 1012 – 1024.

Li, H., Marsden, B. J. & Fok, S. L. (2004), Relationship between nuclear graphite moderator brick bore profile measurement and irradiation-induced dimensional change, *Nuclear Engineering and Design,* No. 232 pp.237-247

Pecht, M. (2008), Prognostics and Health Management of Electronics, Wiley, Interscience, New York, NY, 2008.

Pecht, M. and Jaai, R. (2010), A prognostics and health management roadmap for information and electronics-rich systems, *Microelectronics Reliability*, Volume 50, Issue 3, p. 317–323

Shennan, J.V. (1983), Graphite R&D reveals long life for AGRS. *ATOM,* No. 323, pp.188-191

Bond, L.J., Ramuhalli, P., Tawfik, M.S. & Lybeck, N.J. (2011), Prognostics and life beyond 60 years for nuclear power plants. *IEEE Conference on Prognostics and Health Management (PHM), 2011,* pp.1,7, June 20-23, doi: 10.1109/ICPHM.2011.6024316

U.S. Department of Energy (2013) Light Water Reactor Sustainability Program: Integrated Program Plan, INL/EXT-11-23452, Revision 1

Varde, P. V. and Pecht, M. G., (2012) Role of Prognostics in Support of Integrated Risk-based Engineering in Nuclear Power Plant Safety, *International Journal of Prognostics and Health Management*, ISSN 2153-2648, 2012 008

West, G.M., Wallace, C.J., Jahn, G. J., McArthur, S. D.J., & Towle, D. (2010), Predicting the ageing of advanced gas-cooled reactor (AGR) graphite bricks. *Seventh American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies,* Las Vegas Nov. 2010.

Biographies

Graeme M. West is a Research Fellow at the Institute for Energy and Environment in the Department of Electronic and Electrical Engineering at the University of Strathclyde. He received a BEng (Hons) degree in Electrical and Mechanical Engineering and a PhD in Electrical Engineering in 1998 and in 2002 respectively, both from the University of Strathclyde.

His current research interests include intelligent system applications and data mining within power engineering and in particular applying the techniques to applications within the nuclear industry.

Christopher J. Wallace is a Research Assistant at the University of Strathclyde researching the application of Multi Agent Systems for data mining and data fusion of condition monitoring data from nuclear power stations. His research interests include the application of AI techniques to condition monitoring and engineering decision support and web technologies for human interface.

Stephen D. J. McArthur received his B.Eng. (Hons) and PhD degrees from the University of Strathclyde in 1992 and 1996 respectively. He is a Professor in the Institute for Energy and Environment, within the Department of Electronic and Electrical Engineering. He is Director of the EDF Energy Advanced Diagnostics Centre. He chaired the IEEE PES Working Group on Multi-agent Systems and the IEEE Intelligent Systems subcommittee. His research interests include intelligent system applications in power, condition monitoring, fault diagnosis, smart grids and multi-agent systems.

Physics-Based Prognostics for LCF Crack Nucleation Life of IMI 685 Aero-engine Compressor Disc

Avisekh Banerjee[1], Jun Zhao[1], Ashok K Koul[1], Amar Kumar[2], Alka Srivastava[2], Nita Goel[2]

[1]*Life Prediction Technologies Inc., 23-1010 Polytek Street, Ottawa, ON, K1J 9J1, Canada*
banerjeea, zhaoj, koula@lifepredictiontech.com

[2]*Tecsis Corporation, 210 Colonnade Road, Ottawa, ON, K2E 7L5, Canada*
amar, alka, nita@tecsis.ca

ABSTRACT

A life cycle management-expert system (LCM-ES) framework is employed in this work for physics-based prognostics of a compressor disc. The modeling approach involves the integration of both global behavior and localized response of component at the microstructural level. This paper presents the results of a low cycle fatigue (LCF) case study for a near alpha titanium alloy (IMI 685) high pressure compressor disc using a microstructure based damage model and finite element analysis results. Both deterministic and probabilistic crack nucleation lives are determined at the two critical locations. The lognormal distributions of α-grain structure of IMI685 and hard alpha (HA) inclusions is considered in the probabilistic analysis, while the deterministic life is predicted based on their extreme values that would represent the worst life. In the LCF modeling, the plastic strain estimation assumes an empirical coefficient that has a strong dependence on the alpha grains and defect size. The proposed life prediction model is capable of capturing the effect of the grain size and hard alpha particle density variation on the LCF crack nucleation life. The worst case deterministic life corresponds well with 0.1% probability of failure and lie around 3542 and 4710 cycles respectively for the primary fracture critical location in the disc.

1. INTRODUCTION

Gas turbine engine (GTE) components are subjected to extreme cyclic loads of different nature, namely mechanical, thermal and environmental during the engine operation. The performance and remaining life of the components reduce progressively because of the structural degradation and this poses a number of challenges. During the engine start-up and shut down, low cycle fatigue (LCF) and thermal mechanical fatigue (TMF) are the two dominant damage mechanisms for materials failure (Joseph and Zuiker, 2006). High cycle fatigue (HCF) arising out of vibrations can cause additional damage thus further shortening the engine life. The HCF accounts for 56% of the major aircraft engine failures and ultimately limits the service life of the most critical rotating components (Lütjering and Williams, 2007; Lütjering, Williams, and Gysler, 2003). The fan and compressor blades in a GTE are prone to HCF failures because of the high mean operating stresses and foreign object damage (FOD) (Leyens and Peters, 2003, Metzger and Seifert, 2012). For efficient design and life estimation of the GTE components, both LCF and HCF effects need to be considered. Relatively larger amplitude, smaller frequency and lower number of load cycles are encountered in the LCF failures while higher frequency, smaller amplitude and larger number of cycles are generally present in the HCF failures. The integration of these two contrasting situations makes any performance and life prediction model complex and cumbersome. Nonetheless, highly reliable and integrated materials- mechanism(s) computational models are required for the performance and lifing analysis. The aero-industries, on the other hand, are in constant demand for improved performance of aero engines by pushing the operating variables like the temperature and structural stress to higher ranges. This trend makes the structural materials more vulnerable to early damage evolution and their faster growth resulting in shorter life and greater risks of failures.

High energy aerospace grade rotor materials are thermal-mechanically processed (TMP) following a number of critical steps within a small window of temperature and strain rate. The TMP follows various heat treatments to tailor specific microstructure and properties in the alloys (SWRI, 2008; Semiatin, Nicolaou, Thomas and Turner, 2008). In titanium alloys, the materials and manufacturing anomalies and defects are observed relating to material/metal flow and cracks/cavities. The hard alpha inclusions potentially degrade the structural integrity of high

Avisekh Banerjee[1] et. al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

energy rotors. The life cycle management of rotors based on either the safe life approach or the damage tolerance philosophy is impacted. A damage tolerance based probabilistic fracture mechanics approach was considered by South West Research Institute (SWRI) to develop DARWIN code to enhance the life of conventional transport aero-engine rotors. The software essentially focuses on the LCF of titanium rotors/discs containing HA anomalies and considering residual stress effects (Lütjering and Williams, 2007; SWRI, 2008).

Our current on-going research focuses on the physics-based component specific technology solution for the assessment of the current damage state and the remaining useful life (RUL). The approach involves both global behavior and localized response of the material at microstructural level. The physical damage evolution and accumulation at micro level in components during operations of the GTE provide the vital inputs for life estimation. The material's response to loading and environmental conditions and experimental/ simulated damage state constitute the basis of our approach. The intended physics-based prognostics solution combines both flight usage and microstructural damage data for reliable prediction of RUL of the GTE. In this paper, our objective is to apply the material microstructure - LCF model for the assessment of deterministic and probabilistic life in titanium alloy. High temperature near alpha (α) Titanium (Ti) alloys, in general are used for demanding applications such as static and rotating gas turbine engine components. The combination of high strength-to-weight ratio, excellent mechanical properties, and corrosion resistance makes titanium the material of choice for many critical applications. The maximum operating temperature for the forged and heat treated Ti alloys in aero-engines has been raised from 300°C to 600°C in the last 50 years (Lütjering, Williams, and Gysler, 2003; Leyens and Peters, 2003). Near alpha high temperature alloy, IMI 685 is considered in this work as a test case. The LCM-ES framework employed in this work incorporates thermodynamics-based off-design engine modeling, computational fluid dynamics (CFD) and heat transfer analysis, finite element method (FEM) analysis and physics-based damage models.

2. ALLOY AND DEFECT CHARACTERIZATION

2.1. Near-α IMI685

Ti alloys with small amount of β stabilizers (< 2 wt%) offer excellent oxidation resistance rather than high temperature creep resistance. Typical applications of the alloy in aerospace industries include airframe skin components and jet engine parts (compressor casing and other parts) requiring high strength at 455°C. Nominal compositions and maximum allowable usage temperature are respectively Ti-6Al-5Zr-0.5Mo-0.25Si and 520°C. The range of microstructures for the IMI685 alloy consists of heavily deformed α–grains along with some spheroidized α to widmannstatten structure to acicular to martensitic structures (Boyer, Welsch and Collings, 1994). Quenching from beta phase fields produces laths of martensitic alpha which are delineated by thin films of beta phase. Ageing causes precipitation of fine alpha phase dispersion, while air cooling from beta phase fields gives a basket weave structure of widmannstatten structure of alpha phase delineated by beta phase (Wanhill and Barter, 2012). A large number of qualitative and quantitative models have been established describing their deformation and fracture behavior as a function of host of material parameters, including microstructure.

For the probabilistic life analysis presented in this paper, the size distribution of alpha grain size in IMI 685 alloy is assumed to be lognormal as displayed in Figure 1. The mean size of the alpha grain is considered as 55 microns (Nag, Praveen and Singh, 2006). The variance resulting from the mean and assumed distribution is assumed to be 22.5.

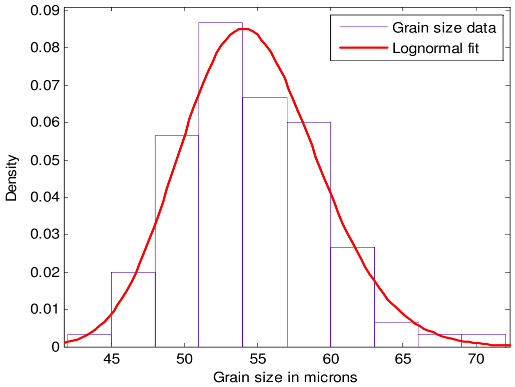


Figure 1: Lognormal distribution of grain size in IMI 685

2.2. HA Defect

Two types of defects (I and II) are known to be highly detrimental for the aircraft-grade titanium alloys. Type I or hard alpha (HA) represents the interstitially stabilized alpha phase with higher hardness and lower ductility as compared to matrix phase. The HA is characterized by excessive concentrations of elements like N (TiN), C and /or O. Type II are abnormally stabilized alpha-phase resulting from segregation of metallic alpha stabilizers, such as aluminum, contain an excessively high proportion of primary alpha and are slightly harder than the adjacent matrix (SWRI, 2008; Semiatin, Nicolaou, Thomas, and Turner, 2008; US patent 4622079). The HA inclusions tend to cause premature LCF crack nucleation. The inclusions are particularly detrimental as they are infrequently and sporadically found in ingot and finished forged products. Excellent coherency between low density HA defects and the matrix in the deformed product makes it difficult to be inspected during NDE. Microporosity as well as microcrackings are always associated with HA imperfections. SWRI has developed computer models for the prediction of the HA size and distribution as well as for diffusion zone (DZ) that correlates well with measured data (SWRI, 2008; Semiatin, Nicolaou, Thomas, and Turner, 2008; McKeighan, Perocchi, Nicholls and McClung, 1999).

For the probabilistic test case analysis presented in this paper, the size distribution of HA defects is assumed to be lognormal as displayed in Figure 2. The mean and variance for the data set are considered to be 17.7 µm and 7.5 respectively to reflect the lower values usually found in thin IMI 685 discs. The two lognormal parameter estimates are µ=2.862 for location and σ = 0.154 for scale. However, much larger HA defect sizes are also reported with much less likelihood of occurrence (Semiatin, Nicolaou, Thomas, and Turner, 2008; McKeighan, Perocchi, Nicholls and McClung, 1999).

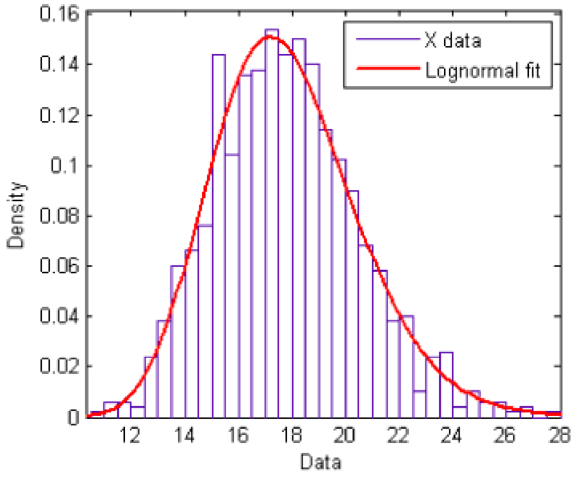


Figure 2: Lognormal distribution of hard alpha defect size in IMI 685

2.3. Alloy Property

The temperature dependent mechanical and thermal properties of IMI 685 used in this work are presented in Figure 3. As the temperature increases, the modulus and tensile strength tend to decrease while thermal conductivity and coefficient also decrease. The Low Cycle Fatigue (LCF) life data at a temperature of 500°C was also obtained from existing literature as shown in Figure 4 (Lütjering, Williams, and Gysler, 2003; Leyens and Peters, 2003). The proposed microstructure based damage model described in section 3.2 was also calibrated with this data. An average grain size of 55µm and hard alpha particle density of 0.057 per unit area was also used (Nag, Praveen and Singh, 2006; Ramachandra, Verma and Singh, 1988).

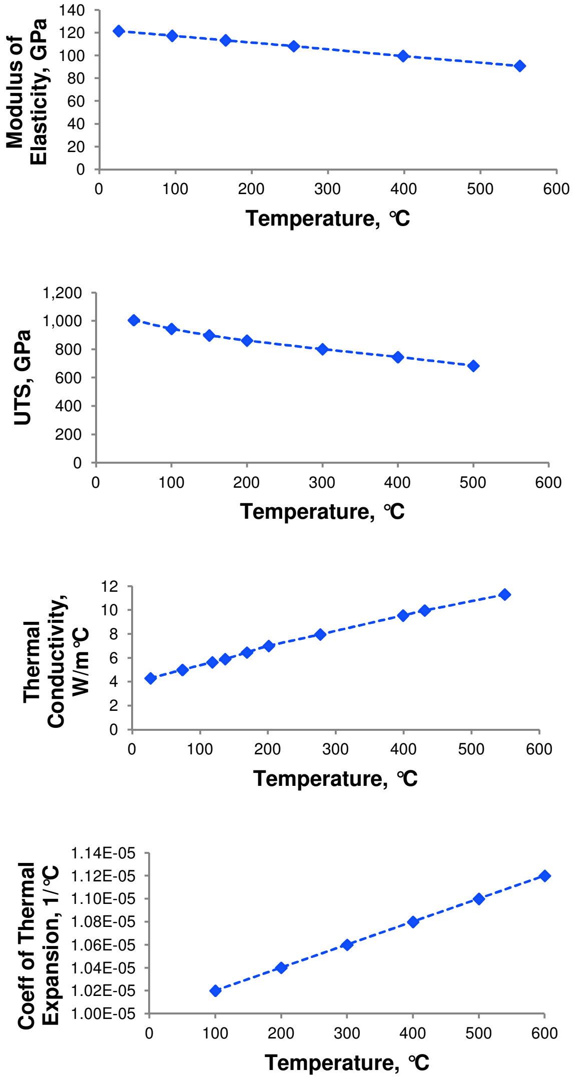


Figure 3: Temperature dependent mechanical and thermal properties

3. MODELING APPROACH

The approach here aims to integrate two levels of materials response, e.g. global and local to external stress under the influence of environmental conditions. The continuum mechanics approach for deformation and fracture is combined with localized failure and microstructural variations and dimensions (micro-mechanics). The localized behavior in the model is assumed to be controlled by two microstructural parameters, namely the alpha grain size and the inclusion size and distribution. To account for the global behavior and damage accumulation, the total strain is considered in the analysis. Following sections outline the salient features of both in the light of the prognostics and life management issues.

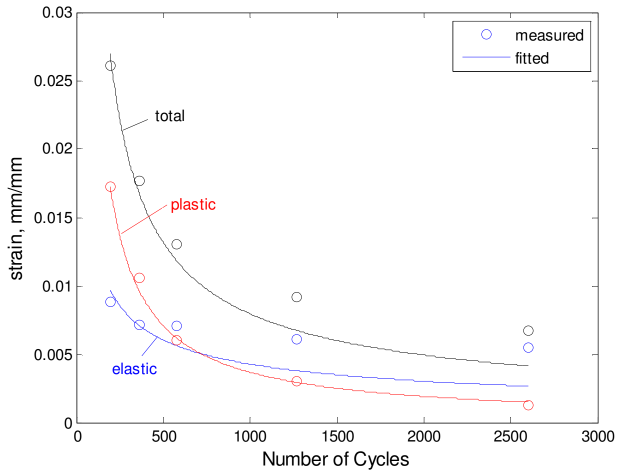


Figure 4: LCF data for IMI 685 at 500°C (Nag, Praveen and Singh, 2006)

3.1. Physics-based Prognostics

A patented Life Cycle Management-Expert System (LCM-ES) framework for true physics-based prognostics is shown in Figure 5. It incorporates the engine cycle thermodynamics analysis, computational fluid dynamics (CFD) and heat transfer analysis, finite element method (FEM) analysis, and materials science based damage models.

In this framework, the engine operational data obtained from the actual or intended usage of the engine is filtered into damage loads based on embedded metallurgical rules. The off-design engine modeling module is incorporated in the proposed system and built using the engine design parameters, this module is capable of generating the boundary conditions for heat transfer analysis for accurate temperature prediction over the engine components. A coupled thermal-structural FEA module is also included that is capable of obtaining the stress and strain states of each component that is in turn used as an input for the microstructural damage modeling module that predicts the mission profile based remaining life of each component. The same damage models can also be used to conduct probabilistic analysis that would allow the prediction of component reliability upfront as a function of the microstructure variability from one material to another. The inputs required to run the prognostics analysis include the component geometry and mesh files, material information and on-design engine design parameters. The lifing solution combines both flight usage and microstructural damage data for reliable life prediction.

3.2. Proposed Damage Model

Proposed microstructure based model is essentially strain based. The total strain ($\frac{\Delta\varepsilon_t}{2}$) based model with the elastic component ($\frac{\Delta\varepsilon_e}{2}$) modeled using Mason-Coffin equation and the plastic component ($\frac{\Delta\varepsilon_p}{2}$) modeled using the

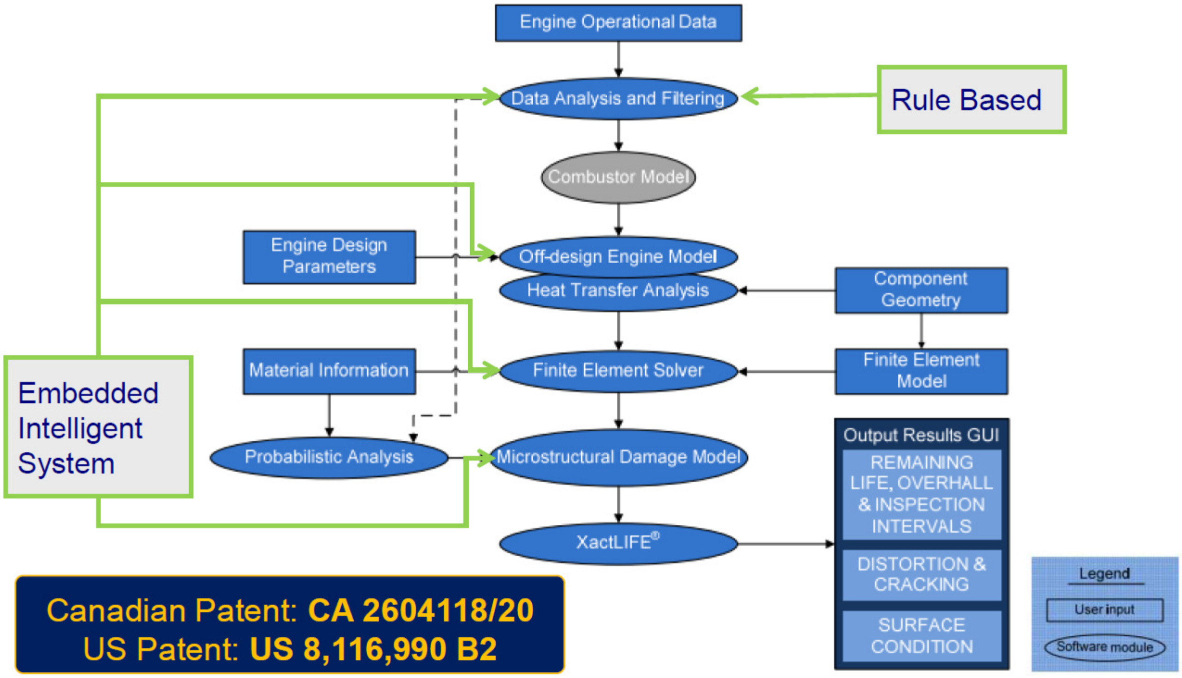


Figure 5: Life Cycle Management - Expert System for physics-based prognostics

microstructural parameters (Koul, 1998). The model is represented as below,

$$\frac{\Delta\varepsilon_t}{2} = \frac{\Delta\varepsilon_e}{2} + \frac{\Delta\varepsilon_p}{2} \tag{1}$$

The elastic component of the stress is given below,

$$\frac{\Delta\varepsilon_e}{2} = \acute{\varepsilon}_f 2{N_f}^c \tag{2}$$

where $\acute{\varepsilon}_f$ and c are the fatigue coefficient and exponent, respectively and N_f is the number of loading cycles to crack nucleation. The plastic component is represented as below,

$$\frac{\Delta\varepsilon_p}{2} = \frac{A(n,d)}{\sqrt{2N_f}} \tag{3}$$

where d is the grain size, n is the number of hard alpha (HA) particles per unit area and A is an empirical coefficient. The probabilistic analysis will involve the variation of the microstructural parameters within their distribution range at a life limiting node identified with deterministic analysis.

4. CASE STUDY

For the purpose of a case study, a disc that is made of IMI 685 has been selected that belongs to the High Pressure Compressor (HPC) section of an aero-engine with a thrust rating of around 30kN. The LCF life was determined for a typical operating condition of the engine operation that would represent a simple mission with take off-cruise-landing. The cruise speed, altitude and the rpm (10,000 rpm) of a 30kN thrust aero-engine was taken into consideration for the off-design modeling to predict the mean-line temperature and pressure of the engine core that would serve as boundary conditions for subsequent thermal analysis. The following tasks were performed as per the LCM-ES framework requirement described earlier using a physics-based prognostics system called XactLIFE.

4.1. Geometry and Finite Element Modeling

In the first step the 3D CAD model of the compressor disc was created and its finite element model generated using structured mesh as shown in the Figure 6. A single segment model was also created and periodic symmetry to reduce the computational expense. A structured mesh with 13,280 brick elements and 60 wedge elements were generated and five single segments used for FEA with periodic constraints to reduce the computational time while preserving accuracy.

4.2. Thermal Analysis

The mean-line temperature and pressure predicted through thermodynamics based off-design model was converted into metal temperature over the five dovetail segment disc sector as shown in Figure 7. The temperature at the bore, rim and the left and right welds of the disc were computed with the help of heat transfer analysis using a bladed disc model.

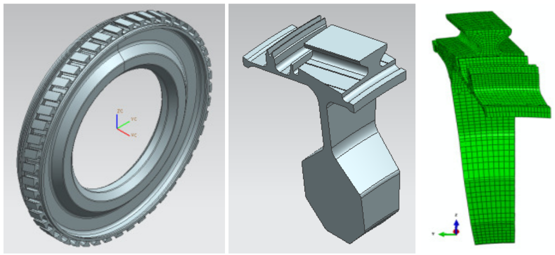


Figure 6: Geometric and finite element models

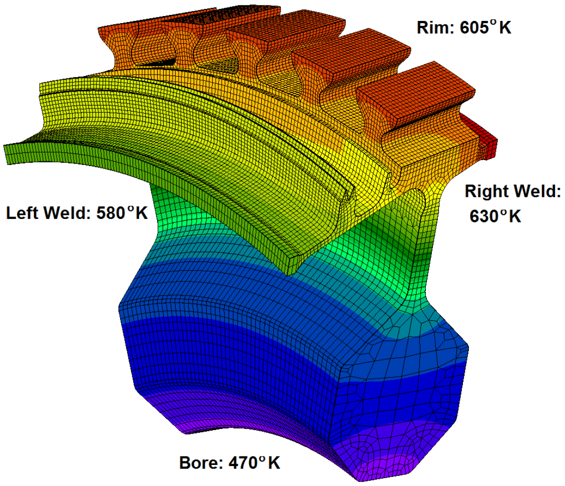


Figure 7: Predicted temperature profile

4.3. Finite Element Analysis

The finite element analysis (FEA) with a coupled steady-state thermal and structural analysis was conducted. The thermal loads obtained from the heat transfer analysis were combined with the centrifugal load and the equivalent stresses and strains of the bulk material were computed. The boundary conditions representing the actual assembly condition were applied along with temperature dependent material properties for accurate results. The predicted von Mises stress profile is shown in Figure 8 where the high stress regions at the sides are due to the boundary constraints and will be ignored while computing the LCF life.

4.4. Deterministic Life Prediction

In the final step the deterministic LCF life to crack nucleation is predicted using the proposed model and the strain result obtained from the FEA. The extreme values of the grain size and hard alpha density has been selected from the lognormal distribution shown in Figure 1 and 2. The distribution of the LCF crack nucleation life over the disc is shown in Figure 9.

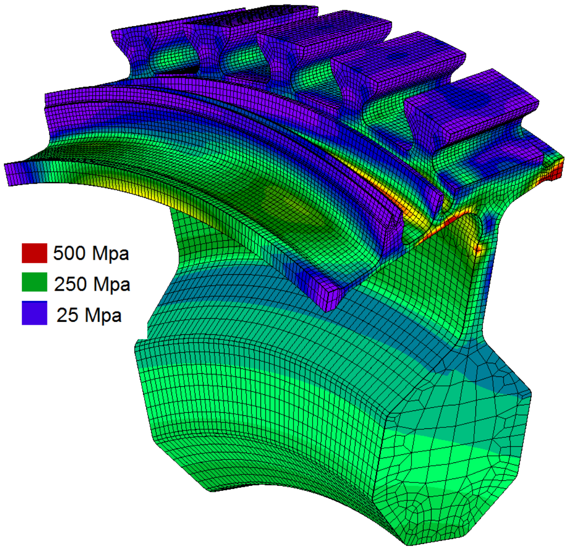


Figure 8: Predicted von Mises stress profile

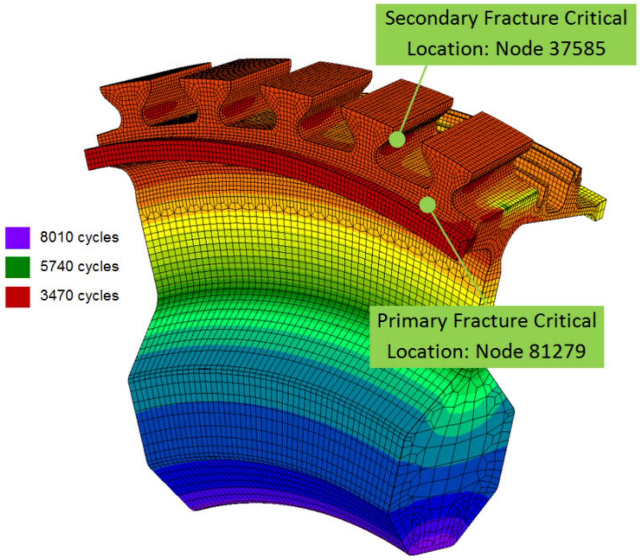

Secondary Fracture Critical Location: Node 37585
Primary Fracture Critical Location: Node 81279
8010 cycles
5740 cycles
3470 cycles

Figure 9: Predicted LCF life to crack nucleation over the disc

It can be seen from the figure that apart from the boundary region corresponding to the rear right weld, the minimum life is in the serration region. At these fracture critical location, the minimum life to crack nucleation was estimated to be 3,542 and 3,639 cycles at the nodes 81279 and 37585, respectively as shown in the figure above. The location of these life limiting regions corresponds to the high stress regions at the rear right welded region and the disc serrations. In addition to the stress, the temperature may also be high due to the maximum temperature being at the disc rim.

4.5. Probabilistic Life Prediction

The grain size and the hard alpha particle density are the two microstructural parameters used in the proposed LCF damage model. A lognormal distribution of these parameters as per Figures 1 and 2 are applied on the primary fracture critical location at the primary and secondary fracture critical locations corresponding to the node numbers 81279 and 37585 respectively, and the result are shown in Figures 10 and 11. The Figure 12 also shows the variation of the life with a combined lognormal distribution of grain size and hard alpha particle distribution. The Figure 10 and Figure 11 suggest a more significant effect of variation of LCF with the distribution of the grain size compared to that of the hard alpha particle density. This observation may be true for thin discs but the distribution of the hard alpha particle density is more difficult to estimate due to the manufacturing process variability. An accurate estimation will be required for reliable probabilistic lifing analysis.

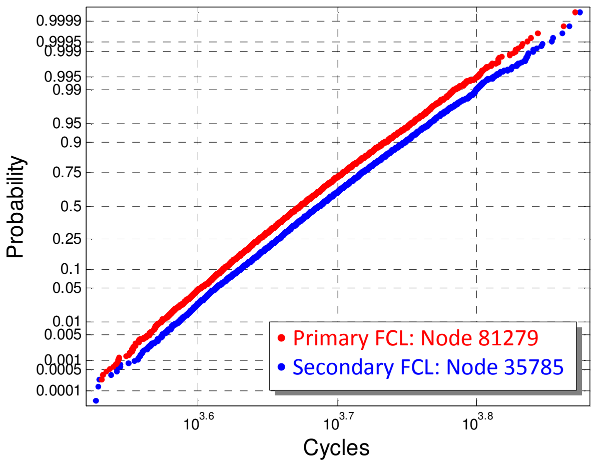


Figure 10: Lognormal probability distribution of life with variation in grain size at primary and secondary fracture critical locations (FCL)

The probability plots show that although the variation in the grain size and the hard alpha particles have different effects on the LCF crack nucleation of the IMI 685 disc, but the primary fracture critical location node has consistently lower life than the secondary fracture critical location node. To study the reliability of the life prediction approach, the deterministic crack nucleation life for the extreme combination of the grain size and the hard alpha particle density that would result in the worst life and the probabilistic life for failure of 1 in 1000 components are compared in Table 1. It can be seen that the deterministic life for both the fracture critical locations are comparable with the probabilistic life computed with the variation of both the microstructure parameters, although the effect of the grain size is more significant on the probabilistic LCF life prediction. But the combined effect of the variation of the grain size as well as the hard alpha particle density will provide the best result compared to the deterministic life prediction. The majority of the distributed values for the two microstructural parameters are quite different than the extreme values selected for the deterministic life prediction, making the probabilistic life consistently higher. Hence the effect of the variations in both the microstructure parameters has to be considered for computing the probabilistic life. The probability plots and the above table also suggest that the effect of the grain size and hard alpha particle density variation on the LCF crack nucleation life has been sufficiently captured through this proposed life prediction model.

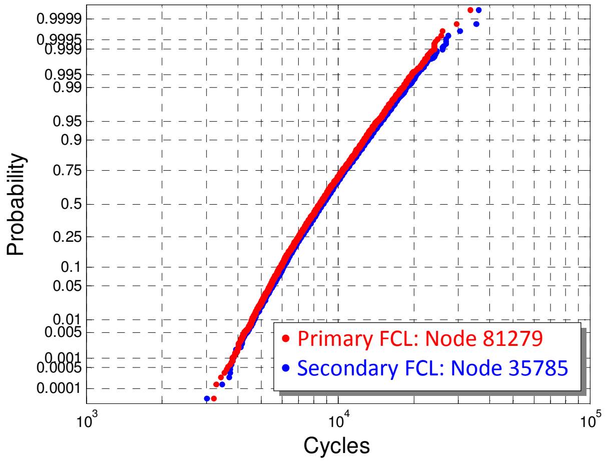


Figure 11: Lognormal probability distribution of life with variation in hard alpha particle density at primary and secondary fracture critical locations (FCL)

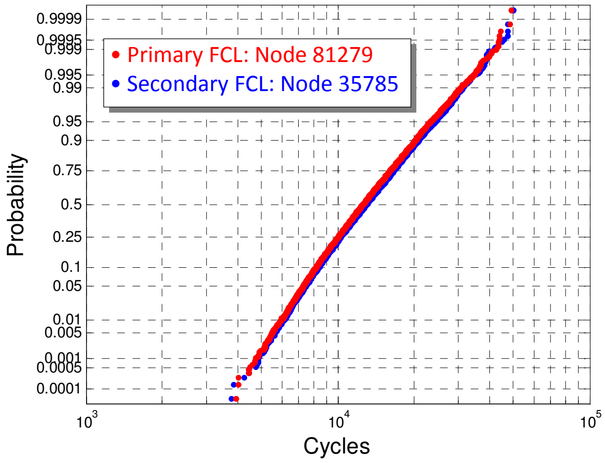


Figure 12: Lognormal probability distribution of life with variation in grain size as well as hard alpha particle density at primary and secondary fracture critical locations (FCL)

Table 1: Comparison of deterministic and probabilistic life at primary and secondary fracture critical locations (FCL)

	LCF Life	
Type of Calculation	**Primary FCL**	**Secondary FCL**
Deterministic	3,542	3,639
Probabilistic @ 0.001 for grain size variation	3,482	3,599
Probabilistic @ 0.001 for hard alpha density variation	3,820	3,801
Probabilistic @ 0.001 for grain size and hard alpha density variation	4,710	4,827

5. CONCLUSIONS

Physics-based prognostics approach has been applied to determine the LCF life for crack nucleation of an IMI 685 high pressure compressor (HPC) disc in a drum assembly. Thermal and structural analysis was performed on a representative disc sector to calculate the stress, strain and temperature profiles over the component. The two fracture critical locations are identified, namely the primary at the disc web and the secondary at the disc serration and LCF life analysis is reported for both.

A microstructure based damage model is proposed with grain size and hard alpha inclusions distribution as input parameters. The lognormal distributions for both these parameters are considered. Deterministic LCF life to crack nucleation computed using the extreme grain size and hard alpha particle density values that would result in the worst life is observed to be closely matching with the predicted probabilistic life at 0.001 probability of failure, suggesting the high reliability of the proposed model. This trend is also observed for both the primary and secondary fracture critical locations and also for the input microstructure parameters distributions in isolation as well as in combination. The result suggests that the grain size distribution has a stronger effect on the probabilistic LCF life in thin IMI 685 Discs.

REFERENCES

Joseph, R. I & Zuiker, R (2006). In *High Cycle Fatigue-A Mechanics of Materials Perspective* by Theodore Nicholas, Elsevier Ltd.

Lütjering, G. & Williams, J., C. (2007). *Titanium-Engineering Materials and Processes*, Second ed. Springer.

Lütjering, G., Williams, J. C. and Gysler, A. (2003). *Microstructure and mechanical properties of titanium alloys*, Titanium, Springer.

Leyens, C. & Peters, M. (ed.), (2003). *Titanium and Titanium Alloys - Fundamentals and Applications*, Wiley-VCH Verlag GmbH & Co. KGaA.

Metzger, M. & Seifert, T., (2012). *A Mechanism-Based Model for LCF/HCF and TMF/HCF Life Prediction: Multiaxial Formulation, Finite-Element Implementation and Application to Cast Iron*", TECHNISCHE MECHANIK, 32, 2-5.

Southwest Research Institute (2008). University of Texas at San Antonio, Mustard Seed Software, GE Aviation, Honeywell, Pratt & Whitney, Rolls-Royce Corporation (2008), Turbine Rotor Material Design, Phase 2 Final Report, FAA Grant 99-G-016, Federal Aviation Administration, Washington, DC.

Semiatin, S.L., Nicolaou, P.D.,Thomas, J.P. and Turner, T.J., (2008). *Defect occurrence and modeling for the thermo-mechanical processing of aerospace alloys*, AFRL-RX-WP-TP-2008-4336, Air Force Research Lab., Materials and Manufacturing Directorate, Wright Patterson Air Force base, OH, 45433-7750, Air Force Material Command.

Boyer, R., Welsch, G. & Collings, E.W. (1994). *Materials properties handbook: Titanium alloys*, eds. ASM International, Material Park, OH.

Wanhill, R., & Barter, S. (2012). *Fatigue of beta processed and beta heat treated titanium alloys*, Springer science and Business Media.

Winston H. C., Robert, A. S., Joseph A. S. (1986). *A Method for the dispersion of hard alpha defects in ingots of titanium or titanium alloy and ingots produced thereby*, US patent publication number, US4622079 A, GE Publication.

McKeighan, P. C., Perocchi, L. C., Nicholls, A. E., McClung R. C. (1999). Characterizing the Cracking Behavior of Hard Alpha Defects in Rotor Grade Ti-6-4Alloy, http://www.darwin.swri.org/html_files/pdf_docs/pubs/1 999 /tms98.pdf

Nag, A. K., Praveen . V. U. K. and Singh, V. (2006). *Low cycle fatigue behaviour of Ti – 6Al – 5Zr – 0·5 Mo – 0·25Si alloy at room temperature*, Bull. of Mater. Sci., Vol. 29, No. 3, pp. 271 – 275. Indian Academy of Sciences.

Ramachandra, C., Verma, V. K. and V. Singh, V. (1988). *Low cycle fatigue behaviour of titanium alloy 685*, International Journal of Fatigue, Vol. 10, Issue 1, 21–26.

Koul, A. (1998). *Some aspects of mechanistic modeling of creep, fatigue and fracture*, Proceedings of Advanced Material Technologies, Banaras, India.

Dr. Avisekh Banerjee is Senior Mechanical Engineer at Life Prediction Technologies Inc. (LPTi), Ottawa, ON. He is working on the development of diagnostics and prognostics tools for turbo-machinery and avionics. His area of research and interests is diagnostics, prognostics and data trending for failure detection and development of PHM framework.

Dr. Jun Zhao is Senior Mechanical Engineer at Life Prediction Technologies Inc. (LPTi), Ottawa, ON. He is working on the Stress analysis of Gas turbine hot section components. His area of research interests is Materials performance and Finite Element Analysis.

Dr. Ashok Koul is the President of Life Prediction Technologies Inc. (LPTi), Ottawa, ON and also acts as an overall technical advisor. He has 25 years experience in the field of materials engineering and life prediction with extensive experience in managing and supervising research and development activities in gas turbine structures, materials and life cycle management strategies. Over the years he has made key contributions in identifying and applying existing as well as emerging technologies in the field of gas turbine engineering.

Dr. Amar Kumar has more than 25 years of research and consulting experience in the fields of structural materials characterization and development, fracture mechanics, failure analysis and applications. Dr. Kumar is currently working as senior research scientist in the R&D project of diagnostics, prognostics and health management of aeroengine components. He specializes in both data driven approaches and physics-based modeling and simulations.

Alka Srivastava graduated with a B.A.Sc. (Electrical Engineering) from the University of Ottawa. She joined Nortel Networks in 1990 as a Member of Scientific Staff. She joined Tecsis Corporation in 2003 as a member of the Research and Development team. Her research interests are in the areas of project management, software quality assurance, fault tolerant computing, numerical analysis and statistical methods.

Nita Goel graduated with a B.E. (Electronics & Telecommunications in 1985 from the University of Jodhpur, India. She completed her M.A.Sc (Electrical Engineering) from the University of Ottawa in 1994. She founded Tecsis Corporation in 1994 and is currently engaged in Research and Development. Her research interests are in the areas of software engineering, fault tolerant computing, numerical analysis and statistical methods.

A New Prognostic Approach for Hydro-generator Stator Windings

Normand Amyot, Claude Hudon, Mélanie Lévesque, Mario Bélec, France Brabant, François-Xavier Frenette

Institut de recherche d'Hydro-Québec (IREQ), Varennes, QC, J3X 1S1, Canada
amyot.normand@ireq.ca , hudon.claude@ireq.ca , levesque.melanie@ireq.ca , belec.mario@ireq.ca , brabant.france@ireq.ca
frenette.francoisxavier@ireq.ca

ABSTRACT

Significant improvements in hydro-generator diagnostics were achieved, in the past decades, by using continuous online measurements and a number of periodic tests. In recent years, the diagnostic raw data has been converted into more useful information by way of integrated diagnostic systems that used expert knowledge. For example, an integrated methodology for hydro-generator diagnostics was developed at Hydro-Québec's research institute (IREQ) using a Web-based application. This comprehensive diagnostic system gives the degradation state of generator stator winding insulation by using a portfolio of diagnostic tools. Combining the results leads to a health index ranging from 1 (good condition) to 5 (worst condition). This system is used by Hydro-Québec's power plant managers as well as technical support and maintenance engineers in the context of condition-based maintenance (CBM). The next step of development is to add new prognostic-related features. This involves automatic identification of active failure mechanisms, root cause analysis and estimation of the stage of advancement of any active mechanism. These characteristics form the basis of predictive maintenance and support the optimization of maintenance strategies.

The approach is based on a number of causal trees (the failure mechanisms) formed by the combination of sequential physical degradation states that ultimately lead to a failure mode. Each combination of sequential physical states is unique and defines a particular failure mechanism. Failure mechanism analysis was followed by identification of all symptoms (diagnostics measurements, observations) with their respective thresholds defining each physical state.

This paper presents the development of a prognostic approach where the modeling of failure mechanisms is combined with observable symptoms from our diagnostic system for the identification of active failure mechanisms.

1. INTRODUCTION

In the case of hydro power generation plants, most of the forced outage time is due to the hydro-generator. Within the generator, the stator winding is the most critical part as it accounts for more than two thirds of the major failures as can be found in the 2003 CIGRE survey on hydro-generator failures.

For years, Hydro-Québec has adopted a maintenance strategy based on three types of maintenance: corrective, time-based and, more recently, condition-based, the last directly linked to diagnostic tests. An integrated generator diagnostic system implemented in 2008, provides information about the actual overall condition of all generators stator windings. This system ranks their condition for all Hydro-Québec power plants. The health index (I) ranges from 1 to 5, the highest being the worst condition (Hudon, Bélec, Nguyen, 2009). This information is used to prioritize the generators for maintenance. However, it does not suggest any particular maintenance action that should be performed in order to mitigate specific failure mechanisms affecting a generator stator.

In the past, a number of authors have worked on degradation state diagrams as a prognostic approach for maintenance optimization for hydro-generators and other equipment (Anders, Endrenyi, Ford & Stone, 1990; Sim & Endrenyi, 1988; Welte, 2009). Figure 1 shows an example of state diagram adapted from Endrenyi et al. (2001).

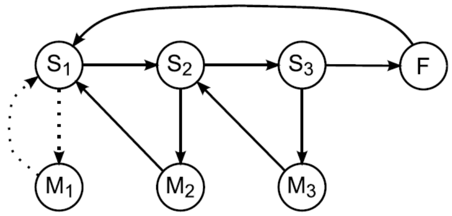


Figure 1. State diagrams including maintenance states (M1-M3) for a failure (F) following a three stages process (S1-S3). Endrenyi et al. (2001)

Normand Amyot et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

These state models were based on Markov or semi-Markov processes. However, they did not take into account the real physical states that can be identified by conducting a failure mechanism and symptoms analysis (FMSA) or a causal tree analysis as described in standard ISO 13379-1 on condition monitoring and diagnostics of machines (2012). They rather used health indices to characterize their degradation states as good, fair or bad for example. This type of approach does not lead to the identification of the specific maintenance action to perform within a particular physical state.

The approach taken in this work is to identify the specific failure mechanisms in play for any given hydro-generator unit in order to take the proper maintenance action. Currently, this requires that knowledgeable experts study all observable symptoms and relate them to all possible degradation mechanisms through the identification of the related physical states. Much of this tedious work could be performed by an automated prognostic tool.

In the context of existing prognostic approaches described by Byington, Roemer & Galie (2002) and shown in Figure 2, the proposed approach would fit the upper part of the prognostic approach hierarchy. It can be considered as model-based as it uses knowledge-rich information provided by a diagnostic portfolio that accounts for physical degradation states.

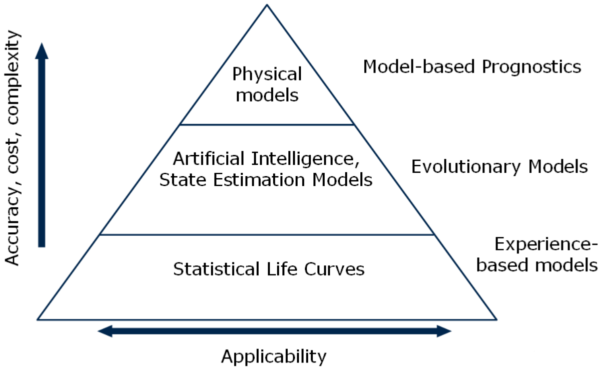


Figure 2. Hierarchy of prognostic approaches. Adapted from Byington et al. (2002).

Figure 3 illustrates how different maintenance strategies can coexist depending on the condition of components and subcomponents and information available about them. Since maintenance can be optimized through a predictive failure mechanism and symptoms analysis, the present work systematizes this approach. It is based on an analysis of the possible failure mechanisms for hydro-generator stator windings that was carried out by Nguyen & Yelle (2001). These failure mechanisms lead to one of the seven failure modes listed in Table 1. A failure mode is defined as the final stage of a failure mechanism, after which the equipment can no longer perform its function.

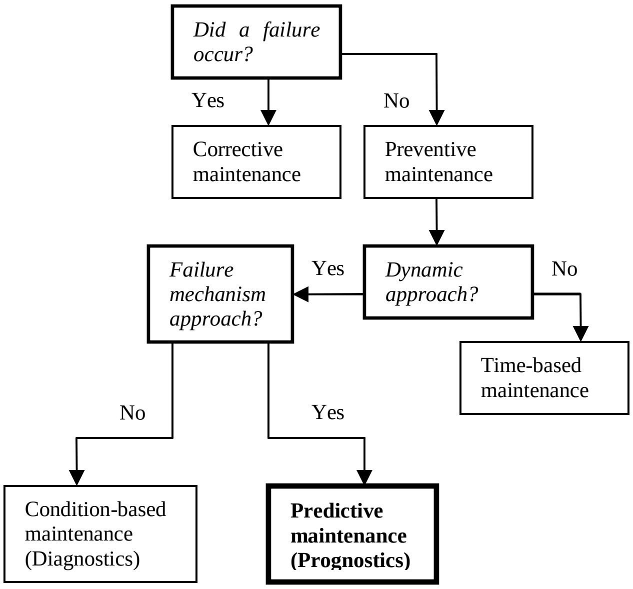


Figure 3. Existing maintenance strategies and their interrelations.

Table 1. Hydro-generator failure modes

f1	Phase-to-phase breakdown
f2	Phase-to-ground breakdown in the slot
f3	Phase-to-ground breakdown outside the magnetic core
f4	Excessive rotor vibration
f5	Loss of magnetic field
f6	Melting of damper bar
f7	Stator electrical connection failure

2. FAILURE MECHANISMS ANS SYMPTOMS ANALYSIS

A failure mechanism is any physical, chemical or other process that leads to failure. For generators, it originates from one or a combination of four stresses: Thermal, Electrical, Ambient and Mechanical (TEAM). As illustrated in figure 4, under these stresses, root causes are responsible for initiating the failure mechanisms in the same way as in a causal tree such as described in standard ISO 13379-1 (2012). Failure mechanisms result in a sequence of events leading from one physical state to the next. In this model, each physical state is labeled according to its stress category (for example e3 is an electrical process). Each sequence in Fig. 2 leads to a potential failure mode (f1…f6). Several mechanisms may sometimes be active at the same time, but only one will lead to failure. Each potential failure mechanism is defined by a unique sequence of physical states. For example, figure 4 shows three possible failure mechanisms given the available symptoms: (T1,t1,…,t4,f1), (T1,t5,…,e3,f3) and (A3,a6,..,e3,f3).

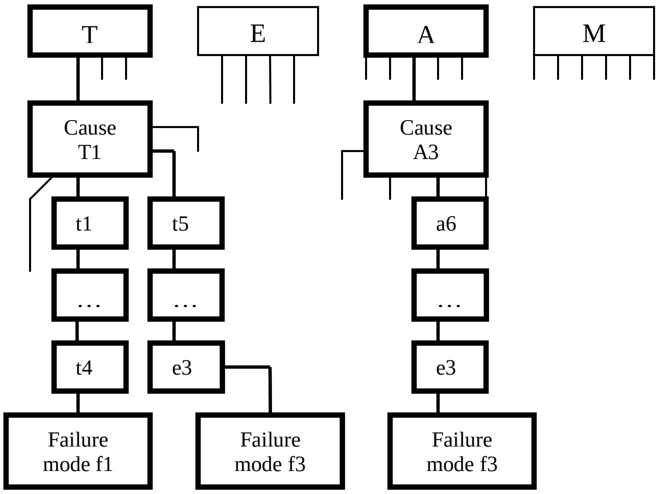


Figure 4. Three active failure mechanisms. Causes are in upper case and physical states are in lower case.

Table 2 shows, by category of stress, all root causes for each of the 111 failure mechanisms that were identified. Of these failure mechanisms, 8 are initiated by root causes related to thermal stress, 8 to electrical stress, 35 to ambient stress and 60 to mechanical stress. The number of intermediate physical states in these failure mechanisms is given in Table 3 for each type of stress category.

3. Prognostic Model

The prognostic model is based on automatic identification of the physical states from available symptoms. One specific set of symptoms, with their respective thresholds to comply with, defines one single physical state. Each characteristic symptom comes from the results of diagnostic tool measurements or visual inspections logged into our integrated diagnostic system for generators.

The health index of the generator is computed by combining individual diagnostic results but each diagnostic tool also provides detailed information (symptoms) that can be used to identify the generator's physical state at a given time. For instance, a generator's condition could come from combining partial discharge (PD) measurements, visual inspection input, and polarization/depolarization current. In addition to this overall index, it is possible to mine data to the level of symptoms and determine whether gap discharges, say, are accompanied by visual signs of powder between end arms and, if so, the number of such sites. These symptoms are the key to identifying active physical states. Table 4 overviews the actual diagnostic tools logged in the database and the number of detailed symptoms that each can provide. Note that the scope of such an analysis is currently being expanded to include other characteristics logged in other monitoring system, such as air gap measurement, temperature and vibration analysis.

Table 2. Root causes for failure mechanisms

Root causes per stress category	Number of failure mechanisms
***THERMAL STRESS* (T)**	**8**
T1 Thermal aging (normal operation)	3
T2 Accelerated aging (operation above specified rated temperatures)	3
T3 Aging due to thermal cycling (frequent start/stop operation)	2
***ELECTRICAL STRESS* (E)**	**8**
E1 Improper manufacturing or design of bars	2
E2 Poor semiconducting coating on the straight part of the bars (slot discharges)	1
E3 Poor design or manufacturing of end winding stress grading material (corona discharges)	1
E4 Insufficient spacing between end windings (gap discharges)	1
E5 Overvoltage transients	3
***AMBIENT STRESS* (A)**	**35**
A1 Conducting contamination (carbon, steel or copper dust)	6
A2 Non-conductive contamination (construction dust or oil)	9
A3 Moisture in ambient air	7
A4 Abrasive material attack	3
A5 Water leakage (cooling system failure, fire protection and spills)	10
***MECHANICAL STRESS* (M)**	**60**
M1 Loose windings	17
M2 Bad connection	6
M3 Presence of external objects or loose parts	5
M4 Mechanical shocks	4
M5 Projectiles	4
M6 Rotor and/or stator deformation	24

Table 3. Number of physical states per process

Types of process	Number of physical states
Thermal (t)	9
Electrical (e)	22
Ambient (a)	14
Mechanical (m)	35

Table 4. Diagnostic tools and detailed symptoms.

Diagnostic tool		Number of detailed symptoms
(1)	PD analysis (intensity, number) – PDAH	3-4
(2)	Phase-resolved PD – PRPD	6
(3)	Visual inspection	70
(4)	Polarization/depolarization currents (stator)	4
(5)	Ramped voltage current measurements	5
(6)	Semiconducting coating integrity measurement	2
(7)	Ozone concentration measurements	2
(8)	Dissection (postmortem)	23

Once problematic generators have been identified in our integrated generator diagnostic system (I=5), the prognostic tool analyzes the database for each of them to identify the most probable active failure mechanisms. In order to do so, a prognostic database was built including all potential failure mechanisms (sequences of physical states) and the set of symptoms with their threshold values associated with each physical state. Figure 5 shows an example of failure mechanism with the corresponding symptoms and thresholds defining the physical states.

Active failure mechanisms are identified automatically using the available symptoms obtained from diagnostic tools. A search engine was developed to retrieve the symptoms from the integrated generator diagnostic system and compare them to the defined symptoms with thresholds for each physical state in the prognostic database. The active failure mechanisms proposed by the system are then displayed. The list of active failure mechanisms clearly depends on the data available. When many diagnostic symptoms are available, the search engine usually displays fewer possible failure mechanisms with higher confidence. When only a few symptoms are available for the generator, more possible failure mechanisms are displayed with lower confidence. Work is currently underway on confidence levels to develop a feature that would automatically propose the best test to minimize the uncertainty of failure mechanism identification. Pinpointing this one mechanism is therefore the issue to address first.

4. FUTURE WORK

The main objective of using such a prognostic approach is to improve maintenance strategies. The key is to include in the database all maintenance actions for each physical state in the failure mechanisms. Figure 6 shows the two step process: available symptoms identify the active physical state and for each physical state a maintenance action is defined.

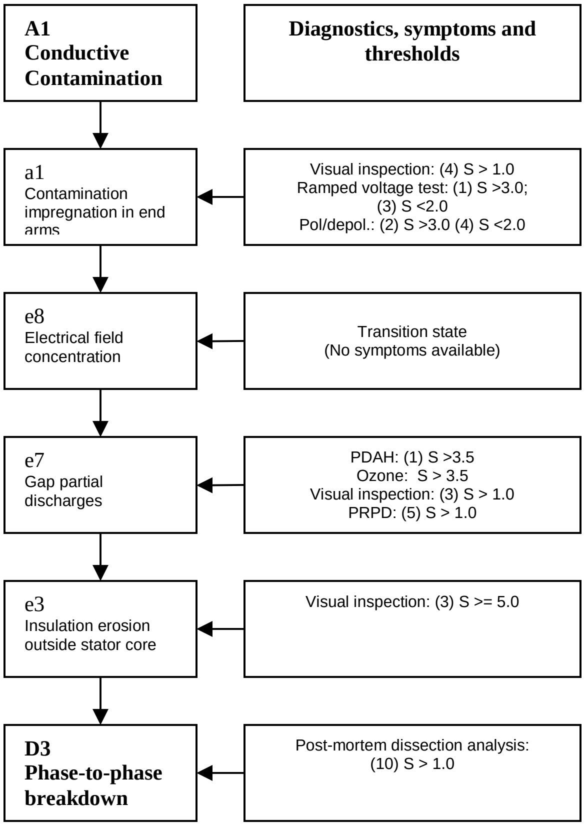

A1
Conductive
Contamination
Diagnostics, symptoms and thresholds
a1
Contamination impregnation in end arms
Visual inspection: (4) S > 1.0
Ramped voltage test: (1) S >3.0;
(3) S <2.0
Pol/depol.: (2) S >3.0 (4) S <2.0
e8
Electrical field concentration
Transition state
(No symptoms available)
e7
Gap partial discharges
PDAH: (1) S >3.5
Ozone: S > 3.5
Visual inspection: (3) S > 1.0
PRPD: (5) S > 1.0
e3
Insulation erosion outside stator core
Visual inspection: (3) S >= 5.0
D3
Phase-to-phase breakdown
Post-mortem dissection analysis:
(10) S > 1.0

Figure 5. Failure mechanism with symptoms and thresholds defining physical states. Between parenthesis are symptoms pertaining to the diagnostics. S is the severity ranging from 1 to 5.

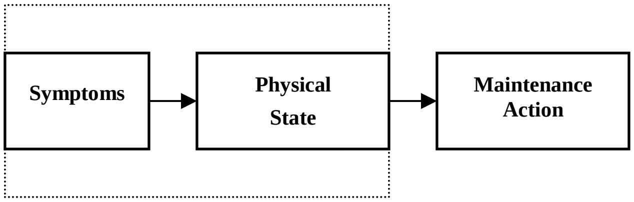


Figure 6. Symptoms identify the active physical state for which maintenance actions are defined.

The doted area shows what has been accomplished yet. Future work will consist in the identification of maintenance actions for each physical state in the prognostic database. This will enable the predictive maintenance strategy.

When an active mechanism is identified as the most critical, the system would thus propose customized maintenance action to solve the specific problem. Every maintenance action for a specific physical state would either completely restore the condition of the generator ("as good as new") or just restore one of its previous physical state ("as bad as old”). The effect of every maintenance action for each state/mechanism must also be included in the database.

A corollary to the effect of maintenance is to determine the transition time between successive physical states as degradation evolves. This feature will allow us to know the proper timeframe for maintenance as well as the impact of performing the job before a failure occurs. Automatic analysis will also be extended to rotor degradation mechanisms and, most importantly, incorporate all economic considerations, e.g., the cost and duration of each maintenance action, and the loss of revenue in the event of a forced outage.

5. CONCLUSION

The development of a prognostic model was initiated in order to optimize future maintenance of hydro-generators. At this stage, the model consists of a database of potential failure mechanisms combined with automatic recognition of active mechanisms from symptoms that define physical states. Embedded in the database are all the criteria used to define the physical states. This may be viewed as a means of capturing expert knowledge. A search engine can already be used to data mine the integrated diagnostic system Web application, and automatically identify and sort failure mechanisms from the data available for each generator.

Many desirable features are not yet implemented, such as relating maintenance actions to physical states and estimating transition times between states, to name but two. Future work will address these features and also broaden the scope to the rotor. The prognostic engine will continue to evolve in the years to come and will be validated by case studies.

REFERENCES

International Council on Large Electric Systems (CIGRÉ) (2003). Hydrogenerator Failures – Results of the survey, *CIGRE Study Committee SC 11, EG11.02 Report*, 129 p., Paris, France: Conseil International des Grands Réseaux Électriques.

Hudon, C., Bélec, M. & Nguyen N.D. (2009). Innovative web system for condition-based maintenance of generators”, *Proc. IEEE Electrical Insulation Confrence* (EIC), pp. 234–245.

Anders, G.J., Endrenyi, J., Ford, G.L. & Stone, G.C. (1990). A probabilistic model for evaluating the remaining life of electrical insulation in rotating machines, *IEEE Trans. on Energy Conversion*, Vol. 5, no.4, pp.761-7.

Sim, S.H. & Endrenyi, J. (1988). Optimal preventive Maintenance with repair, I*EEE Trans. on Reliability*, Vol. 37, No. 1, pp. 92-6.

Welte, T.M. (2009). Using state diagrams for modeling maintenance of deteriorating systems, *IEEE Trans. on Power Systems*, Vol. 24, No.1, pp.58-66.

Endrenyi, J., Arboreshid, S., Allan, R. N., Anders, G.J., Asgarpoor, S., Billingtonn, R., Chowdhury, N., Dialynas, E.N., Fipper, M., Fletcher, R.H., Grigg, C., McCalley, J., Meliopoulos, S., Mielnik, T.C., Nitu, P., Rau, N., Reppen, N.D., Salvaderi, L., Schneider, A. & Singh Ch.. (2001). The present status of maintenance strategies and the impact of maintenance on reliability, *IEEE Trans. on Power Systems*, Vol. 16, No.4, pp. 638-46.

International Standards Organization (ISO) (2012). Condition Monitoring and Diagnostic of Machines – general guidelines on data interpretation and diagnostic techniques, In ISO, *ISO13379-1:2012*, (p.24). Genève, Switzerland: International Standards Organization.

Byington, C.S., Roemer, M.J., Gallie, T. (2002). Prognostic enhancements to diagnostic systemes for improved condition-based maintenance, *Proc. IEEE Aerospace conf.*, Vol 6., pp. 2815-24.

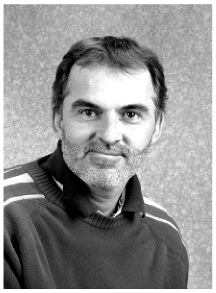
Nguyen, D.N. & Yelle, R. (2001). Analyse des modes de défaillance et de leurs effets sur les alternateurs : Rapport de synthèse, Report IREQ-2001-173.

BIOGRAPHIES

Normand Amyot was born in Sorengo Switzerland in 1964. He received the M.Sc.A. degree in physics engineering from École Polytechnique de Montréal in 1990. He has worked one year at the Université des Sciences et Techniques (USTM) de Masuku in Gabon, after which he joined the Research Institute of Hydro-Québec (IREQ), where he is now employed as a senior research scientist. His field of expertise includes electrical insulation aging and characterization, diagnostic techniques, condition based maintenance and more recently prognostics. He is an active member of CIGRÉ working groups and member of IEEE DEIS. He is the author or co-author of more than 40 scientific papers.

Claude Hudon was born in Montreal, Québec, Canada on 21 December 1963. He received the Ph.D. degree in engineering physics from École Polytechnique de Montréal in 1993. He worked for two years at the Corporate Research and Development center of General Electric, after which he joined Hydro-Québec’s Research Institute, IREQ, where he is now employed as a senior researcher. His fields of interest are generator and motor diagnostics, the development of diagnostic tools and the measurement and analysis of partial discharges. He is the author of more than 45 scientific papers.

Mélanie Lévesque was born in Chicoutimi, Québec, Canada on 24 November 1980. She received the PhD degree in electrical engineering from the École de Technologie Supérieure (ÉTS) of Montréal in 2012. Her PhD's degree was done in collaboration with the Research Institute of Hydro-Québec (IREQ). She is currently working as a researcher at IREQ on generator diagnostics.



Mario Bélec was born in Ste-Anne-du-Lac, Québec, Canada on 20 March 1962. He has received the B.Ing. degree in electrical engineering from École Polytechnique de Montréal in 1987. He joined the Research Institute of Hydro-Québec (IREQ) in 1987 as a researcher. As a generator diagnostic specialist, he has been involved in partial discharge recognition activity, and development of new diagnostic tools and correction methods for high voltage rotating machines. He is also working on a condition-based maintenance diagnostics methodology of generators. During the recent years, he is also working on development of partial discharge monitoring system for gas-insulated system (GIS). He is active member of IEC and CIGRE working groups. He is the author of more than 50 scientific papers.

France Brabant was born in St-Jerome, Quebec, Canada . She received a Bachelor degree in Computer Sciences from the University of Sherbrooke in 1988. Upon graduation she was hired by Hydro-Quebec for Field Testing, particularly in the opening of Post Radisson for the James Bay project. She joined IREQ in 2002 and has done software development for many innovation projects. Her specialities include object technology such as Java, UML, etc., human-machine interfaces, data processing, and programming in a Matlab environment .

François-Xavier Frenette was born in Montreal, Canada in 1991. He will receive, in December 2013, a Bachelor degree in computer science from the Université de Sherbrooke. During his studies, he did two internships at the Research Institute of Hydro-Québec (IREQ), where he mainly worked on developing web interfaces for two projects related to diagnostic tools.

Determination of Primary Chemical Constituents of PBX(AF)-108 Warhead Explosive Using microPHAZIR^{TM} Near Infrared (NIR) Handheld Platform

Sami Daoud[1], Michal J. Villeburn[2], Kevin D. Bailey[3], Gordon Kinloch[4], Lydia Biegert[5], and Craig Gardner[6]

[1,2,3]*Raytheon Missile Systems, Tucson, Arizona, 85734, United States of America*
sami_daoud@raytheon.com
mjvilleburn@raytheon.com
kdbailey@raytheon.com

[4]*United Kingdom Ministry of Defense, Bristol, BS34 8JH, United Kingdom*

DESBVRAAMPIM2a@mod.uk

[5]*Alliant Techsystems Launch Systems, Brigham City, Utah, 84302, United States of America*

lydia.biegert@atk.com

[6]*Thermo Fisher Scientific, Tewksbury, Massachusetts, 01876, United States of America*
craig.gardner@thermofisher.com

ABSTRACT

An innovative prognostics and health management (PHM) technique for quantifying and characterizing health status of Plastic Bonded Explosive (PBX) Air force (AF) formula PBX (AF)-108 warhead explosive was developed using Near Infrared (NIR) spectra emitted by microPHAZIR^{TM} NIR, a handheld platform developed by Thermo Fisher Scientific. Benchtop High Performance Liquid Chromatography (HPLC) was used as a reference technique for correlation to microPHAZIR^{TM} NIR measurements.

Near infrared spectra were acquired from twenty freshly manufactured mixes of PBX (AF)-108 explosive formulae, which were used in setting up a D-Optimal full-factorial design of experiment (DOE). Three-hundred and sixty measurements were recorded and analyzed using Partial Least Squares (PLS) regression analysis for model building and method development. Results were correlated to spectra, which were measured using HPLC reference technique. All recorded measurements performed with microPHAZIR^{TM} handheld platform were successfully validated with HPLC measurements. An algorithm was developed for microPHAZIR^{TM} NIR thus qualifying the platform as a real-time nondestructive test (NDT)/ nondestructive evaluation (NDE) tool for quantification of primary chemical constituents of PBX (AF)-108. Primary chemical constituents of PBX(AF)-108 are Polyurethane (PU) binder, Royal Demolition Explosive (RDX) oxidizer/ fuel, Isodecyl Pelargonate (IDP) plasticiser, and E-702 (4, 4'-methylenebis(2,6-di-tert-butyl-phenol) [MBDTBP] anti-oxidant/stabilizer.

This teaming effort between Raytheon Missile Systems (RMS), United Kingdom Ministry of Defence (UK MoD), Alliant Techsystems Launch systems (ATK LS), and Thermo Fisher Scientific demonstrated outstanding ability to utilize miniature cutting edge technology to perform real-time NDT of PBX (AF)-108 warhead explosive without generating chemical waste and/or residue. The new technique will further be adapted for use to measure primary chemical constituents of other warhead explosives and solid rocket propellants. The new technique will significantly reduce costs associated with performing ordnance surveillance and Service Life Extension Program (SLEP) assessment, which is often destructive and requires use of lengthy and expensive test techniques described in North Atlantic Treaty Organization (NATO) Standardization Agreement (STANAG)-4170 and Allied Ordnance Publication (AOP)-7 manuals.

Daoud et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Tactical missiles are often exposed to severe thermal and dynamic stresses associated with long- term exposure to harsh environments, including transportation handling, transportation vibration, diurnal cycling, and much more. These stresses may act individually or synergistically to factor into the aging, deterioration, and eventual decommissioning of some of Department of Defense (DoD) critical warfighting assets. Adverse reliability associated with long-term aging and deterioration of assets significantly affects the total life cycle cost of fielding these weapon systems in a high state of readiness. Reliability evaluation of legacy data has indicated failures in missile structural, energetic and electronic components, all associated with the long-term exposure to static (heat, humidity, salt, etc.), and dynamic (transportation shocks, vibration, etc.) stressors.

Today, most common methods of NDT for evaluating the health of energetic systems are radiographic (X-ray imaging, X-ray computed tomography (CT), etc.), electrical (Eddy-current and electro-magnetic methods), dye penetrant, and acoustic and ultrasonic, or a combination thereof. These methods are used by manufacturers during the production process, mostly for quality control, and are seldom used once the system has been fielded. Moreover, for military energetic systems, it is usually impractical to use these methods in the field. For health monitoring in the field, deployable or portable platforms become valuable as NDT/NDE tools.

A joint effort was carried out between the UK Ministry of Defence (MoD), Raytheon Missile Systems, ATK Launch Systems, and Thermo Fisher Scientific to qualify microPHAZIR^{TM} NIR platform as a portable real-time NDE tool. The effort was successfully executed and will enable RMS, other defense contractors, US DoD and UK MoD to quantify chemical constituents of PBX (AF)-108 warhead explosive, a high energy plastic bonded explosive composition used in the armament subsystem of tactical missiles. The platform will further be adapted for use to measure numerous other explosives, solid and liquid rocket propellants.

The proposed technology is hybrid, in that it does not provide wireless and/or continuous monitoring of the health status of the energetic material (i.e. PHM), yet it is a proactive NDE/NDT technique which replaces the old destructive test methodologies, described in NATO STANAG-4170 and AOP-7 manuals, imposed by Surveillance and Life Extension Programs (SLEP) of past and present day techniques. The proposed technology will define new means for realizing anticipated residual useful life (RUL) of an explosive from a chemical perspective, by quantifying chemical constituents within the explosive matrix, which can shed valuable information about the anticipated mechanical and structural behavior of the explosive matrix. The combination of chemical and mechanical (structural) health of the explosive determines whether a warhead (armament subsystem) would be warranted as “safe and suitable for service (S3)”.

Today RMS and the UK MoD surveillance strategies seek to extend time between periodic evaluations, henceforth reducing tasks associated with subsystem breakdown, test and criticality analysis (BTCA) by as much as 50%. On average, a surveillance program is often recommended once every 4 years on a sample population which represents the fielded and/or stored weapons inventory, and with the introduction of microPHAZIR^{TM} NIR real-time technology it will be feasible to extend the time between surveillance programs and/or reduce the number of assets that undergo surveillance evaluation. When a SLEP plan is established for warhead (armament subsystem) inventory, complex steps must be executed and comprise disassembly, dissection and extensive testing (physical, chemical, and mechanical tests) of the warhead explosive matrix, often referred to as “breakdown, test and criticality analysis (BTCA)”. BTCA coupled with arena testing (static fire) of the warhead as well as other subsystems are challenging tasks, from manpower, cost and schedule perspectives, and therefore the need to exercise cost controls while at the same time maintain absolute confidence in assets health demands that new technologies such as microPHAZIR^{TM} NIR platform and more advanced (exploratory) technologies become integral part of SLEP cost consciousness and technology readiness. Generally, the ultimate goal is to be able to (i) predict subsystems, and henceforth system anomalies proactively and sufficiently in advance to institute corrective actions and/or preventive measures; and (ii) reduce generated chemical waste, logistics footprint, logistics response time, and life-cycle costs, which will ultimately increase systems availability, and enhance customer-supplier business relationship.

The proposed technology will be adopted by RMS and the UK MoD as the principal means for realizing early warnings of unsafe conditions using real-time data, collected with mobile microPHAZIR^{TM} NIR handheld platform and other advanced technologies of Thermo Fisher Scientific. Gaining real-time knowledge about the current health of an explosive matrix will offer effective insight to predicting future SLEP test plans.

Successful application of microPHAZIR^{TM} NIR handheld platform as a NDE/NDT tool is the cornerstone and the spring board for future development of PHM of energetic subsystems: Cartridge Actuated Devices (CADs), Propellant-Actuated Devices (PADs), and electro-explosive devices (EEDs) of tactical and strategic missiles. microPHAZIR^{TM} NIR handheld platform offers enormous potential for applications requiring real-time monitoring of the health status of warheads and solid rocket motors subject to fatigue, chemical and mechanical (structural) degradation.

2. EXPERIMENTAL

All starting raw materials were obtained commercially, in accordance with (IAW) military (MIL) standard MIL-E-8289 specifications and used without further processing.

2.1. Hazards and Safety Protocol

Hazards data comprising impact, friction, electrostatic discharge (ESD), and differential scanning Calorimetry (DSC) were performed on two uncured 10- gram high-speed mixes with the highest and lowest solids DOE iterations (worst and best case scenarios, respectively). Cured 1⁄4-pint (120 gram) mixes of the high solids loaded and low solids loaded PBX (AF)-108 mixes were prepared to identify safety concerns during mixing and preparation of the one-pint (600-gram) mixes and for handling and storage of the cured material. All hazards data were concluded to be low-order and in conformance with actual production mixes. Most importantly, ATK Launch Systems conducted a hazards assessment to evaluate use of the microPHAZIR™ NIR with the PBX (AF)-108 warhead explosive. ATK LS Hazardous Operation Standards (AHOPS) require all portable electrical equipment within 25 feet of explosives or propellants to receive written approval from the Operating Building Electrical Classification Board (OBEC) committee chairman at ATK LS. Hazards analysis was performed for use of microPHAZIR™ NIR with PBX(AF)-108 and assessment of the platform was reviewed and approved by the OBEC, OSCB (Operations Safety Control Board) and PPCB (Plant Process Control Board) committees.

Thermo Fisher Scientific has also conducted numerous tests on various explosives and constituents of the same family, using microPHAZIR™ NIR with the following external group explosives test facilities and DoD depots, to identify RDX explosive:

I. Lew Kansas, Picatinny Arsenal, Dover, NJ.

II. Federal Law Enforcement Training Center.

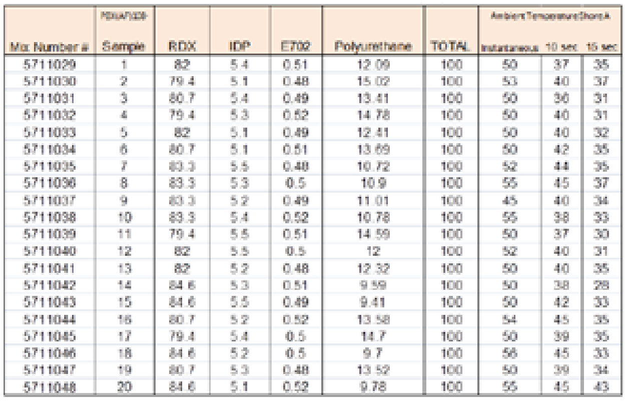
III. National Forensic Science Technology Center (NFSTC).

IV. Jimmie Oxley, University of Rhode Island.

V. A TF Homemade Explosives Training Course.

A D-Optimal Design of Experiment (DOE) was established, and consisted of manufacture of twenty mix iterations of PBX (AF)-108 with each constituent varied above and below specification limits, to capture high values and low values of each constituent. Twenty 600- gram (1-pint each) mixes of PBX (AF)-108 warhead explosive (Table 1) were prepared using a 1-pint Baker Perkins mixer. Each of the twenty 1-pint mixes was vacuum cast into a teflon-tape - lined cup-like carton. The Teflon tape facilitated carton removal and simulated a production-tooling surface. Each of the twenty cast mixes was cast to produce a rectangular-shaped block of 1-inch by 4-inch by 5-inch geometrical dimension, as depicted in Figure 1. Each of the twenty cast blocks of PBX (AF)-108 mixes was measured in triplicate, at the six sides, depicted in Figure 7, using microPHAZIR™ NIR handheld platform. Afterwards and to validate test results, samples measuring 0.5 to 1.0-gram were removed from the same locations of each rectangular cast block using a stirrup cutter and analyzed in triplicate by benchtop HPLC. The primary measured constituents were:

Table 1. Six-Hundred Gram mix Iterations of PBX(AF)-108



	PBX(AF)108						Ambient Temperature Shore A		
Mix Number #	Sample	RDX	IDP	E702	Polyurethane	TOTAL	Instantaneous	10 sec	15 sec
5711029	1	82	5.4	0.51	12.09	100	50	37	35
5711030	2	79.4	5.1	0.48	15.02	100	53	40	37
5711031	3	80.7	5.4	0.49	13.41	100	50	36	31
5711032	4	79.4	5.3	0.52	14.78	100	50	40	31
5711033	5	82	5.1	0.49	12.41	100	50	40	32
5711034	6	80.7	5.1	0.51	13.69	100	50	42	35
5711035	7	83.3	5.5	0.48	10.72	100	52	44	35
5711036	8	83.3	5.3	0.5	10.9	100	55	45	37
5711037	9	83.3	5.2	0.49	11.01	100	45	40	34
5711038	10	83.3	5.4	0.52	10.78	100	55	38	33
5711039	11	79.4	5.5	0.51	14.59	100	50	37	30
5711040	12	82	5.5	0.5	12	100	52	40	31
5711041	13	82	5.2	0.48	12.32	100	50	40	35
5711042	14	84.6	5.3	0.51	9.59	100	50	38	28
5711043	15	84.6	5.5	0.49	9.41	100	50	42	33
5711044	16	80.7	5.2	0.52	13.58	100	54	45	35
5711045	17	79.4	5.4	0.5	14.7	100	50	39	35
5711046	18	84.6	5.2	0.5	9.7	100	56	45	33
5711047	19	80.7	5.3	0.48	13.52	100	50	39	34
5711048	20	84.6	5.1	0.52	9.78	100	55	45	43

I. E-702 (4, 4’-methylenebis(2,6-di-tert-butyl-phenol) [MBDTBP]) anti-oxidant/stabilizer (Figure 2).

II. HMX/RDX oxidizer/fuel (Figure 3).

III. Isodecyl Pelargonate – IDP plasticizer (Figure 4).

IV. Binder content was determined by difference between the sum of the primary constituents/ingredients in (I), (II), and (III) and 100%].

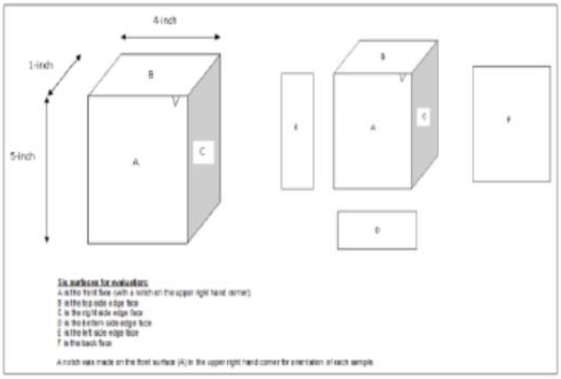

4-inch
1-inch
5-inch
B
A
C
E
B
A
C
F
D
Six surfaces for evaluation:
A is the front face (with a notch on the upper right hand corner)
B is the top side edge face
C is the right side edge face
D is the bottom side edge face
E is the left side edge face
F is the back face
A notch was made on the front surface (A) in the upper right hand corner for orientation of each sample.

Figure1. Final Dimensions of PBX (AF)-108 Cast Block

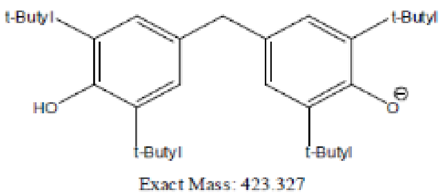


Figure 2. 4,4'-Methylenebis (2,6-Di-Tert-Butyl-Phenol) (MBDTBP)

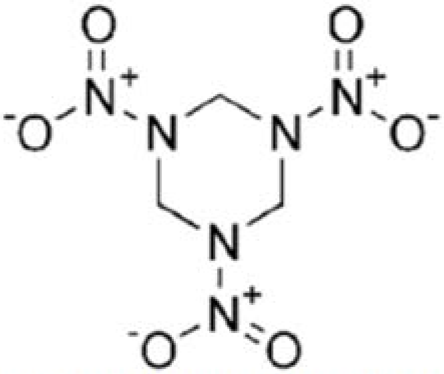


Figure 3. RDX (1,3,5-Trinitro-1,3,5-Triazacyclohexane)

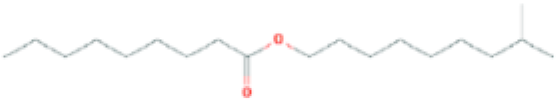


Figure 4. Isodecyl Pelargonate Plasticiser

2.2. Instrumentation

2.2.1. The microPHAZIR^{TM} (NIR) Platform

Near Infra-Red (NIR) spectroscopy is a well-established technique, which has been widely used since the mid-1970s. Only recently has new technology permitted NIR systems to be miniaturized into truly handheld system. One of the most important products is the microPHAZIR^{TM} NIR handheld platform. MicroPHAZIR^{TM} NIR handheld platform is based on near-infrared spectroscopy. The near-infrared region, depicted in Figure 5 is located between the infrared and visible region with wavelengths that range from 800-900 nanometers to 2500 nanometers.

MicroPHAZIR^{TM} NIR handheld platform was developed by Thermo Scientific and is based on vibrational spectroscopy. All molecules perpetually rotate, move, and contort in a complex manner at temperatures above absolute zero. Vibrational spectroscopy probes these contortions (or vibrations) of a sample to determine the chemical functional groups present. A common type of vibrational spectroscopy is infrared (IR) absorption/reflectance. It relies on illumination of the sample with optical radiation to probe the molecular vibrations.

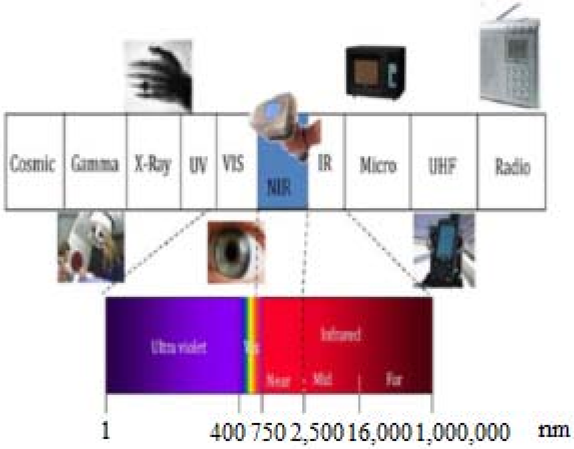


Figure 5. Near Infrared Region of the Light Spectrum

In NIR spectroscopy, the sample is illuminated with a broad spectrum of light in the near- infrared region and the transmission or reflection is recorded as a function of the frequency of the incident light. When the frequency of incident light equals the frequency of a specific molecular vibration, the sample tends to absorb some of the light. A material "fingerprint" results from recording the amount of light absorbed as a function of the wavelength (or frequency). The instrument is depicted in Figure 6. MicroPHAZIR^{TM} NIR is a rugged handheld chemical identification unit designed for point-of-use applications, either in contact or analysis can be conducted through transparent bags and vials. This product allows the identification of chemicals and white powders using the principles of NIR spectroscopy. It is enclosed in a lightweight, rugged, resistant package. The microPHAZIR^{TM} handheld contains a broadband NIR source, a Hadamard interferometer to separate the different wavelengths of light interacting with the sample, and a detector to collect the resulting energy.

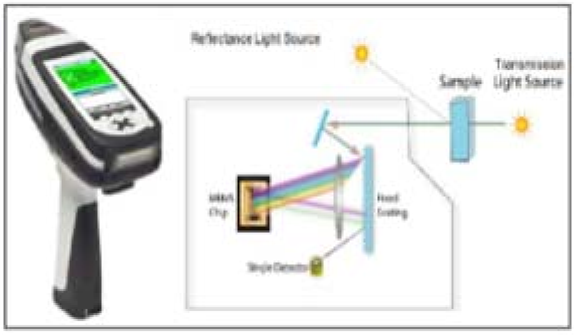


Figure 6. MicroPHAZIR^{TM} NIR and Principle of Operation

2.2.2. Agilent 1100 HPLC Platform

Agilent 1100 Series system with different configurations comprises a vacuum degasser, isocratic pump, high-pressure binary pump, low-pressure quaternary pump, autosampler, thermostatted column compartment, variable wavelength detector and diode array detector. Key measurements are necessary to evaluate the performance of HPLC systems. Some characteristics are influenced by only one part of the system. For example, linearity, spectral resolution and detection limits are influenced mainly by the detector, delay volume and composition accuracy by the pump and carryover by the autosampler. In contrast, other characteristics such as baseline noise and precision of retention times and peak areas are influenced by the complete system. This note describes the following measurements:

I. Detector — baseline noise, drift, wander, linearity, spectral resolution, sensitivity.

II. Pump — composition accuracy, precision, ripple, precision of retention times, delay volume.

III. Column compartment — temperature stability.

IV. Autosampler — precision of peak areas, linearity, carry-over.

2.3. Measurements

Measurements were performed on two platforms: microPHAZIR^{TM} NIR handheld platform and Agilent 1100 High-Performance Liquid Chromatography (HPLC) platform. In the case of microPHAZIR^{TM} NIR handheld platform, measurements were performed and recorded on each of the six faces of each of the twenty blocks (representing twenty mixes), as depicted in Figure 7.

Upon manufacture and vacuum cast of the explosive mix, the energetic surface is binder rich, henceforth prior to performing measurements approximately 1/8” of the binder-rich surface of each cast block is peeled-off and removed from the surface, exposing the homogeneous material. As illustrated in Figure 8.

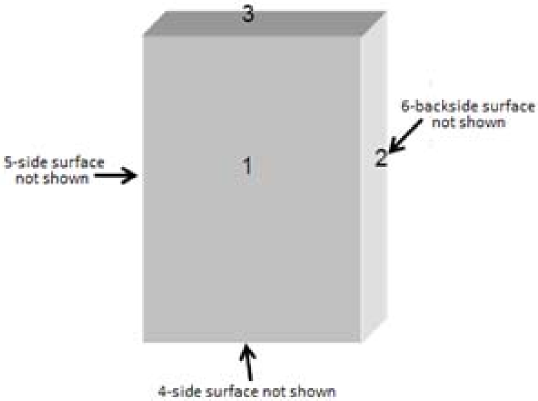


Figure 7. Cast PBX(AF)-108 Blocks Depicting Measured Surfaces

Figure 8. Peeled-Off Binder-Rich Layer

All measurements were recorded in the middle region of the surface of each block, as depicted in Figure 9 (right), in an area that was flat and free of any discoloration. Three independent NIR spectra were collected at each location (six total locations) per 600-gram of PBX(AF)-108 sample block, for a total of 18 readings per block.

In the case of benchtop Agilent 1100 High-Performance Liquid Chromatography (HPLC) platform, samples were prepared for analysis by extracting in triplicate with acetonitrile at a level of 10.0 mg/ml. Extracts were analyzed using an Agilent 1100 High-performance liquid chromatography (HPLC) instrument with an Octadecyl Silane (ODS) column and a diode-array detector. Samples were extracted from each of the six locations of each of the twenty mixes to evaluate the levels of E-702 anti-oxidant (4,4-methylenebis (2,6-di-tert-butyl-phenol)), Royal Demolition explosive (RDX) oxidizer (1,3,5- Trinitro-1,3,5-triazacyclohexane), and Isodecyl Pelargonate (IDP) plasticizer. The pint mixes were formulated with RDX that contained a small percentage of High Melting explosive (HMX). Thus the sum of HMX and RDX percentage by HPLC analysis represented the amount of RDX formulated in PBX (AF)-108 pint mixes.

Figure 9. Measurements Recorded in the Middle and Surrounding Regions of each Face of the Sample Block

2.4. Model Building

2.4.1. Data Collection

Collecting PBX (AF)-108 data using microPHAZIR^{TM} NIR handheld platform followed best practices recommendations of the platform manufacturer, as follows:

I. Obtain representative samples for the library.

A. Obtain realistic sample mixes that will form the library. These sample mixes should be representative of the PBX (AF)-108 material that will be identified. No selectivity is implied for materials until the library is built and validated.

B. Measure samples, as illustrated in Figures 5A and 5B. Perform measurements in triplicate.

C. Label all materials with name (Group ID or Method/Sample), and if appropriate reference value for PLS quantitative analysis.

D. Transfer all names into a ".csv" file, and then use this to populate "GroupID.csv" on the microPHAZIR^{TM} "Config" directory.

II. Obtain reference values.

A. For quantitative analysis, the full range of measurement shall be included in the library. Models only are considered robust over the data range actually referenced.

B. Obtain replicate samples for at least 3 points over the measurement range.

C. For realistic model building, at least 10 reference values over the measurement range shall be obtained. As the size of the range increases, so should the reference values collected. iv. Since samples may change over time, it is appropriate to collect the spectra from the same sample as the reference values are obtained from.

2.4.2. Spectra Generation

I. Pre-spectral collection.

A. Prior to collecting spectra ensure that self-test performance qualification (PQ) has been performed.

B. Ensure that group identifications (groupids) are transferred into GroupID.csv.

C. Also ensure that the GroupID name is the correct name for the material and is present on the Collect screen on the microPHAZIR^{TM}.

II. Spectral collection'

A. The minimum number of spectra collected for any library building is triplicate scans in 3 positions. Position the nose of microPHAZIR^{TM} firmly against the material to be measured, as depicted in Figure 9 (right), and take triplicate scans of the material without moving the sample. This will give information about instrument variability. Repeat twice.

B. Repeat measurements for each side of the block.

C. Repeat steps (A) and (B) for each mix.

2.4.3. Spectral Evaluation

I. Initial spectral evaluation.

A. Load the collected data into Method Generator

B. Ensure that there are no data which show absorbance (y-axis) past 3.

C. Observe if there are any noisy spectra, especially at high absorbance. If so, delete them. These usually arise if the trigger was pressed either without a sample in front or if sample is inadequate.

D. Highlight each group to make sure that all spectra look similar in the same group. Any obvious single outliers may be deleted. The best scenario is when the triplicate scans are right on top of each other, and there is little difference between positional scans. However, as long as the positional replicates appear similar and are close together, this is adequate. If one position is obviously off from the others, keep it, but watch to see if it affects the final results.

E. Delete any spectra where there was awareness of probable mistake in measurement. Do not delete scans just to make everything pretty. Deviations from the norm could be due to actual inherent sample differences and will need to become part of the model.

F. Reference values must be inputted at this time, using the Edit Y-value option.

G. Save the final edited data.

II. Method generation

A. Progress through the standard preprocessing options, and then evaluate the model using Spectral Match.

B. Adequate separation should be observed between samples. There should be a gap between the colors associated with one group and the next closest color of the nearest group.

C. Save the model if the model is acceptable.

D. Load the data files onto the microPHAZIR^{TM} to test the model.

III. Method validation

A. Load a set of spectra into method generator (MG). For true method validation these should be unique spectra, not used in library building.

B. Select Model |Model validation. Browse to locate the application. Press OK

C. A panel will open with the validation results. It will be sorted by sample groups. Therefore it is very important that the GroupID of new spectra be identical to the GroupID of the library spectra. Otherwise a No ID label will be inserted.

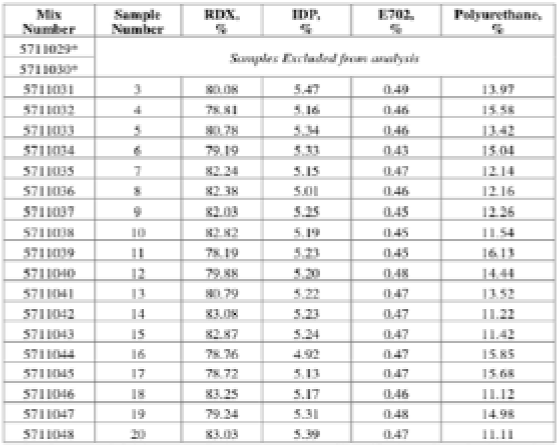
D. The results show number of mismatches, false positives/false negatives, and then the full results of the model validation for each material. It will list the top 3 matches returned and their associated correlation coefficients.

E. The results can be saved as a ".csv" file by selecting File | Save all

3. RESULTS AND DISCUSSION

Test sets collected from all twenty samples were analyzed using microPHAZIR^{TM} NIR handheld platform. Upon reduction and analysis of the data, initial findings indicated near identical readings between those measured using microPHAZIR^{TM} NIR handheld platform and those measured using benchtop HPLC, as listed in Tables 2 and 3 and compared to the initial added quantities of each constituent in Table 1. This finding concluded platform capability to measure oxidizer (RDX), stabilizer (Methylene di-tertiary butyl phenol), and plasticiser (IDP) contents on real-time basis with excellent precision.

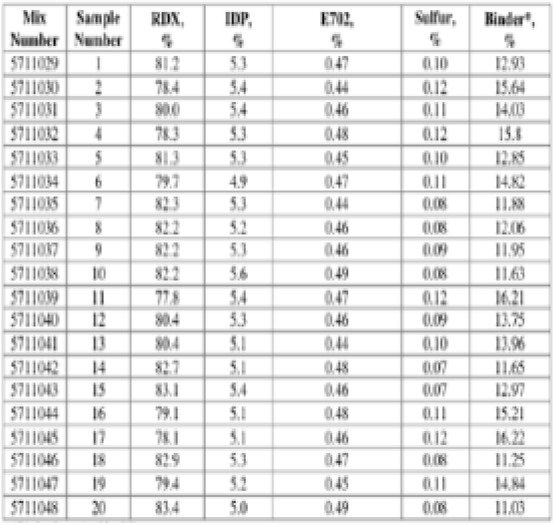
Table 2. MicroPHAZIR^{TM}NIR Results of 20 DOE Mixes



Mix Number	Sample Number	RDX, %	IDP, %	E702, %	Polyurethane, %
5711029*	Samples Excluded from analysis				
5711030*					
5711031	3	80.08	5.47	0.49	13.97
5711032	4	78.81	5.16	0.46	15.58
5711033	5	80.78	5.34	0.46	13.42
5711034	6	79.19	5.33	0.43	15.04
5711035	7	82.24	5.15	0.47	12.14
5711036	8	82.38	5.01	0.46	12.16
5711037	9	82.03	5.25	0.45	12.26
5711038	10	82.82	5.19	0.45	11.54
5711039	11	78.19	5.23	0.45	16.13
5711040	12	79.88	5.20	0.48	14.44
5711041	13	80.79	5.22	0.47	13.52
5711042	14	83.08	5.23	0.47	11.22
5711043	15	82.87	5.24	0.47	11.42
5711044	16	78.76	4.92	0.47	15.85
5711045	17	78.72	5.13	0.47	15.68
5711046	18	83.25	5.17	0.46	11.12
5711047	19	79.24	5.31	0.48	14.98
5711048	20	83.03	5.39	0.47	11.11

*Samples excluded due to anomalies in measurements.

Table 3. Average Readings of Benchtop HPLC Results**



Mix Number	Sample Number	RDX, %	IDP, %	E702, %	Sulfur, %	Binder*, %
5711029	1	81.2	5.3	0.47	0.10	12.93
5711030	2	78.4	5.4	0.44	0.12	15.64
5711031	3	80.0	5.4	0.46	0.11	14.03
5711032	4	78.3	5.3	0.48	0.12	15.8
5711033	5	81.3	5.3	0.45	0.10	12.85
5711034	6	79.7	4.9	0.47	0.11	14.82
5711035	7	82.3	5.3	0.44	0.08	11.88
5711036	8	82.2	5.2	0.46	0.08	12.06
5711037	9	82.2	5.3	0.46	0.09	11.95
5711038	10	82.2	5.6	0.49	0.08	11.63
5711039	11	77.8	5.4	0.47	0.12	16.21
5711040	12	80.4	5.3	0.46	0.09	13.75
5711041	13	80.4	5.1	0.44	0.10	13.96
5711042	14	82.7	5.1	0.48	0.07	11.65
5711043	15	83.1	5.4	0.46	0.07	12.97
5711044	16	79.1	5.1	0.48	0.11	15.21
5711045	17	78.1	5.1	0.46	0.12	16.22
5711046	18	82.9	5.3	0.47	0.08	11.25
5711047	19	79.4	5.2	0.45	0.11	14.84
5711048	20	83.4	5.0	0.49	0.08	11.03

*Binder determined by difference.

**For each mix, recorded value is average of 3 readings.

As part of the developed algorithm, binder content would also show a reading, which is calculated based on the difference between the total measured primary constituents and 100 percent. Data produced on the microPHAZIR^{TM} NIR handheld platform were plotted for the primary constituents of PBX (AF)-108. Data are depicted in Figure 10 for IDP plasticizer, Figure 11 for E-702 stabilizer/anti-oxidant, and in Figure 12 for RDX oxidizer, respectively.

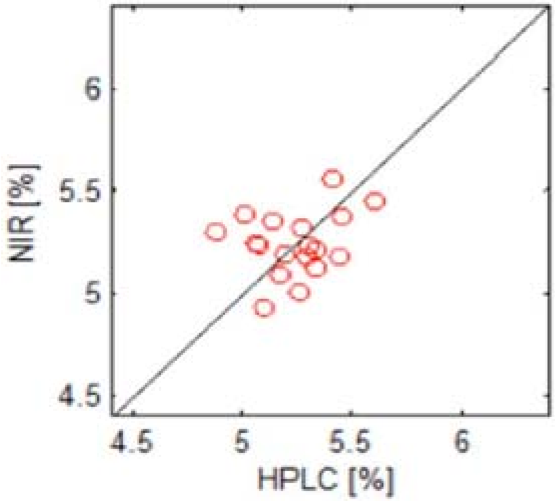


Figure 10. NIR vs. HPLC Concentration for IDP

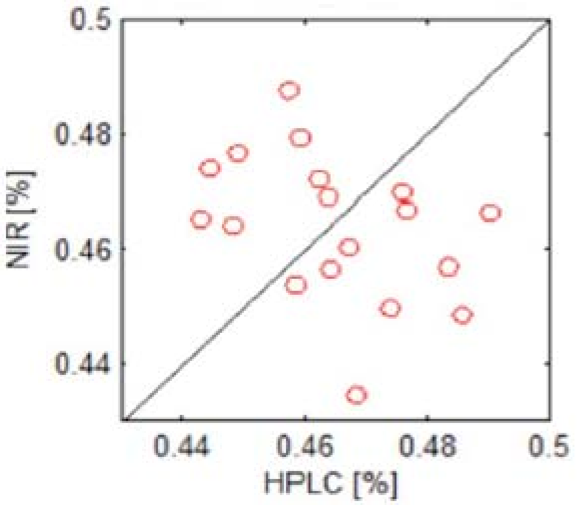


Figure 11. NIR vs. HPLC Concentration for E-702

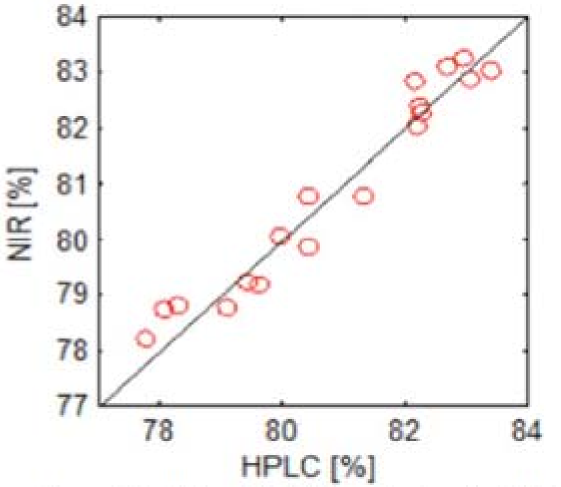


Figure 12. NIR vs. HPLC Concentration for RDX

Results indicate very good agreement for RDX, good agreement for IDP and acceptable agreement for E-702, between both sets of measurements in both handheld platform and benchtop platform, which further confirmed efficacy of using microPHAZIR^{TM} NIR handheld platform as a tool for real-time non-destructive measurements of primary chemical constituents of PBX (AF)-108.

As noted earlier, microPHAZIR^{TM} NIR handheld platform did not directly measure binder content. The algorithm was developed to calculate a value for the binder, which is based on the difference between the sum of the primary ingredients and 100%. Results which were measured using microPHAZIR^{TM} NIR handheld platform are summarized in Table 2 in units of percent by weight.

Results from the HPLC analysis are summarized in Table 3. All measurements indicated close readings between the actual formula of the twenty explosive mixes and HPLC readings, with the exception of readings which were measured for percent Isodecyl Pelargonate (IDP) plasticizer. This is a common trend often observed with plasticizer-containing composite energetics (explosives, boosters, and solid rocket propellants), and was determined to be a result of diffusion (migration) and eventually loss of the plasticizer.

A common limitation with plasticized PBXs is that of plasticizer migration over time and at elevated temperatures during their service and even in the early stages of manufacture, during cure of the explosive charge. Plasticization is one of two mechanisms: internal plasticization, in which plasticizer molecules are attached to the polymer covalently as part of the polymer chain. And external, in which DOA is homogenized and absorbed by the polymer and oxidizer ingredients (as in PBXN-11) and other polymer-based explosives or rocket propellants. In internal plasticization, migration of plasticizers is absent. Plasticizer migration is more a concern when external plasticization occurs.

In almost all plasticizers, loss occurs primarily due to their volatility, which comprises dual transport phenomena. The first is diffusion, which occurs when DOA moves through the explosive matrix from the inner core to the surface, and that is largely influenced by temperature coupled with viscosity and molecular weight of the plasticizer. The second is evaporation, which is greatly influenced by temperature and vapour pressure of the plasticizer.

In the case of PBX (AF)-108 mixes, which were vacuum cast and subjected to high temperature cure, plasticizer migration was present, and could be noticed in coefficient of variation readings, depicted in Figure 13. For each mix, six faces (locations) of the rectangular block (Figure 4) were measured in triplicate, using HPLC bench top instrument. Coefficient of variation was stable (under or around 5%) with the exception of mixes 1, 16, 17, 18, 19, and 20. This variation is more an indication of IDP migration (diffusion) throughout the explosive matrix.

Since coefficient of variation (CV) is a statistical tool generally used to express the standard deviation as a percentage of what is being measured relative to the sample or population mean, this variation is expressed as a percent of the mean. CV is determined by the relationship of “equation (1)”:

$$CV = (SD/\ \bar{X}) \times 100 \qquad (1)$$

Where $\bar{X}$ and SD represent the sample mean and the sample standard deviation, respectively.

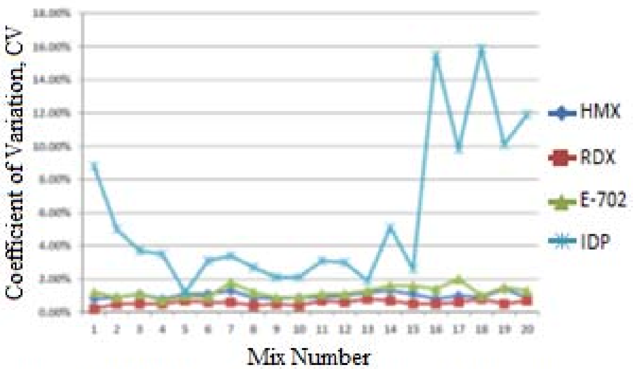


Figure 13. HPLC Coefficient of Variation vs. Mix Number

CV is a reliable reading when standard deviation (SD) rises proportionally with concentration. For example, in a replicate experiment data could indicate a value of 4 for standard deviation at a concentration of 100 units. This value could indicate a reading of 8 for standard deviation but this time at a concentration of 200 units. In both cases, coefficient of variation is the same, i.e. 4.0%. Coefficient of variation in such case is more useful than standard deviation for describing method performance at concentrations in between. In some tests however, standard deviation may be constant over the analytical range. Generally speaking, coefficient of variation CV provides a general "feeling" about the performance of a method. CVs of 5% or less generally give us a feeling of good method performance, whereas CVs of 10% and higher sound bad. However, the mean value becomes important before judging a CV. At very low concentrations, the CV may be high and at high concentrations the CV may be low. For example, a test with an SD of 0.1 mg/dL at a mean value of 0.5 mg/dL has a CV of 20%, whereas an SD of 1.0 mg/dL at a concentration of 20 mg/dL corresponds to a CV of 5.0%.

In Figure 14 the average standard deviation is plotted against the mix number. In the graph, standard deviation for RDX,/HMX, and E-702 is markedly low, and within specifications of +/- 1.7% and 0.01%, respectively (RDX/HMX = 82.0 +/- 1.7%; E-702 = 0.50 +/- 0.01%; IDP = 5.3 +/- 0.10%). In the case of RDX/HMX maximum measured standard deviation value is 0.8%, significantly below the maximum allowable specification and well within acceptable range. Similarly, standard deviation for E-702 is just 0.01 and within its specification limit of +/- 0.01%. In the case of the plasticiser (IDP), standard deviation is relatively higher than would be expected for most composite explosives and propellants with high solids content. This is attributed to (i) migration of the plasticiser within the explosive matrix until steady state is attained (when the explosive is fully annealed); (ii) as depicted in Figure 8, approximately 1/8” of the binder-rich surface of each cast block was peeled-off and removed from the surface of each sample block, henceforth exposing the homogeneous surface of the explosive. In a freshly manufactured mix, this binder-rich layer often contains higher amounts of plasticiser than fully annealed mixes. Because this process was performed by hand, rather than with a precision tool, it would be reasonable to expect higher variability in liquid plasticiser content between mixes, and therefore higher standard deviation. Lessons learned will be instituted in future work by (i) allowing mixes to sit for a period of 6-8 weeks, to allow for the plasticiser component to attain steady state within the matrix; and (ii) a high precision tool will be used in removing the binder-rich layer, so that consistency between all mixes is exercised.

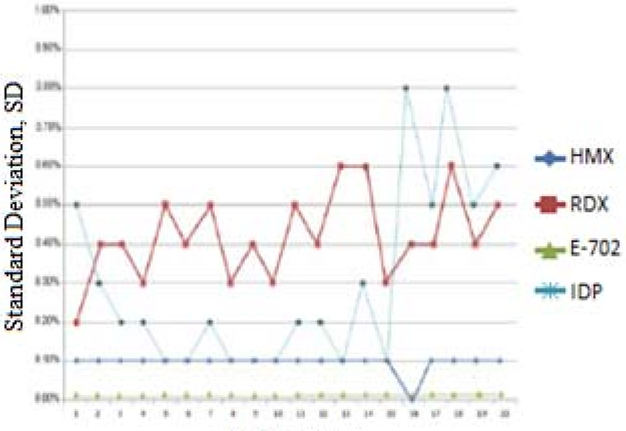


Figure 14. HPLC Standard Deviation vs. Mix Number

In the case of microPHAZIR^{TM}, standard deviation values are higher than HPLC values. This is due to the fact that NIR penetrates as high as 5 mm under the surface, whereas in the case of HPLC, a much larger sample is extracted from several locations and deeper than 5 mm. In both techniques (HPLC vs. microPHAZIR^{TM}) and as summarized in Tables 2 and 3, measured values of primary constituents of each the twenty mixes, correlated quite well.

Generally speaking, it would be reasonable to expect plasticizers in composite polymeric mixes to take longer to reach steady state levels throughout the matrix. These levels however tend to vary from one plasticizer to another, depending on molecular weight, vapour pressure, and viscosity of a plasticizer.

4. CONCLUSION

The D-optimal design of experiment (DOE) was successful in developing an algorithm for microPHAZIR^{TM} NIR handheld platform for use in quantitative determination of primary chemical constituents of PBX (AF)-108. Therefore, use of microPHAZIR^{TM} NIR handheld platform for real-time non-destructive quantification of constituents of plastic bonded explosives was determined to be a valid test method without generating chemical waste and/or residue. Datasets from both microPHAZIR^{TM} NIR handheld platform and Agilent 1100 High-pressure liquid chromatography (HPLC) platform were close to each other and representative of the constituents of PBX (AF)-108 explosive. In the case of the microPHAZIR^{TM} NIR handheld platform, the dataset indicates more stability in variance across the full datasets however observations of measurements collected using Agilent 1100 High-pressure liquid chromatography (HPLC) platform indicates sample inhomogeneity as a result of plasticizer migration. In the case of microPHAZIR^{TM} NIR handheld platform the following observations may be stated:

I. Averaging data improves results.

II. Excellent ability to quantify HMX+RDX concentration 3. Good ability to quantify IDP plasticizer concentration and migration trends

III. Migration trends of IDP plasticizer requires multiple measurement points to improve reliability and confidence levels

IV. Good ability to quantify E-702 stabilizer concentration.

V. Repeat measurements of E-702 stabilizer content will improve reliability and confidence levels

Some drawback associated with microPHAZIR^{TM} NIR handheld platform may be summarized as follows:

I. The platform has not yet been validated to offer reliable measurement of components, which exist in trace concentrations (below 0.1%). More testing will be deemed necessary in future DOEs.

II. The platform periodically undergoes automatic calibration, during measurements, which is a necessary step, but can be frustrating at times.

III. For measuring chemical constituents of other explosives, a new DOE will be required for each explosive formula and an associated algorithm must be developed.

IV. Use of microPHAZIR^{TM} NIR handheld platform for measuring chemical constituents of rocket motor solid rocket propellants will require retrofitting of an optical fiber wand to the machine. The wand will enable the user to measure chemical constituents inside the solid rocket propellant bore.

V. Because microPHAZIR^{TM} NIR handheld platform is a handheld instrument, it requires the use of a rechargeable lithium ion battery, and therefore a spare battery and a battery charger are important components of the platform and are supplied by the manufacturer, Thermo Fisher Scientific.

NOMENCLATURE

AF	Air Force
AOP	Allied Ordnance Publication
ATK	Alliant Techsystems
LS	Launch Systems
BTCA	Breakdown, Test and Criticality Analysis
CAD	Cartridge-Actuated Device
CT	Computed Tomography
CV	Coefficient of Variance
DOA	Dioctyl Adipate
DoD	Department of Defense
DSC	Differential Scanning Calorimetry
DSTO	Defence Science and Technology Organization
E-702	4,4-methylenebis (2,6-di-tert-butyl-phenol)
EED	Electro-Explosive Device
ESD	Electrostatic Discharge
HMX	High Melting eXplosive
HOPS	Hazardous Operation Standards
HPLC	High-Performance Liquid Chromatography
IDP	Isodecyl Pelargonate
MoD	Ministry of Defence
NATO	North Atlantic Treaty Organization
NDE	Non-Destructive Evaluation
NDT	Non-Destructive Testing
NIR	Near-Infrared
OBEC	Operating Building Electrical Classification Board
ODS	Octadecyl Silane
OSCB	Operations Safety Control Board
PAD	Propellant-Actuated Device
PBX	Plastic-Bonded Explosive
PHM	Prognostics and Health Management

REFERENCES

C. Gardener, and M. Hargreaves, 2012. “Near Infrared Data Report for PBX(AF)-108 Warhead Explosive”. Thermo Fisher Scientific, Tewksbury, MA 01887, USA.

S. Schreyer, 2012, Thermo Scientific Training Course Tutorial Series: “Building Quantitative (PLS-1) Models”. Thermo Scientific, Tewksbury, MA 01887, USA.

S. Schreyer, 2012, “Thermo Scientific Best Practices for Collecting and Evaluating Spectra from *microPHAZIR*^{TM} NIR handheld platform”. Thermo Scientific, Tewksbury, MA 01887, USA.

K. Spalding, 2012. ANALY-CY11-0102 Rev A report: “HPLC analysis of PBX (AF)-108 explosive”. ATK Launch Systems, Aerospace Systems, Brigham City, UT 84302, USA.

G. Bocksteiner, and D.J. Whelan, *November 1995*, DSTO-TR-0228: “The Effect of Ageing on PBXW-115(Aust.) PBXN-103 and PBXN-105”. *Department of Defence, Defence Science* and Technology Organization (DSTO).

Mattos et al., 2004, “Determination of the HMX and RDX Content in Synthesized Energetic Material by HPLC, FT-MIR, and FT-NIR Spectroscopies”, Química Nova, Vol. 27, No. 4, pp. 540-544.

Urbanski et al, 1977, “Handbook of Analysis of measures of Synthetic Polymers and Plastics”, John Wiley & Sons, New York, 494 p.

M. Blanco, and I. Villarroya (2002) NIR spectroscopy: “A rapid-response analytical tool”. Trends in analytical chemistry 21:240-250.

H.W. Siesler, Y. Ozaki, S. Kawata, and M. Heise (Eds) (2002). Near Infrared Spectroscopy Principles, Instruments, Wiley-VCH.

MicroPHAZIR^{TM} User Manual, Thermo Scientific Handheld Near-Infrared Analyzer. Thermo Fisher Scientific, Tewksbury, MA 01887, USA.

Performance Characteristics of the Agilent 1100 Series Modules and Systems for HPLC. Agilent Technologies, Publication Number 5965-1352E.

Battery Charge Depletion Prediction on an Electric Aircraft

Quach Cuong Chi[1], Brian Bole[2], Edward Hogge[3], Sixto Vazquez[1], Matthew Daigle[4], José Celaya[5], Adam Weber[1], Kai Goebel[4]

[1] *NASA Langley Research Center, Hampton, VA 23681*
cquach@nasa.gov, sixto.l.vazquez@nasa.gov, adam.k.weber@nasa.gov

[2] *Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332*
bbole3@gatech.edu

[3] *Northrop Grumman Technical Services , Hampton, VA 23681*
edward.f.hogge@nasa.gov

[4] *NASA Ames Research Center, Moffett Field, CA 94035*
matthew.j.daigle@nasa.gov, kai.goebel@nasa.gov

[5] *SGT, Inc., NASA Ames Research Center, Moffett Field, CA 94035*
jose.r.celaya@nasa.gov

Abstract

Validation of prognostic technologies through ground and flight tests is an important step in maturing these novel technologies and deploying them on real-world systems. To this end, a series of flight tests have been conducted using an unmanned electric vehicle during which the motor system batteries were monitored by a prognostic algorithm. The research presented here endeavors to produce and validate a technology for predicting the remaining time until end-of-discharge of the batteries on an electric aircraft as a function of an expected future flight and online estimates of the charge contained in the batteries. Flight data and flight experiment results are presented along with an assessment of model and algorithm performance.

1. Introduction

Recent improvements in battery technology have increased energy density and capacity to the point of considering them for general aviation vehicles. Battery health management (BHM) is a safety-critical enabling technology for electric aviation. Safe adoption of battery-powered propulsion in aviation, however, suffers from difficulty in accurately estimating the total storage capacity in the batteries and determining the remaining useful capacity at any given instant during flight. This paper describes the application and assessment of a battery health prognostics system in an unmanned all-electric subscale aerial vehicle.

We develop a BHM system using a model-based framework on the Edge-540T electric aircraft (Saha et al., 2011). In model-based prognostics (Saha & Goebel, 2009; M. Orchard, Tobar, & Vachtsevanos, 2009; Daigle & Goebel, 2013; Luo, Pattipati, Qiao, & Chigusa, 2008), a model of the system under prognosis is developed for the purposes of state estimation and remaining life prediction. In this work, we compare the original particle filter-based implementation with a new unscented Kalman filter-based implementation that takes advantage of an improved battery model and new input prediction methods in order to improve health state estimation and end-of-discharge prediction performance.

This paper is organized as follows. Background information and motivation for the implementation of onboard battery health management algorithms for electric vehicles are given in Section 2. The prototype electric aircraft used to demonstrate battery charge estimation and battery end of discharge (EOD) prediction is described in Section 3. Battery state of charge (SOC) estimation and EOD prediction results are presented in Section 4 along with a description of relevant model-based filtering techniques. Uncertain EOD predictions made over a sample flight of the unmanned aerial vehicle (UAV) are presented and assessed in Section 5. Conclusions and future work are discussed in Section 6.

Quach Cuong Chi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. Background

Electric propulsion has long been used in cars and small UAVs. Recently, improvements in battery storage capacity have made it possible for general aviation vehicle manufacturers to consider battery-powered solutions as well. With recent urgency to address environmental concerns, vehicle manufacturers are increasingly investing in electric alternatives (Harrup & Davis, 2010). Further, electric propulsion provides a number of operational and control advantages: reduced noise, no emissions, more responsive thrust, and reduced part count, to name a few. Use of battery-powered propulsion systems in manned and unmanned aircraft alike will require more sophisticated means of estimating available battery charge during operation. The estimation of remaining charge is not easily determined during usage and, in addition, each battery's charge storage capacity degrades over its life span.

As a result, operators of electrically-propelled aircraft are left making conservative estimates of mission time. And, if a significant change to the mission is required during flight, no simple method exists to determine actual versus required battery charge. In other words, an operator currently must have a reliable way to answer the following questions: 1) What is the required charge to complete the new mission?, and 2) What is the actual charge left in the batteries?

2.1. Battery Health Management Approaches

The main objective of BHM research is to create prognostic algorithms that provide accurate estimates of battery storage capacity during flight planning and accurate indication of remaining charge during flight. Battery system models for electric aircraft have been developed based on previous laboratory and field experiments by (Saha, Goebel, Poll, & Christophersen, 2009; Saha et al., 2011) and by (Daigle, Saxena, & Goebel, 2012). The work reported here covers testing and adapting the battery prognostic model to the flight environment of a real electric vehicle.

The applied approach is to develop and implement onboard BHM which monitors usage of the motor batteries and which runs estimation and prediction algorithms to: 1) determine the SOC, which expresses the remaining battery charge in a relative percentage; 2) predict the EOD, which is the total flying time; and 3) estimate the Remaining Useful Life (RUL), which is the remaining flying time from the present instant. The SOC is intended to be much like the fuel gauge in a conventional liquid fueled system. The RUL and EOD both describe similar information, which is to provide the operator some notion of how much operating time is remaining. The difference is that the EOD predicts the total flying time relative to the start of the flight, whereas the RUL predicts the remaining flying time relative to the current time. Because the charge storage capacity and other battery parameters are known to degrade over the battery's life, the underlying battery model must be tuned such that EOD predictions account for the life cycle degradation of the batteries.

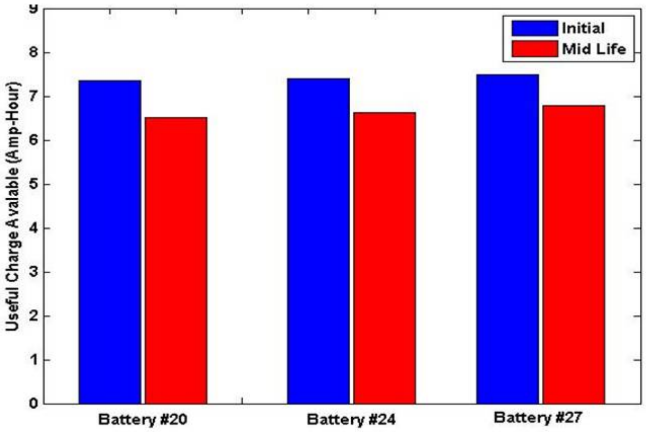


Figure 1. Useful charge available from new vs. old battery

2.2. Challenges in Online Computation of Battery SOC

In conventional liquid fuel systems, the remaining fuel level can be reliably measured and thus the remaining operating time can be obtained using vehicle and motor performance characteristics. This is because the volume of the tank is constant over the vehicle's lifespan. To the contrary, charge capacity in batteries can diminish over recharge cycles, and, depending on the chemistry used, over time as well. Part of the challenge of predicting the current battery SOC is determining the maximum capacity of constituent cells, which represents the initial condition of the discharge curve. As batteries age and experience an increasing number of recharge cycles, their maximum capacities diminish. Figure 1 illustrates this typical life cycle degradation for three batteries used in several flight experiments.

For this research, the charge capacity for a battery is the charge it can supply between its maximum rated voltage and the point when voltage drops precipitously under load. The precipitous drop is figuratively called the ”knee point”.

3. Prototype Electric Vehicle Description

A 33% scaled Edge-540T, with electric propulsion, is used for this BHM research and development, as shown in Figure 2. It is 98 inches long with a 100 inch wing span and weighs 47.4 lbs., has 1881 in^2 of wing area with an average wing load of 0.025 psi.

The power system consists of two outrunner brushless DC electric motors mounted in tandem to drive a 26-inch propeller. The motor assembly turns the propeller up to 6000 RPM to develop about 37 pounds of thrust. Its airspeed ranges from a stall of 12 m/s to a dash of about 40 m/s (23-77

Figure 2. Picture of Edge-540T during landing

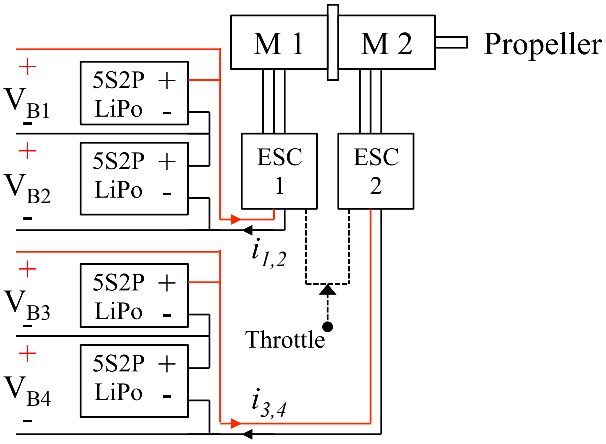


Figure 3. Motor System Diagram

knots).

Electrical connections in the aircraft powertrain are illustrated in Figure 3. The two propeller motors are each powered by a series connection of two lithium-polymer (LiPo) battery packs. The power flow to the motors is controlled by a Jeti 90 Pro Opto electric speed controller (ESC). The ESC sends synchronized voltages to the motors at a duty cycle that is determined by a throttle input signal sent either by remote control from a pilot, or by an onboard autopilot.

Inductive loop current sensors are mounted on the positive lead feeding each ESC. Additional current sensors are also mounted on the positive feed from each of the four batteries. The positive lead of each battery is tapped to provide the data system with battery voltage measurements. These are the signals that the BHM system uses to estimate SOC, EOD, and RUL.

3.1. Energy Storage System

Each battery used to power the tandem motors consists of two sets of five series-connected 4.2 V 3900 mAh LiPo pouch cells, wired in parallel (Figure 4). The total rated capacity of each pack is 7800 mAh with a 50 C max burst discharge. The

Figure 4. Powertrain battery packs

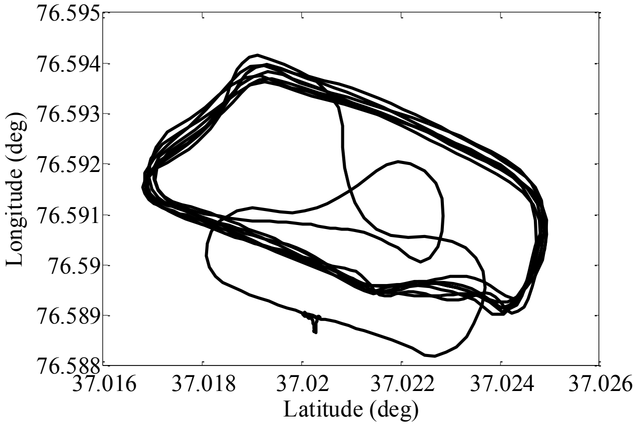


Figure 5. Ground track of flight

50 C burst discharge allows a takeoff draw of up to 390 A. Takeoffs generally peak about 140 A. When fully charged, each 5-cell pack has an open circuit voltage of 21 V (4.2 V per cell). Two serially connected packs provide a maximum potential of 42 V to each motor.

The airplane is equipped with an autopilot that has the capability to navigate using a stored flight plan. A safety pilot typically flies the airplane from runway to test altitude and switches control to the autopilot, which proceeds to fly the flight plan. Figure 5 shows the ground track for a typical flight. Flight activities typically occur at 200 m above ground level. Flights last about 15 minutes with flight duration depending largely on throttle management.

A ground station interface monitors navigation and flight status using a vertical situation display, a moving map, and various text parameter displays. A second ground display monitors the powertrain batteries showing voltages, currents, remaining flight time, and state of charge for all four powertrain batteries.

3.2. Data System and Raw Data

The aircraft is instrumented with a real-time data system, described in (Hogge, Quach, & Hill, 2011). For the BHM research, the data system records battery voltage, current, and

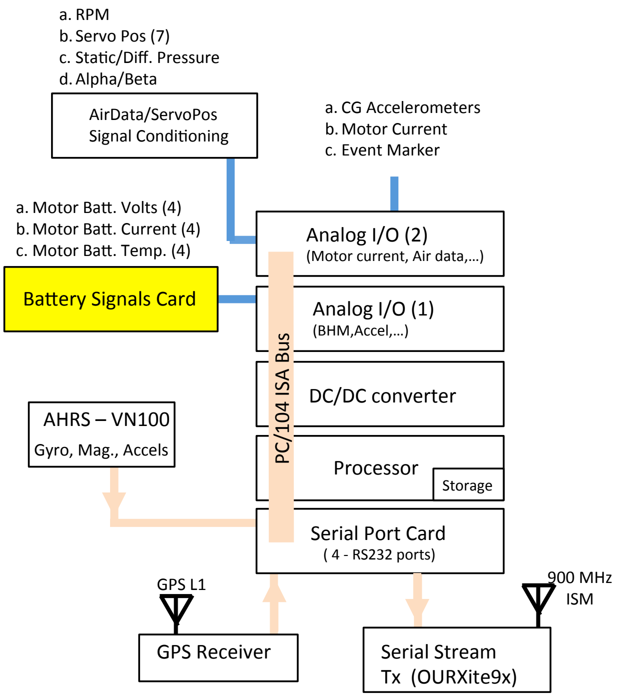


Figure 6. DAQ hardware diagram

temperature at 400 Hz. Figure 6 diagrams the data system hardware components. The data system also executes a battery prognostic model that computes the EOD, RUL, and SOC for all four batteries.

3.3. Flight Plan (Mission)

A sample flight of the Edge-540T electric aircraft is described in this section. This flight test includes: 1) a flight segment from take-off at 0 s to landing at 692 s; 2) taxiing to position from 692 s to 885 s; 3) operating the motor on the runway to discharge the batteries (885-1435 s). The experiment is stopped when the battery voltages drop below a specified cut-off threshold of 17 volts that designates the end of useful power delivery from the batteries.

The first flight segment (0-692 s) consists of essentially four activities. These include a take-off, followed by flight in auto-mode executing a flight plan at constant throttle set to 75%. The throttle is increased to 85% from 322 to 550 s. The throttle is then decreased to 75% from 550 to 692 s.

Figure 7 shows the throttle, propeller RPM, and battery power profiles recorded over a sample flight. Recall from Figure 3 that each motor is fed by a different circuit and that each circuit is powered by two batteries wired in series. The blue/square and red/diamond traces shown in Figure 7 give the net battery power input to the ESCs powering motors M1 and M2, respectively. The net battery power input to an ESC is given by the product of the current flowing through the ESC and the sum of the two series-connected battery voltages powering the ESC. The black trace in the battery power plot gives the sum of all battery power input to the two onboard ESCs.

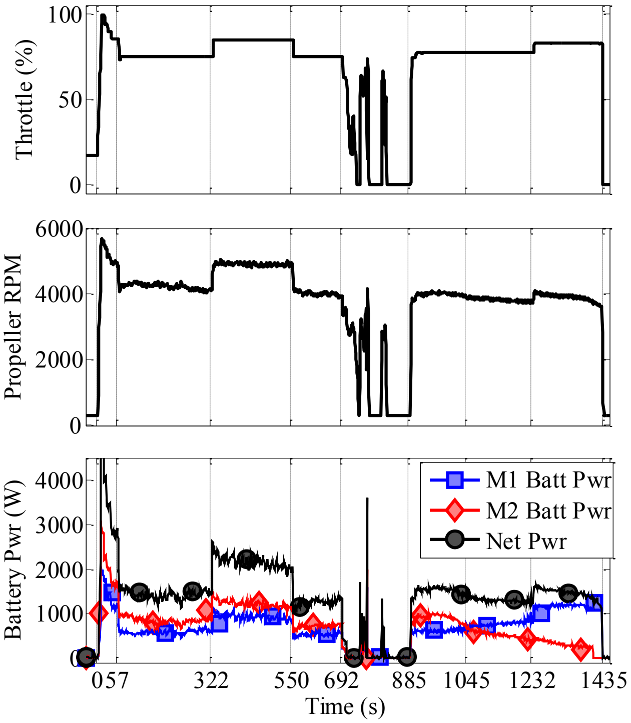


Figure 7. Plots of throttle command, propeller RPM, and battery power draw measured over a sample flight

Figure 8 shows the current and voltage profiles for three of the four batteries used to power the two propeller motors onboard the Edge-540T. The current and voltage data for battery B4 showed anomalous readings and are omitted from the plots in this paper. The ticks on the x-axis in Figures 7 and 8 denote the time of notable activities.

An interesting observation to note regarding the battery power draw over the sample flight is that motor M2 draws more power than motor M1 the majority of the time. This occurs until the batteries supplying power to motor M2 are depleted to about 18 volts each. At this point, the batteries powering motor M1 begin to take the majority load. This crossover is seen in both the battery power and battery current profiles at about the midpoint between 885 and 1232 s.

Relatively constant current draw is seen in Figure 8 for the three periods of constant throttle. Note the steep drop in voltage during the takeoff climb to below 20 V at time 0 and the subsequent voltage recovery by 57 s.

The period from 692 to 885 s in the figures shows the battery load when the pilot resumes control to land the airplane and get the airplane in position to operate the motors on the runway. Operating the motors on the runway to the end of

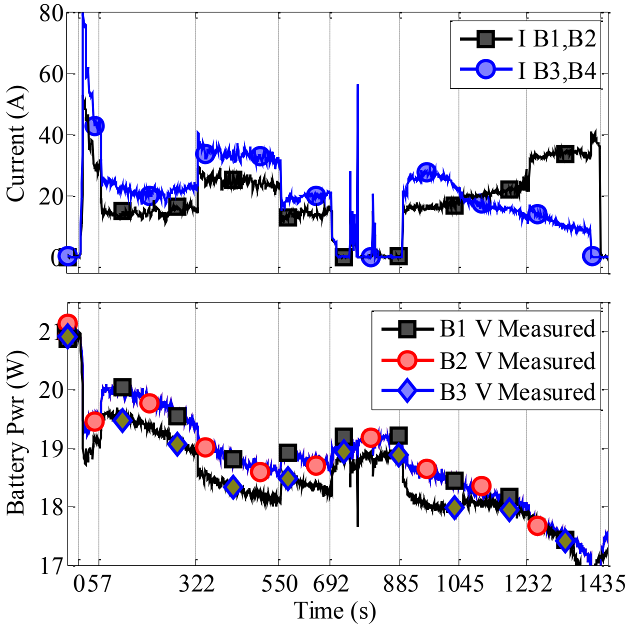


Figure 8. Battery current and Voltage profiles measured over sample flight

discharge for validation purposes, in lieu of flying until end of discharge, is more desirable as it abates the risk of a ”dead stick” landing or a potential crash.

The period between 885 to 1435 s captures the ground discharge of the aircraft’s batteries. This phase of the flight test is used to generate a ground truth measurement of the flight time that the batteries would have been able to support had the plane stayed in the air. For this segment, the pilot initially set the throttle to match the RPM range observed during the 75% constant throttle phase of the flight between 550 s and 692 s. For a constant throttle setting in the runway discharge, the propeller RPM, shown in Figure 7, trends downward and eventually drops below 4000 RPM. This is due to the weakened battery condition in the latter part of the discharge. Because the ground discharge is intended to imitate flight, the throttle was increased at time 1232 sec to raise the RPM above 4000 so as to meet output requirements for cruise flight.

The motors were stopped at 1435 s, after the batteries voltages were seen to fall below the 17 V threshold. Note that the net current draw is seen to increase as the net battery voltage decreases between 885 and 1232 s in order to meet the power demand dictated by the throttle set-point.

4. Battery Charge Depletion Prediction

Online battery prognostic algorithms are intended to estimate the present SOC and predict the EOD for the LiPo batteries that are used to power the aircraft’s propeller motors. Separate battery systems are used to power the data acquisition and other flight communications and control hardware. The battery systems are sized such that it is always the case that the batteries powering the propeller motors will be the first to be depleted. For this reason, onboard battery discharge prognostic algorithms and supervisory decision making actions are only concerned with the propeller-driving batteries. Onboard battery charge depletion prediction routines will first estimate the current battery state, then these uncertain state estimates will be propagated into the future using uncertainty modeling of battery physics and uncertainty models for the future demands to be placed on the battery system.

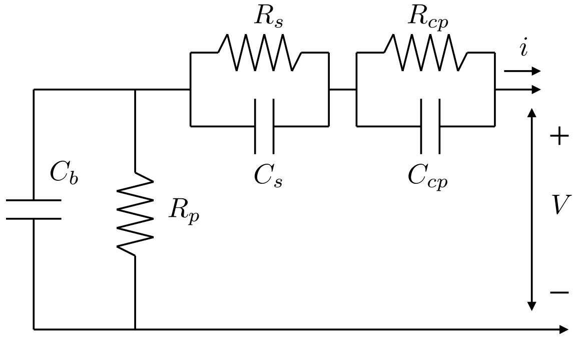


Figure 9. Equivalent circuit battery model

4.1. Battery Modeling

The current SOC of powertrain batteries is estimated using periodic battery current and voltage measurements along with a model that captures the current-voltage characteristics and how they vary as a function of battery charge (Pang, Farrell, Du, & Barth, 2001).

The model used here is an extended version of that presented in (Daigle et al., 2012). Figure 9 shows an equivalent circuit battery model that makes use of three resistor and three capacitor components that are each tuned to match the observed current-voltage dynamics of the batteries used to power the propeller motors on the Edge-540T. Battery charge is stored in the capacitor, C_b, in the equivalent circuit battery model. The R_s, C_s and R_{cp}, C_{cp} circuit element pairs capture battery internal resistance drops and concentration polarization effects, respectively. The resistor R_p is a large parasitic resistance that accounts for the slow battery self-discharge that is seen to occur over weeks or months of storage. Because battery current-voltage dynamics are known to vary as a function of battery SOC some of the resistive and capacitive (RC) components in the equivalent circuit model must be parametrized as functions of battery SOC (Zhang & Chow, 2010). It was decided based on qualitative observation that defining C_b, C_{cp}, and R_{cp} as parameterized functions of battery SOC gave an acceptable trade-off between the number of parameters to be identified and model error.

Table 1. Parameter values used in equivalent circuit model

Parameter	Value	Parameter	Value
q_{max}	2.88×10^4 C	C_s	89.3 F
C_{max}	2.85×10^4 C	R_{cp0}	$1.60 \times 10^{-3}\ \Omega$
C_{Cb0}	19.4 F	R_{cp1}	8.45
C_{Cb1}	1576 F	R_{cp2}	-61.9
C_{Cb2}	41.7 F	C_{cp0}	2689 F
C_{Cb3}	-203 F	C_{cp1}	$-2285\ F$
R_s	2.77×10^{-2}	C_{cp2}	-0.73 F

Battery SOC is defined as:

$$\text{SOC} = 1 - \frac{q_{max} - q_b}{C_{max}} \tag{1}$$

where q_b is the charge stored in the battery, q_{max} is the maximum charge that the battery can hold, and C_{max} is the maximum charge that can be drawn from the battery. The term coulombic efficiency is used to refer to the portion of stored charge that is recoverable during the discharge of the battery. There are some mechanisms including resting the battery that can unlock some of its lost capacity, however, the overall trend is inevitably downward.

C_b, C_{cp} and R_{cp} are parameterized as:

$$C_b = C_{Cb0} + C_{Cb1} \cdot \text{SOC} + C_{Cb2} \cdot \text{SOC}^2 + C_{Cb3} \cdot \text{SOC}^3 \tag{2}$$

$$C_{cp} = C_{cp0} + C_{cp1} \cdot \exp\left(C_{cp2}\left(\text{SOC}\right)\right) \tag{3}$$

$$R_{cp} = R_{cp0} + R_{cp1} \cdot \exp\left(R_{cp2}\left(\text{SOC}\right)\right) \tag{4}$$

where C_{Cb0}, C_{Cb1}, C_{Cb2}, C_{Cb3}, R_{cp0}, R_{cp1}, and R_{cp2} are empirical coefficients that are tuned based on observed current-voltage battery data over a range of SOC.

The current and voltage dynamics of the equivalent circuit model are defined as:

$$\mathbf{x}^B = \begin{bmatrix} q_b & q_{cp} & q_{Cs} \end{bmatrix}^T \tag{5}$$

$$\dot{\mathbf{x}}^B = \begin{bmatrix} -\frac{1}{C_b R_p} & \frac{1}{C_{cp} R_p} & \frac{1}{C_s R_p} \\ \frac{1}{C_b R_p} & -\frac{1}{C_{cp} R_p R_{cp}} & \frac{1}{C_s R_p} \\ \frac{1}{C_b R_p} & \frac{1}{C_{cp} R_p} & \frac{1}{C_s R_p} \end{bmatrix} \mathbf{x}^B + \begin{bmatrix} i \\ i \\ i \end{bmatrix} + \boldsymbol{\xi} \tag{6}$$

$$\mathbf{y}^B = V_p = \begin{bmatrix} \frac{1}{C_b} & \frac{1}{C_{cp}} & \frac{1}{C_s} \end{bmatrix} \cdot \mathbf{x}^B \tag{7}$$

where q_b, q_{cp}, and q_{cs} represent the charge stored in C_b, C_{cp}, and C_{cs} respectively, and the voltage drop across the battery terminals is equal to the sum of the voltage drops across each of the three capacitors.

The fitting of equivalent circuit RC parameters to the observed current-voltage response of the Edge-540T powertrain batteries is described in (Bole et al., 2013). The identified parameter values for the batteries used in the sample flight described in Section 3.3 are given in Table 1.

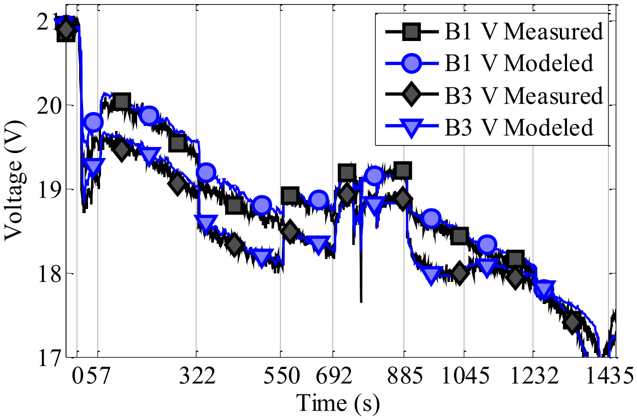


Figure 10. Modeled and measured voltages of batteries B1 and B3 for a sample flight loading profile

Figure 10 shows the predicted and measured voltage profiles for batteries B1 and B3 using the recorded battery current profiles shown in Figure 8. The close match between observed battery voltages and open-loop predictions made using the given loading profile provides a measure of validation for the fitted battery models.

4.2. Estimation

The identified battery model may be used to implement an observer for the internal battery states based on sampled voltage and current data. The unscented Kalman filter (UKF) (Julier & Uhlmann, 1997, 2004) and the particle filter (Arulampalam et al., 2002), are two flexible tools for preforming closed-loop updates of the probabilistic belief in system state estimates based on stochastic (and possibly nonlinear) models of system dynamics. Both approaches operate by drawing samples from probability distributions that are estimated for all of the uncertain parameters in the system state estimate and system dynamics model.

Particle filters use a discrete set of weighted samples, called *particles* to represent the belief in current system state estimates. Particles are sampled stochastically. The number of particles used and the methods used to assign weights and resample particles are design choices that will determine the computational overhead of this approach.

The UKF assumes a general nonlinear form of the state and output equations, and efficiently propagates model and state uncertainties without the need to calculate Jacobians (unlike the extended Kalman filter). The UKF is restricted to additive Gaussian noise random processes; however use of the unscented transform, a deterministic sampling method, allows random variables with non-Gaussian distributions to be incorporated using a minimal set of weighted samples, called *sigma points* (Julier & Uhlmann, 1997). The number of sigma points is only linear in the dimension of the random variable,

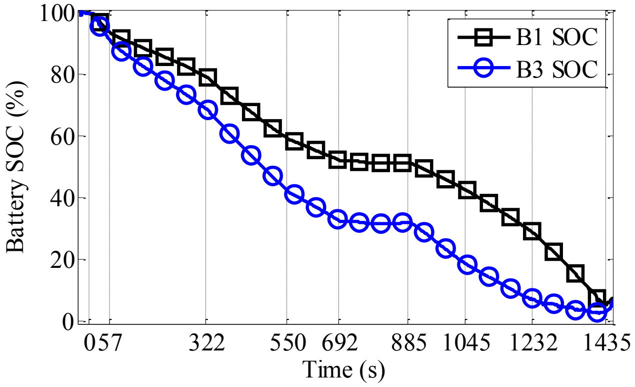


Figure 11. Estimated SOC for batteries B1 and B3 over flight

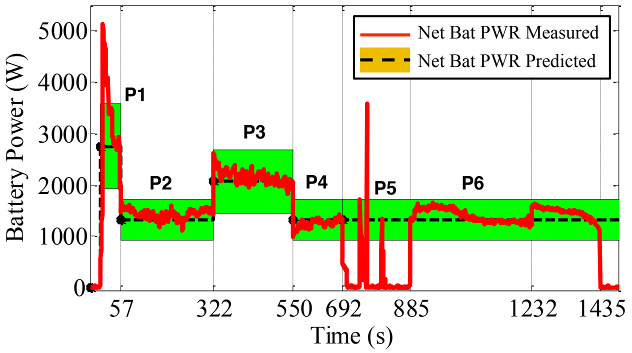


Figure 12. Measured and predicted net battery power consumption over sample flight. The six phases of the sample flight are annotated P1-P6

and so the statistics of the transformed random variable, i.e., mean and covariance, can be computed much more efficiently than by random sampling.

The use of a Kalman filter to update battery SOC estimates based on current and voltage samples is described in (Pang et al., 2001). The output of such closed-loop state estimation techniques will be much less susceptible to initialization and measurement errors than the Coulomb counting method currently used in many battery monitoring systems (Dai, Wei, & Sun, 2006).

Figure 11 shows the evolution of SOC estimates that are produced by a UKF algorithm acting on the voltage and current measurements for batteries B1 and B3 over the sample flight. The prior distribution of the process noise used in the UKF model is small here due to high confidence in the fitted battery model. The measurement noise is also assumed to be small due to accurate current and voltage sensing onboard the vehicle. The low measurement and process noise priors result in negligible uncertainty around SOC estimates in this case study. Therefore, uncertainty in the battery discharge prognostic estimates presented here is caused almost entirely by uncertainty in estimates of the future loads to be placed on system batteries.

4.3. Prediction

The batteries are considered to have reached the EOD condition when the battery voltage knee is observed under flight loads. The battery voltage knee is characterized by a precipitous drop in voltage that occurs at low battery SOC. The voltage knee is observed at approximately 1410 seconds for all of the powertrain battery voltages plotted in Figures 8 and 10.

In order to make a prediction, the future loading on the batteries and the corresponding uncertainty must be characterized in some fashion. In the sample flight described in Section 3.3, the aircraft flight plan is composed of the following phases;

1. takeoff and climb to ~200 meters (duration = 60 s)
2. maintain altitude, set throttle to 75% (duration = 275 s)
3. maintain altitude, set throttle to 85% (duration = 228 s)
4. maintain altitude, set throttle to 75% (duration = 142 s)
5. land and taxi down the runway (duration = 193 s)
6. fully deplete batteries by spinning the propeller at similar RPMs those observed in phase 4.

The purpose of spinning the propeller at similar RPMs to that observed during 75% throttle flight is to safely obtain an approximate measurement for the amount of flight time that would have been supported by the battery pack if the aircraft had continued to be flown at the approximately the same speed as it was going in phase 4. This measurement allows comparison between battery EOD predictions made at various points over the sample mission, and the EOD time observed experimentally.

Subtracting the time spent landing and taxing on the runway (during which vary little power is drawn) from 1410 seconds, gives an approximate measurement for the time at which the EOD condition would have been observed if the aircraft had continued to be flown at the same speed as in phase 4 of the flight plan until EOD, $1410 - 193 = 1217$ seconds.

Figure 12 shows measured and predicted net battery power consumption over the sample flight. Future battery loading is estimated for the sample flight using knowledge of phases 1-4 in the flight plan enumerated above. The battery load predicted for phase 4 of the flight plan is assumed to continue until the battery system EOD condition is observed. The mean battery load prediction, shown as a dashed line in Figure 12, is estimated using a prior characterization of the energy required to perform aircraft maneuvers that are specified by a given flight plan. Characterization of net battery power required for the Edge-540T UAV to fly a given set of maneuvers is described in (Bole et al., 2013). A uniform probability distribution ranging ±30% from the mean battery load estimate is added to future load estimates, denoted by the green band in Figure 12, to account for unmodeled system dynamics.

Given a state estimate, the prediction step proceeds by simulating that state estimate through time up until the EOD point, incorporating assumed future loading. For both the particle filter and UKF, the state estimate is represented by a set of weighted samples. The distribution of future loading, i.e., the battery power demand, is assumed to be constant over each leg of the flight. The constant demand over each leg of the flight is assumed to be drawn from a uniform distribution ranging ±30% from the mean battery load estimate. Here, we use the unscented transform to sample from the future loading distribution and obtain sigma points that cover the distribution, as described in (Daigle et al., 2012).

The equivalent circuit battery model is initialized for EOD prediction by taking a sample from the battery state estimate. The battery model is then simulated until EOD, using a battery load input that is drawn from the estimated distribution of possible future battery power demands. Just three sigma points are required to represent the uniformly distributed future battery loading demands. For a uniform distribution with the value used for the free parameter in the unscented transform yields the three points as the mean of the distribution and its two endpoints, thus naturally capturing the input bounds. The belief distribution for current system states is represented using nine samples from the UKF. The EOD belief estimate is then constructed by generating EOD estimates for all combinations of the nine samples representing battery state estimates, and the three samples of future battery load ($9 \cdot 3 = 27$ samples). This is much more efficient than when using the much greater number of samples from the particle filter.

5. Assessment

Figure 13 shows upper and lower bounds of RUL and EOD predictions for batteries B1 and B3, obtained using UKF for state estimation and the unscented transform for sample point identification. Although these projections are computed for all four batteries during the flight, only batteries B1 and B3 are shown in these graphs to avoid clutter. The figure shows the median, and range of the probability distribution calculated from the three simulated sigma points. RUL and EOD predictions are shown against the ground truth measurement of RUL and EOD obtained by fully discharging the batteries at flight loads on the ground. The ground truth measurements for EOD and RUL are shown as dashed lines in the figure. The ground truth RUL measurement is calculated by subtracting the current time from the EOD time, which was estimated to occur at 1217 seconds in Section 4.3 if the UAV had continued to fly at approximately the same speed as it was going in step 4 of the flight plan. An accuracy cone defined by $\alpha = 0.3$ is also shown for reference purposes. Predictions are shown at 60 s intervals. Here, we see that while the estimated RUL and EOD distributions seem to be converging to the true RUL and EOD as the flight progresses, there is a consistent over-estimation of EOD for battery B1 and a consistent under-estimation of EOD for battery B3 over the sample flight.

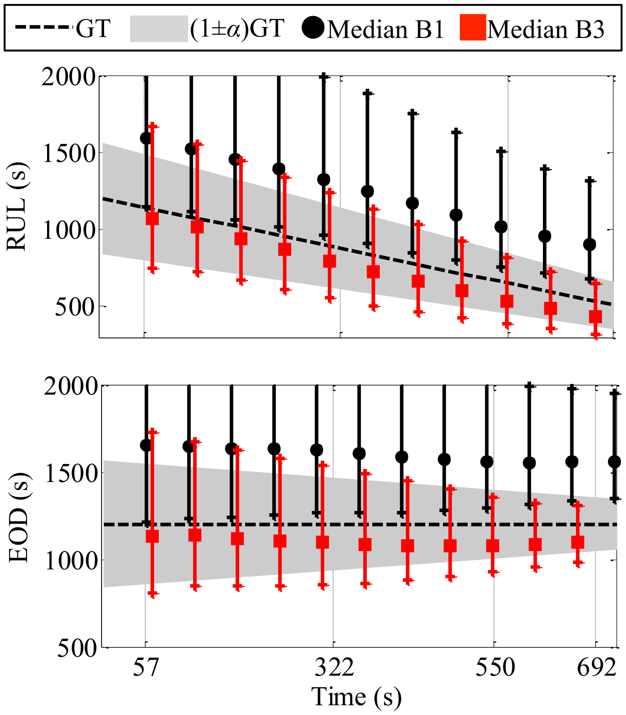


Figure 13. RUL and EOD predictions with future input trajectories drawn from a uniform distribution using the unscented transform.

The EOD estimation bias apparent in Figure 13 is explained by the fact that the battery demand modeling used assumes a constant split of power between the two tandem mounted propeller motors. However, as was shown in Figures 7 and 10, the power split between motors M1 and M2 is seen to change abruptly near the end of discharge.

Although the RUL estimates for B3 are seen to be slightly conservative, the estimates are fairly stable over the flight, and the estimated RUL probability density functions are seen to mostly fall within the 30% relative accuracy cone shown for the sample flight. This result is a considerable improvement on previous particle filter based implementations of battery EOD prognostics, that used the average of battery current over a finite window to estimate the future battery loading over a flight (Saha et al., 2011, 2009; Saha & Goebel, 2008; M. E. Orchard et al., 2012). Not only are the predictions more accurate, but they are more stable as well.

Figure 14 shows the mean RUL and EOD predictions obtained using a particle filter algorithm described in (Saha et

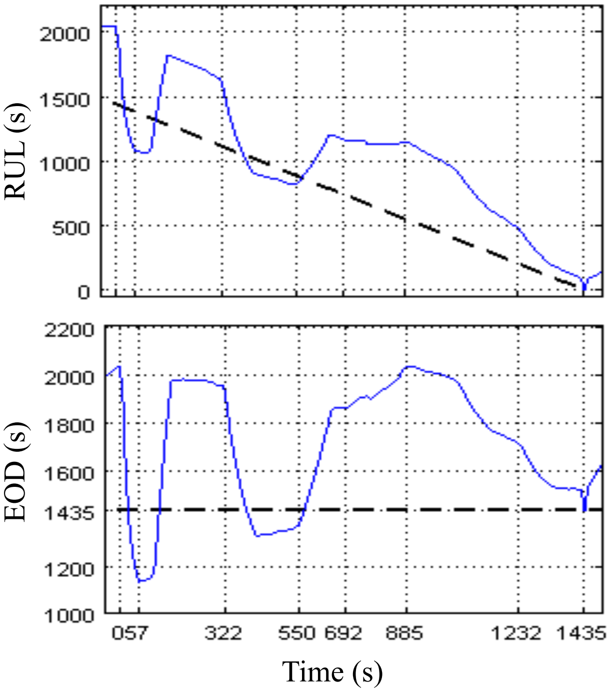


Figure 14. RUL and EOD predictions using particle filtering with future load estimated using a 100 second window of past current data

al., 2009), with a 100 s moving average used to project future battery current draw. Observe that the high current draw during takeoff does not reflect in reduced RUL until 57 s later; likewise, when the 75% constant throttle segment is started at the 57 s mark, the RUL does not stabilize until about 50 s later; and again at the 322 s throttle change. This effect is most pronounced on the throttle transition from the ground motor off to the ground discharge at 885 sec. This 50 s hysteresis could be reduced by reducing the width of the 100 s sliding window but at the expense of getting more jittery RUL projections which makes for a more unnerving operator experience. With a small window, the RUL predictions tend to fluctuate because when the load over the window is higher than what the load will be in the future, RUL is underpredicted, and when the load over the window is lower than the future load, RUL is overpredicted.

6. Conclusions

This paper described the application and assessment of battery charge depletion prognostics onboard an unmanned all-electric subscale vehicle. The paper also described a sample flight test during which a preplanned flight plan is autonomously flown by the aircraft. Predictions of available flight time remaining where generated based on measurements of the battery state of charge and knowledge of a flight plan. After a predetermined time the aircraft was landed and the propeller was spun at flight speeds to obtain a measurement of the actual flight time remaining. A unscented Kalman filtering based battery discharge prognostic algorithm was demonstrated, and the results were compared against a particle filtering based prognostic algorithm that had been published previously. Future work will involve additional flight tests, and improved input prediction methods that take advantage of known flight plans for improved accuracy.

Acknowledgment

The project support by NASA's AvSafe/SSAT and OCT/ACLO are respectfully acknowledged.

References

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.

Bole, B., Teubert, C., Chi, Q. C., Edward, H., Vazquez, S., Goebel, K., & Vachtsevanos, G. (2013). SIL/HIL replication of electric aircraft powertrain dynamics and inner-loop control for V&V of system health management routines. In *Annual conference of the prognostics and health management society.*

Dai, H., Wei, X., & Sun, Z. (2006). Online soc estimation of high-power Lithium-Ion batteries used on HEVs. In *IEEE international conference on vehicular electronics and safety.*

Daigle, M., & Goebel, K. (2013, May). Model-based prognostics with concurrent damage progression processes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(4), 535-546.

Daigle, M., Saxena, A., & Goebel, K. (2012). An efficient deterministic approach to model-based prediction uncertainty. In *Annual conference of the prognostics and health management society.*

Harrup, P., & Davis, S. (2010). 21st century technology propels electric aircraft into the "blue yonder". *Power Electronics Technology*, *36*, 16-21.

Hogge, E. F., Quach, C. C., & Hill, B. L. (2011). *A data system for a rapid evaluation class of subscale aerial vehicle.* NASA/TM-2011-217145.

Julier, S. J., & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of the 11th international symposium on aerospace/defense sensing, simulation, and controls* (pp. 182–193).

Julier, S. J., & Uhlmann, J. K. (2004, March). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401–422.

Luo, J., Pattipati, K. R., Qiao, L., & Chigusa, S. (2008, September). Model-based prognostic techniques applied to a suspension system. *IEEE Transactions on*

Systems, Man and Cybernetics, Part A: Systems and Humans, *38*(5), 1156 -1168.

Orchard, M., Tobar, F., & Vachtsevanos, G. (2009, December). Outer feedback correction loops in particle filtering-based prognostic algorithms: Statistical performance comparison. *Studies in Informatics and Control*, *18*(4), 295-304.

Orchard, M. E., Cerda, M. A., Olivares, B. E., & Silva, J. F. (2012). Sequential Monte Carlo methods for discharge time prognosis in Lithium-Ion batteries. *International Journal of Prognostics and Health Management*, *3*, 1-12.

Pang, S., Farrell, J., Du, J., & Barth, M. (2001). Battery state-of-charge estimation. In *American control conference.*

Saha, B., & Goebel, K. (2008). Uncertainty management for diagnostics and prognostics of batteries using Bayesian techniques. In *IEEE aerospace conference.*

Saha, B., & Goebel, K. (2009, September). Modeling Li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the annual conference of the prognostics and health management society 2009.*

Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009). Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Transactions on Instrumentation and Measurement*, *58*(2), 291-296.

Saha, B., Koshimoto, E., Quach, C. C., Hogge, E. F., Strom, T. H., Hill, B. L., ... Goebel, K. (2011). Battery health management system for electric UAVs. In *IEEE aerospace conference.*

Zhang, H., & Chow, M.-Y. (2010). Comprehensive dynamic battery modeling for PHEV applications. In *IEEE power and energy society general meeting.*

BIOGRAPHIES

Quach Cuong Chi got his M.S. from the School of Physics and Computer Sciences at Christopher Newport University in 1997. He is a staff researcher in the Safety Critical Avionics Systems Branch at NASA Langley Research Center. His research areas include development and testing of software for airframe diagnosis and strategic flight path conflict detection.

Brian M. Bole graduated from the FSU-FAMU School of Engineering in 2008 with a B.S. in Electrical and Computer Engineering and a B.S. in Applied Math. Brian received a M.S. degree in Electrical Engineering from the Georgia Institute of Technology in 2011, and he is currently pursuing a Ph.D. Brian's research interests include: analysis of stochastic processes, risk analysis, and optimization of stochastic systems. Brian is currently investigating the use of risk management and stochastic optimization techniques for optimal adaptation of active component load allocations in robotic and aviation applications. In a previous project, Brian worked with the Georgia Tech EcoCar team to develop an energy management controller for optimizing the fuel economy of a charge sustaining hybrid electric vehicle.

Edward Hogge received a B.S. in Physics from the College of William and Mary in 1977. He has provided engineering services to the government and currently is employed by Northrop Grumman Technical Services. He has recently been supporting aviation safety research through the implementation of electronic systems for subscale remotely piloted aircraft and through commercial aircraft simulation. He is a member of the American Institute of Aeronautics and Astronautics.

Sixto Vazquez Mr. Vazquez obtained MSEE from Old Dominion University in 1990 and BSEE from the University of Puerto Rico in 1983. He has developed real-time 3D graphical simulations to aid in the visualization and analysis of complex sensory data. Has developed techniques to interactively process, analyze, and integrate sensory data from multiple complex, state-of-the-art sensing technologies, i.e. FMCW Coherent Laser Radar range measuring system, Bragg grating Fiber Optic Strain Sensing system, etc., into simulation. In recent years, Mr. Vazquez has developed software for the ardupilot and associated ground station.

Matthew Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems.

Dr. José R. Celaya is a research scientist with SGT Inc. at the Prognostics Center of Excellence, NASA Ames Research Center. He received a Ph.D. degree in Decision Sciences and Engineering Systems in 2008, a M. E. degree in Operations Research and Statistics in 2008, a M. S. degree in Elec-

State of Charge Estimation of Lithium-ion Batteries Considering Model and Parameter Uncertainties

Zhimin Xi[1], Rong Jing[2], Xiao Guang Yang[3], and Ed Decker[4]

[1,2]*University of Michigan - Dearborn, Dearborn, MI, 48128, USA*

zxi@umich.edu
rjing@umich.edu

[3,4]*Ford Motor Company, Dearborn, MI, 48121, USA*

xyang11@ford.com
edecker6@ford.com

ABSTRACT

Up to date, model and parameter uncertainties are generally overlooked by majority of researchers in the field of battery study. As a consequence, accuracy of the SOC estimation is dominated by the model fidelity and may vary from cell-to-cell. This paper proposes a systematic framework with associated methodologies to quantify the battery model and parameter uncertainties for more effective battery SOC estimation. Such a framework is also generally applicable for estimating other battery performances of interest (e.g. capacity and power capability). There are two major benefits using the proposed framework: i) consideration of the cell-to-cell variability, and ii) accuracy improvement of the low fidelity model comparable to the high fidelity without scarifying computational efficiency. One case study is used to demonstrate the effectiveness of the proposed framework.

1. INTRODUCTION

Hybrid Electric Vehicles (HEVs), Plug-in Hybrid Electric Vehicles (PHEVs) and Electric Vehicles (EVs) are gaining the popularity in automotive industry. Lithium-ion (Li-ion) battery is the most promising power source for HEVs/PHEVs/EVs due to its light weight, high energy density and relatively low self-discharge compared to Ni-cad and NiMH batteries. Battery performances of interest, such as State-of-Charge (SOC), capacity and power capability, have been extensively studied due to their importance in real HEVs/PHEVs/EVs applications (Santhanagopalan & White, 2008; He et al., 2012; He et al., 2012). Battery SOC, similar to the remaining gas in the gasoline vehicles, is of particular interest and should be exactly known at any operating time. Unfortunately, the percentage of the charge remaining in the battery, namely the battery SOC, is not a directly measurable quantity and thus should be accurately estimated instead.

Coulomb counting is the most widely employed method for tracking the SOC provided that the initial SOC is known (Ng et al., 2008). Otherwise, Kalman Filter (KF) and Extended Kalman Filter (EKF) are typical methods for fast SOC estimation based on various equivalent circuit models of the Li-ion battery (Plett, 2004). Other methods in machine learning have been recently explored in the SOC estimation and/or degradation parameter (e.g., capacity) estimation (Andre et al., 2012; Lee et al., 2011; Santhanagopalan and White, 2010; Hu et al., 2012; He et al., 2013).

A common limitation in battery SOC estimation is ignorance of the battery parameter uncertainty if they are the same type and come from the same manufacturer. As a consequence, accuracy of the SOC estimation may vary from cell-to-cell. Another important limitation is unawareness of the model uncertainty which comes from the fact that no battery model can truly represent the physical system without any error in various operating conditions. Since the SOC estimation is conducted on the basis of the assumed 'perfect' battery model, any level of model uncertainty will cause biased SOC estimation regardless of the specific numeric algorithms. Up to date, the aforementioned two limitations are generally overlooked by majority of researchers in the field of battery study.

Contribution of this paper is to propose a systematic framework with associated methodologies to quantify the battery model and parameter uncertainties for more effective battery SOC estimation. Such a framework is also generally applicable for estimating other battery performances of interest (e.g. capacity and power capability).

The structure of the paper is organized as follows. Section 2 illustrates model and parameter uncertainties of the battery model. Section 3 presents a framework with associated

Xi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided

methodologies to quantify the battery model and parameter uncertainties for more effective SOC estimation. Case study is presented in Section 4 for the demonstration of the proposed framework. Finally, conclusion is made in Section 5.

2. MODEL AND PARAMETER UNCERTAINTIES

This section first presents a brief review of Li-ion battery models, and then illustrates model uncertainty and parameter uncertainty in the following two subsections, respectively.

2.1 Battery Model

Battery model can be classified into two groups: electrochemical models and equivalent circuit models. Electrochemical models are physics-based models where a set of governing non-linear differential equations are used to predict the battery internal state variables which can be further related to the typical battery performances of interest. They are generally treated as high fidelity models requiring high computational effort and thus are not desirable in real time battery SOC and State of Health (SOH) diagnosis. Equivalent circuit models are simplified physics-based models where a capacitor (or a voltage source) and resistors are used to represent the diffusion process and internal impedance of the battery cell, respectively. Compared to the electrochemical models, they can be viewed as low fidelity models with less accuracy but very high computational efficiency. Thus, majority of the Battery Management System (BMS) employs the equivalent circuit models for battery SOC and SOH diagnosis.

A discrete time state-space model (see Eq, (1)) is typically used to estimate the battery hidden states (e.g. SOC and capacity) using the KF/EFK on the basis of the equivalent circuit models.

$$\begin{cases}\mathbf{x}_{k+1}=f(\mathbf{x}_k,\mathbf{u}_k)+\mathbf{w}_k\\ \mathbf{y}_k=g(\mathbf{x}_k,\mathbf{u}_k)+\mathbf{v}_k\end{cases} \tag{1}$$

where $\mathbf{x}_k$ is the state vector at the k^{th} step; $\mathbf{u}_k$ means the input vector (e.g. current); $\mathbf{w}_k$ is the process noise; $\mathbf{y}_k$ is the output vector (e.g. terminal voltage); $\mathbf{v}_k$ is the measurement noise of the output vector; $f(\cdot)$ is the state transition function; and $g(\cdot)$ is the equivalent circuit model that relates the output vector with the input and hidden state vectors.

2.2 Model Uncertainty

When battery models are accurate and reliable, the output vector predicted from the model would be exactly the same as the true test results under various operating conditions. It is worth noting that above statement is valid when satisfying three conditions: i) no model parameter uncertainty, ii) no numerical algorithm uncertainty, and iii) no test error. However, models are generally built on the basis of many assumptions and simplifications and therefore model uncertainty may always exist because there is probably no ideal model which can predict the real physical system without any model bias.

Eq. (2) shows one specific state-space model used for the SOC estimation.

$$\begin{cases}x_{k+1}=x_k-\eta T i_k / C_n+w_k\\ y_k=OCV(x_k)-i_k R+h_k+v_k\end{cases} \tag{2}$$

where x_{k+1} is the SOC at the $(k+1)^{\text{th}}$ step; $\eta T i_k$ is the coulomb accumulation for given charging/discharging efficiency (η), current (i_k) and time accumulation T; $C_n$is the nominal capacity. The second equation is the equivalent circuit model which builds the functional relationship for terminal voltage y_k, OCV, internal impedance R and voltage change h_k due to the hysteresis effect. A one-state hysteresis model is further expressed in Eq. (3).

$$h_k = \exp\left(-\left|\frac{\eta_i i_{k-1}\gamma\Delta t}{C_n}\right|\right)h_{k-1}+\left(1-\exp\left(-\left|\frac{\eta_i i_{k-1}\gamma\Delta t}{C_n}\right|\right)\right)M \tag{3}$$

where γ is a positive constant which tunes the rate of decay; and M is a polarization coefficient.

For one specific battery cell, model uncertainty is the deterministic difference between the predicted terminal voltage y_k and the true terminal voltage, which indicates the model inadequacy for representing the actual functional relationship under various battery operating conditions. In general, parameter uncertainty which will be illustrated in the next subsection is coupled with the model uncertainty and should be taken into account when characterizing the model uncertainty. Thus, model uncertainty becomes the stochastic difference between the predicted terminal voltage y_k and the true terminal voltage. For the equivalent circuit model considered above, a corrected model after introducing the model uncertainty can be defined in Eq. (4).

$$y_k=OCV(x_k)-i_k R+h_k+\delta(i_k,x_k,\ C_k)+v_k \tag{4}$$

where $\delta(\cdot)$ is the model uncertainty function which is also referred as the model bias in model validation community.

Development of an effective model uncertainty characterization approach can improve model prediction accuracy in the intended uses of the model. Such process is especially useful to improve accuracy of a low fidelity model (e.g. equivalent circuit models with high computational efficiency) comparable to a high fidelity model (e.g. electrochemical models with low computational efficiency) so that battery SOC and SOH diagnosis can be conducted more effectively.

2.3 Parameter Uncertainty

A common mistake in battery SOC and SOH diagnosis is ignorance of the fact that the battery used in laboratory test is different with others in real operation due to various sources of uncertainties (e.g. physical uncertainty) even if they are the same type and come from the same manufacturer. Physical uncertainty can be viewed as the cell-to-cell variability due to manufacturing tolerance. Correspondingly, parameter uncertainty is the realization of the physical uncertainty in the specific battery models.

According to the battery model in Eq. (3) and Eq. (4), model parameters (e.g. internal impedance R, decay factor γ, etc.) contain uncertainty due to the cell-to-cell variability and thus should be quantified appropriately. Otherwise, the battery SOC estimation may be accurate for one cell under perfect calibration condition, but not so accurate for other cells. The accuracy variability depends upon two factors: i) significance of the parameter uncertainty and ii) sensitivity of the accuracy with respect to the parameter uncertainty.

2.4 Remarks on Model and Parameter Uncertainty

In model calibration, the objective is to maximize the agreement between the model prediction and the experimental data. A common approach for simplification is to disregard the model uncertainty by maximizing the agreement between the original model prediction and the experimental data through calibration of unknown model parameters (e.g., internal impedance R, decay factor γ, etc.). It is apparent that the calibrated model parameters may not be the true values. This is acceptable in model calibration because models are treated more pragmatically to increase their predictive power for one or several specified battery cells. However, if the objective is to improve the model prediction accuracy for the population of the battery cells under various operating conditions, it is risky to directly use model calibration technique because the model prediction could be inaccurate out of the calibration domain due to incorrect calibration of the model parameters and ignorance of the model uncertainty.

3. PROPOSED FRAMEWORK FOR SOC ESTIMATION

The proposed systematic framework is shown in Fig. 1 with consideration of the model and parameter uncertainties for more effective SOC estimation. There are two major benefits using the proposed framework: i) consideration of the cell-to-cell variability and ii) accuracy improvement of the initial battery model. Basically, this framework enables user to select a low fidelity battery model with high computational efficiency without scarifying the accuracy because a corrected battery model with high accuracy can be later obtained through characterizing the model uncertainty. Furthermore, battery SOC or SOH diagnosis becomes probabilistic instead of deterministic so that confidence of the estimation is available. Following subsections elaborate each step of the framework.

3.1 Determination of Initial Battery Model

The initial battery model ideally should include major input factors which influence the output performances. For example, OCV, SOC, charge/discharge current, hysteresis and temperature are important inputs for predicting the terminal voltage accurately and thus they should be considered in the empirical model. The purpose is to have a good base model with reasonable accuracy so that model uncertainty can be more effectively quantified to improve the model prediction accuracy. Otherwise, more noise factors would be included in the quantified model uncertainty such that the corrected model prediction would provide much wider confidence intervals to account for the ignorance of the important factor. In this study, the equivalent circuit model in Eq. (2) and Eq. (3) is used without considering the temperature effect. Thus, testing is conducted in the room temperature to eliminate the noise factor from various temperature levels for the SOC estimation.

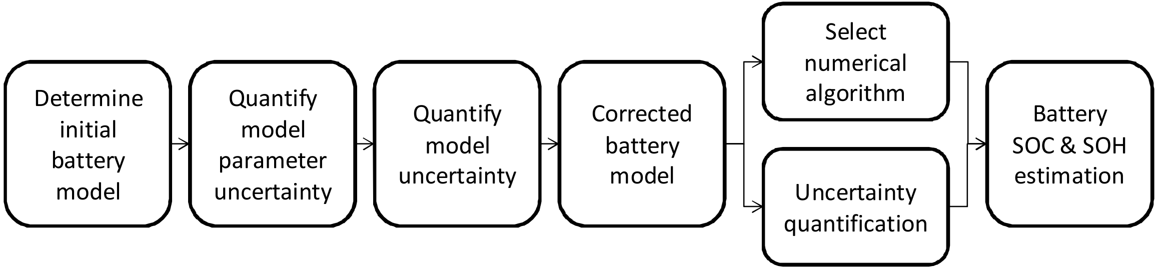

Determine initial battery model
Quantify model parameter uncertainty
Quantify model uncertainty
Corrected battery model
Select numerical algorithm
Uncertainty quantification
Battery SOC & SOH estimation

Figure 1. Flowchart of the proposed framework for battery SOC and SOH estimation

3.2 Quantification of Parameter Uncertainty

A certain number of battery cells should be determined to account for the cell-to-cell variability based on the parameter uncertainty. For example, defined pulse power capability tests of five battery cells result in five sets of model parameters after calibration of each battery cell individually. Thus, five random realizations are available for quantifying the uncertainty of each model parameter. The issue of data sufficiency needs to be addressed in this step.

Typically, random parameters can be classified into two groups: i) irreducible random parameter and ii) reducible random parameter. The irreducible random parameters are characterized using Probability Density Functions (PDFs) with sufficient information. The reducible random parameters are derived from the lack of information for describing the uncertainty. For example, parameters, i.e., the mean and the variance of the PDF, or even distribution types are uncertain unless sufficient information is collected.

This study considers parameter uncertainty as irreducible random parameter and use Maximum Likelihood Estimation (MLE) to select the optimal distribution for each model parameter. The statistics of the random parameter is represented by the statistical parameter Θ of a candidate distribution. For example, in the case of a normal distribution, the parameter is defined as $\Theta=\{\mu_\theta, \sigma_\theta\}$, which includes the mean and standard deviation. Thus, Θ is the calibration parameter and needs to be identified. The statistical model calibration using MLE is formulated as

$$\text{Maximize } L(V_i \mid \Theta) = \sum_{j=1}^{M} \log_{10}\left[f\left(v_{ij} \mid \Theta\right)\right] \qquad (5)$$

where $L(\cdot)$ is the likelihood function; $f(\cdot)$ is the PDF of V_i for a given Θ; i means the i^{th} model parameter; and M is the number of available data. A candidate distribution pool, including Normal, Lognormal, Weibull, Beta, Gamma, and Uniform, is defined and the optimal distribution is determined by the maximum likelihood value among candidate distributions.

3.3 Quantification of Model Uncertainty

The objective for quantifying model uncertainty is to improve the model prediction accuracy by adding the identified model uncertainty to the original model as shown in Eq. (4). A two-step calibration procedure is proposed to accurately characterize the model uncertainty (or model bias) in various battery operating conditions.

Step 1: calibrate unknown model parameter V_i through calibration experiments as described in section 3.2;

Step 2: calibrate the model uncertainty using the statistical calibration technique at several defined battery operating conditions using the Design of Experiment (DOE) technique.

Then three steps are used to obtain the model uncertainty in various battery operating conditions with the aid of the response surface.

Step 1: construct response surfaces for the central moments (e.g., mean and standard deviation) of the model uncertainty using the moving least square method;

Step 2: calculate the central moments of the model uncertainty at any given operating condition on the basis of the response surfaces;

Step 3: approximate the distributions of the model uncertainty at any given operating condition.

Response surface of the model uncertainty plays a critical role in the process of obtaining corrected model prediction. Its accuracy mainly depends on three factors including: i) nonlinearity of the model uncertainty in various battery operating conditions; 2) amount and location of the identified model uncertainty using the DOE technique; and 3) numerical algorithm of the response surface method.

3.4 Correction of Initial Battery Model

The corrected battery model is shown in Eq. (6) by adding the identified model uncertainty to the initial battery model. Furthermore, model parameter uncertainty is characterized by optimal PDFs to account for the cell-to-cell variability. The corrected model is a statistically validated model and is expected to produce more accurate and robust SOC estimation.

$$\begin{cases} x_{k+1} = x_k - \eta T i_k / C_n + w_k \\ y_k = OCV(x_k) - i_k R + h_k + \delta(i_k, x_k, \dot{x}_k) + v_k \end{cases} \qquad (6)$$

3.5 Selection of Numerical Algorithm

KF has been widely used in many applications to estimate the hidden state for the linear state-space model. As an extension, EKF applies for non-linear state-space model using the linear approximation at each estimation step. Other KF related approaches are also reported such as adaptive KF, unscented KF, etc. All KF related approaches use linearity and Gaussian noise assumptions, which could cause numerical estimation error for the non-linear model with non-Gaussian noise. In that scenario, Particle Filter (PF) (Orchard & Vachtsevanos, 2009) is more appropriate to approximate the state PDF using Bayesian approach and avoiding such assumptions. However, PF is much more computational expensive than EKF. This study employs the EKF for demonstration of the proposed framework due to its reasonable accuracy and efficiency.

3.6 Uncertainty Quantification

The objective is to estimate SOC of the battery in a statistical manner using the validated battery model and the EKF. All sources of uncertainties are considered in model parameters and identified model uncertainty/bias. Essentially, the SOC estimation becomes an Uncertainty Quantification (UQ) process to quantify the distribution of the battery SOC subject to the input uncertainties from the model parameter, model uncertainty/bias and the measurement and process noise.

A common challenge in UQ is a multi-dimensional integration to quantify probabilistic nature of system responses. Neither analytical multi-dimensional integration nor direct numerical integration is possible for large-scale engineering applications. Other than those approaches, existing approximate methods for UQ can be categorized into five groups (Youn et al., 2008): i) sampling method, ii) expansion method, iii) the most probable point (MPP)-based method, iv) response surface approximate method, and v) approximate integration method. This study uses sampling method (i.e. the Monte Carlo Simulation) for UQ.

4. CASE STUDY

This section presents a case study to demonstrate the effectiveness of the proposed framework for SOC estimation.

4.1 Background

EIG C020 battery cells are used in this case study with the nominal capacity of 20Ah. Four battery cells were connected parallel to four channels of the battery cycler (Arbin BT2000). All experiments, including static capacity test, energy efficiency test, HPPC test, OCV-SOC test and FUDS test, were conducted at the room temperature (25 ^{o}C). Capacity test results are shown in Table 1 with the mean and standard deviation equal to 19.66 Ah and 0.037 Ah, respectively. Efficiency test results are listed in Table 2 for both charge and discharge efficiency with an overall mean of 0.997. In this study, the nominal capacity (=19.66 Ah) and charge/discharge efficiency (=0.997) are treated as constant value for the SOC estimation. OCV-SOC curve is obtained as shown in Fig. 2.

Table1. Static capacity test

	Cell #1	Cell #2	Cell #3	Cell #4	Mean	STD
Capacity	19.628	19.701	19.629	19.682	19.66	0.037

Table 2. Energy efficiency test

	Cell #1	Cell #2	Cell #3	Cell #4	Mean
Discharge efficiency	0.9970	0.9978	0.9982	0.9990	0.9983
Charge efficiency	0.9900	0.9980	0.9980	0.9965	0.9956

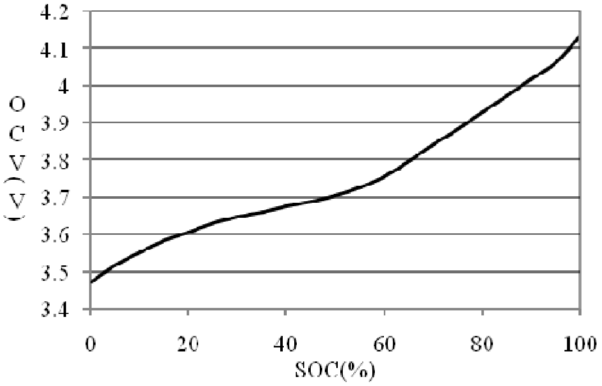


Figure 2. OCV-SOC curve

4.2 Results

Four battery cells were used to study the uncertainty of four model parameters including: i) charging impedance *R*+, ii) discharge impedance *R*-, iii) decay factor *γ*, and iv) polarization coefficient *M*. Table 3 shows that the decay factor *γ* presents no uncertainty and other three parameters present different level of uncertainties. For example, the standard deviation (STD) of charging impedance *R*+ is 5.2% of its mean value. Typically, we should not ignore the parameter uncertainty if the STD is more than 1% of its mean value. Hence, three model parameters were characterized as random parameters and they were assumed to follow normal distribution with identified statistical moments listed in Table 3.

Table 3. Uncertainty quantification of model parameters

Parameter	*R*+	*R*-	*γ*	*M*
Cell #1	0.0023	0.0037	1.6026	0.0172
Cell #2	0.0021	0.0037	1.6026	0.0162
Cell #3	0.0024	0.0036	1.6026	0.0171
Cell #4	0.0022	0.0035	1.6026	0.0163
Mean	0.0025	0.0036	1.6026	0.0167
STD	1.29e-4	9.57e-5	0	5.23e-4
Percentage	5.2%	2.7%	0%	3.1%

The first three battery cells were used as training data to quantify the model uncertainty. In this case study, response surface of the model uncertainty was constructed only for the 1st central moment (i.e. the mean) for simplification. FUDS test was then carried out for cell #4. Mean of the model uncertainty for the cell #4 was estimated for the FUDS test and the results are shown in Fig. 3.

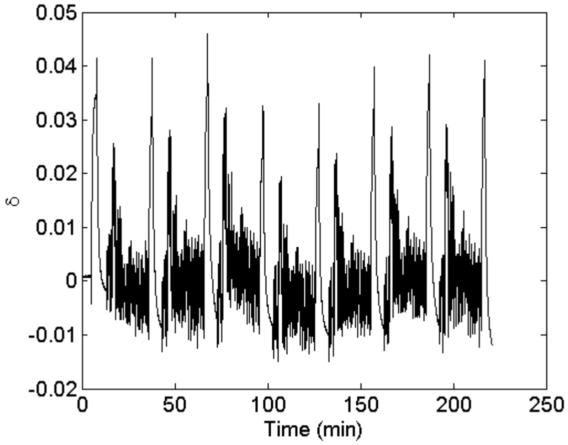
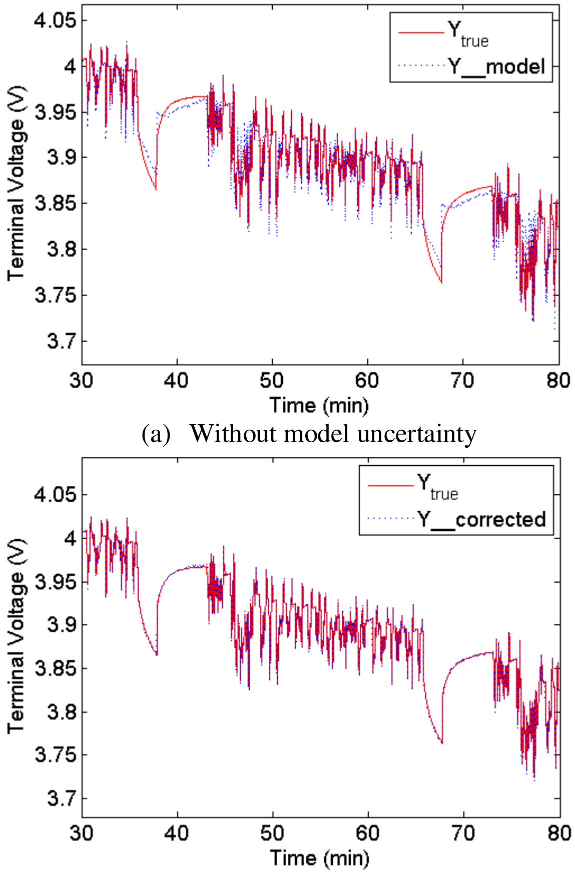


Figure 3. Mean of the model uncertainty of cell #4 for the FUDS test

According to the original battery model, estimated terminal voltage generally agrees well with actual measurement as shown in Fig. 4(a). However, the difference is observable especially during the rest period in between the FUDS cycles, which is mainly due to the model inadequacy for representing the actual physical system. Such model limitation can be overcome by considering the model uncertainty using Eq. (4). Based on the identified model uncertainty (see Fig. 3) for the FUDS test, the corrected terminal voltage agrees much better than the original model as shown in Fig. 4(b).

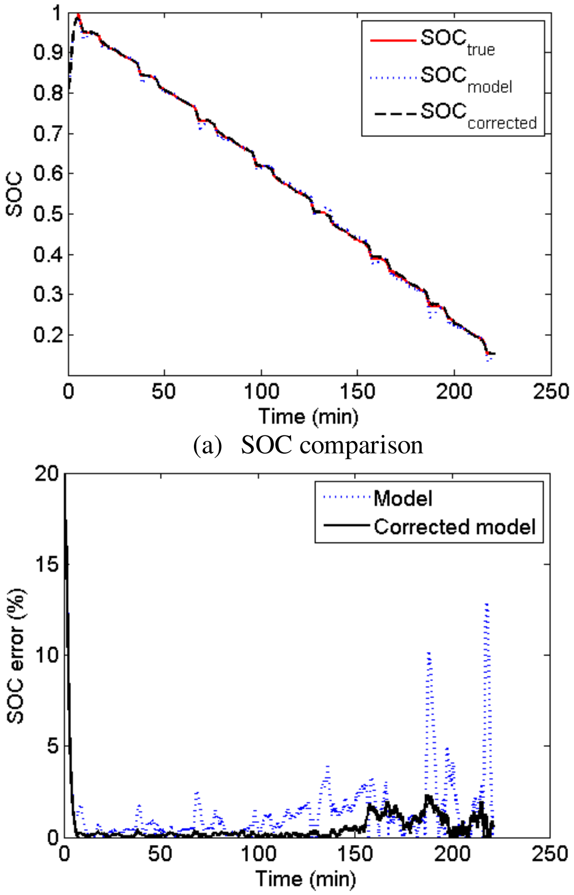


(a) Without model uncertainty

(b) With model uncertainty

Figure 4. Comparison of terminal voltage for the original model and corrected model

SOC estimation was conducted using both original battery model and corrected model with consideration of the model uncertainty. The battery cell #4 was first fully charged with 100% of SOC; and then one FUDS cycle was applied and followed by a constant discharge current and a rest period. Such process was repeated till the SOC reached 10%. SOC calculated using the coulomb counting was treated as the reference value. EKF was used for SOC estimation for both battery models: i) original model without consideration of model uncertainty, and ii) corrected model with consideration of model uncertainty. The initial guess of SOC was set to 80% using the EKF for both models. Results are shown in Fig. 5 with clear indication that accuracy of SOC estimation can be significantly improved by considering the model uncertainty appropriately. In particular, the maximum percentage error (excluding the initial error) of the SOC estimation reaches 14% using the original model, whereas, the error is well below 5% if the model uncertainty is considered.



(a) SOC comparison

(b) Percentage error of SOC

Figure 5. SOC estimation with and without consideration of the model uncertainty

5. CONCLUSION

A novel framework was proposed for quantifying model and parameter uncertainties for battery SOC and SOH diagnosis. Various uncertainty sources should be systematically addressed in order to have reliable battery management system and accurate SOC and SOH diagnosis in real HEV/PHEV/EV applications. In summary, four types of uncertainty play a key role for reliable estimation of the battery performances of interest and they are listed as: i) measurement uncertainty, ii) algorithm uncertainty, iii) model parameter uncertainty, and iv) model uncertainty. Measurement uncertainty includes current and voltage measurement error and has been well considered by most researchers. Algorithm uncertainty focuses on accuracy of numerical algorithms for estimating the battery hidden state. This field of research evolves gradually and typical algorithms include Kalman filter, extended Kalman filter and particle filter. A trade-off between numerical accuracy and efficiency should be considered depending upon specific applications. Model parameter uncertainty is not well considered up to date in the field of battery study. However, as massive products in the foreseeable future, their physical uncertainty due to the manufacturing tolerance should be well addressed. Ignorance of model parameter uncertainty makes unreliable battery SOC and SOH diagnosis. Finally, model uncertainty dominates the accuracy level of battery SOC and SOH diagnosis. Different battery models represent different levels of fidelity comparing to the actual physical system. However, majority of the research focuses on the model development itself. This paper turns an eye on quantifying the model uncertainty so that the low fidelity model can possess the accuracy of high fidelity model without scarifying the computational efficiency. Preliminary case study of the SOC estimation demonstrated the effectiveness of the proposed framework.

ACKNOWLEDGEMENT

Research was supported by Ford Motor Company and we appreciate Professor Chris Mi and Mr. Rui Xiong for sharing part of their test data and having valuable discussion with us.

REFERENCES

Andre, D., Nuhic, A., Guth, T.S., and Sauer, D.U., (2012), Comparative study of a structured neural network and an extended Kalman filter for state of health deternimation of lithium-ion batteries in hybrid electric vehicles. *Engineering Applications of Artificial Intelligence*, v26, n 3, p951-961.

He, H., Xiong, R., Guo, H., and Li, S. (2012), Comparision study on the battery models used for the energy management of batteries in electric vehicles. *Energy Conversion and Management*, v64, p113-121.

He, W., Williard, N., Chen, C., and Pecht, M., (2012), State of charge estimation for electric vehicle batteries under an adaptive filtering framework. *Prognostics & System Health Management Conference*, Beijing, China, 2012.

He Y., Liu X.T., Zhang C.B., Chen Z.H., 2013, A new model for State-of-Charge (SOC) estimation for high-power Li-ion batteries. *Applied Energy*, v101, p808–814.

Hu C., Youn B.D., and Chung J., (2012), A multiscale framework with extended Kalman filter for lithium-ion battery SOC and capacity estimation. *Applied Energy*, v92, p694–704.

Lee S., Kim J., Lee J., and Cho B.H., (2011), Discrimination of Li-ion batteries based on Hamming network using discharging–charging voltage pattern recognition for improved state-of-charge estimation. *Journal of Power Sources*, v196, n4, p2227–2240.

Orchard, M.E., and Vachtsevanos, G.J., (2009), A particle-filtering approach for on-line fault diagnosis and failure prognosis. Transctions of the Institute of Measurement and Control, v31, n3/4, p221-246.

Plett, G.L., (2004), Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 2. Modeling and identification. *Journal of Power Sources*, v134, n2, p262-276.

Plett, G.L., (2004), Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 3. State and parameter estimation. *Journal of Power Sources*, v134, n 2, p277–292.

Ng, K.S., Moo, C.S., Chen, Y.P., and Hsieh, Y.C., (2008), Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied Energy*, v86, p1506-1511.

Santhanagopalan, S., and White, R.E., (2008), State of charge estimation for electrical vehicle batteries. *IEEE, 17th International Conference on Control Applications, Part of 2008 IEEE Multi-conference on System and Control*, San Antonio, Texas, USA, September 3-5, 2008.

Santhanagopalan S., and White R.E., (2010), State of charge estimation using an unscented filter for high power lithium ion cells. *International Journal of Energy Research*, v34, n2, p152–163.

Youn, B.D., Xi, Z., and Wang, P., (2008), Eigenvector Dimension-Reduction (EDR) method for sensitivity-free probability analysis. *Structural and Multidisciplinary Optimization*, v37, p13-28.

BIOGRAPHIES

Zhimin Xi is an Assistant Professor in the Department of Industrial and Manufacturing Systems Engineering at University of Michigan – Dearborn. He received his B.S. and M.S. degree in Mechanical Engineering from Beijing University of Science and Technology in 2001 and 2004, respectively. He received his Ph.D. in Reliability Engineering from University of Maryland – College Park in 2010. He has over 30 publications in prestigious journals and international conferences. He is the two-time winners of the Best Paper Award from ASME – Design Automation Conference in 2008 and 2013, respectively. His research interests are reliability engineering, prognostics and health management, and energy storage device.

Rong Jing is a graduate student in the Department of Industrial and Manufacturing Systems Engineering at University of Michigan – Dearborn. She received her B.S. degree in Industrial Engineering from Xi'an Jiaotong University in 2012. She is the one-time winner of the Best Paper Award from ASME – Design Automation Conference in 2013. Her research interests are diagnostics, prognostics and health management.

Xiao Guang Yang received his Ph.D. from Zhejiang University in 1995, and then worked at University of Hawaii and Penn State Univ. In 2005, Dr. Yang joined Ford for HEV programs. He is a Technical Expert and Supervisor in the Energy Storage and Materials Research Department. He currently focuses on battery modeling, benchmark, and hybrid vehicle battery system simulation and validation. He is a member of USABC (Advanced Battery Consortium Technical Committee). He has 50 journal publications, 7 Ford Technical Reports, 3 US patents and 3 US patent applications.

Ed Decker is a battery cell integration and product development engineer for the Ford Motor Company. He has two degrees from the University of California (Biology and Chemical Engineering). He has worked in battery development since 1984 and spent 16 years working on batteries for the consumer electronic market. He joined Ford Motor Company in 2010 and is currently working on developing batteries for BEV and PHEV applications.

Fleet-wide Diagnostic and Prognostic Assessment

Alexandre Voisin[1], Gabriela Medina-Oliva[2],
Maxime Monnin[2], Jean-Baptiste Leger[2], Benoit Iung[1]

[1] *Centre de Recherche en Automatique de Nancy (CRAN), Université de Lorraine, UMR 7039 CNRS-UHP-INPL, Faculté des Sciences-1er Cycle - BP239, 54506 Vandoeuvre-Les-Nancy Cedex – France*

alexandre.voisin@univ-lorraine.fr
benoit.iung@univ-lorraine.fr

[2] *PREDICT 19, Avenue de la Forêt de Haye, CS 10508, 54519 Vandoeuvre-Lès-Nancy, FRANCE*

gabriela.medina-oliva@ predict.fr
maxime.monnin@predict.fr
jean-baptiste.leger@predict.fr

ABSTRACT

In order to anticipate failures and reduce downtime, “predictive diagnostic” aims not only at warning about the failure events before they occur but also at identifying the causes of degradation leading to such detections. Then, based on the results of predictive diagnostic, “prognostic” aims at estimating the remaining useful life in order to plan a maintenance action before unit performances are affected. However, these are complex tasks. To overcome these difficulties, the notion of fleet may be very useful. In the present paper a fleet is composed of heterogeneous units (mainly components but could be systems or sub-systems) that are grouped together considering some similarities. The fleet can provide capitalized data and information coming from other members of the fleet for the improvement/development of the diagnostic/prognostic models. In order to achieve PHM with a fleet-wide dimension, it is thus necessary to manage relevant knowledge arising from the fleet taking into account heterogeneities and similarities amongst components, operational context, behaviours, etc. This paper will focus mainly in the formalization of a data-driven prognostic model considering a fleet-wide approach. The model is based on a prognostic approach of the system health using Relevant Vector Machine. The proposed model is based on historical data coming from similar units of a fleet. The heterogeneity of the monitored data is treated by assessing a global health index of the units. The proposed approach is shown on a case study. This case study illustrates how the fleet dimension facilitates predictive diagnostic and the definition of the prognostic model in the marine domain.

1. INTRODUCTION

PHM involves the following processes: monitoring the process variables of a current situation, assessing the health of the system, prognosticating the Remaining Useful Life (RUL) of the system and making decision for maintenance action. In that sense, the data coming from the different variables of the process is also used for evaluating and monitoring a global indicator representative of the health state of a unit. The health state allows to supervise easily the degradation behaviour and to detect early enough drifts in operations (Rizzolo et al., 2011). If the health state is not satisfying, then predictive diagnostic could be performed. This process allows to identify the causes of a degradation before a failure occurs. Based on the potential degradation modes producing a drift in operation, the representative variables of this degraded component are used to predict the degradation trajectory and to assess the remaining time to reach a threshold, for instance a specified performance loss level. This time is called Remaining Useful Life. Finally the results are used for decision-making in order to select the maintenance actions to be performed in order to solve the drift.

Implementing a PHM approach at a system level requires the consideration of failure rates of different equipment built on different technologies (mechanical, electrical, electronic or software natures) (Verma et al., 2010) whose behaviour can vary all along the different phases of their lifecycle (Bonissone and Varma, 2005). Moreover, for predictive diagnostic (i.e. diagnostic of drift before failure occurs), maintenance operators/engineers need to analyze the alarms and the symptoms behavior/evolution to understand which components may have caused the symptoms and the reasons for the abnormal behavior of the component. This analysis

Voisin et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

needs to consider the operational context of the symptoms in order to understand the abnormal situation since it influences the component behavior. Finally, prognostic requires some specific model for each degradation, each operational condition, and each material part. Such number of dimension implies that the efforts (according to the type of model, number of data, laboratory tests…) needed for the definition of the model have to be important. Moreover, prognostic deals with the estimation of the future and thus uncertainty appears. However uncertainty could be reduced when more efforts are made (Pecht, 2010).

However to improve PHM processes for large and complex systems such as power plants, ships and aircrafts, one possible approach is to take advantage of the fleet dimension. This dimension can provide knowledge and data to improve diagnostic and prognostic models (Medina-Oliva et al., 2013).

A fleet shall be viewed as a set of systems, sub-systems and equipment. In this paper, the naval domain is addressed but the proposed approach can be broadened to other domains. Hence, in the following a unit of a fleet will be considered as a system (e.g. ship), a sub-system (e.g. propulsion or electric power generation) or equipment (e.g. diesel engine, shaft...) depending on the nature of the study. To be in accordance with the need of improving PHM at the fleet level, an original methodology is proposed in this paper wherein individual knowledge (of each unit) is capitalized for reuse purpose in order to improve PHM activities such as prognostic. To take advantage of the individual knowledge at the fleet level, a semantic model is used for the PHM activities in the naval domain. Such a semantic model enables to reuse particular data, such as maintenance history, reliability analysis, failure analysis, data analysis at a fleet level in order to provide knowledge. As data become available, prognostic process could benefit from more contextual information.

2. PROBLEM STATEMENT

Prognostic process aims at determining the Remaining Useful Life (RUL) of a unit on which a degradation is running. Some literature review, such as (Byington et al., 2002; Jardine et al., 2006; Heng et al., 2009), propose an overview of this domain and consider classification among the prognostic models. (Byington et al., 2002) propose the first classification and classify the prognostic model into three categories:

- Model based approach issued from physical laws of the degradation,
- Data based approach issued from data or indicator monitored on the system,
- Experience based approach mainly issued from reliability model.

In this paper the aim is to benefit from the knowledge acquired during the operation of every unit of the fleet, i.e. events that occurred and have been solved, in order to solve the event occurring on the present unit (Medina-Oliva et al., 2012b). The objective to benefit from the stored knowledge is subject to these constraints:

- Units are heterogeneous (e.g. technically, structure, mission, environment…) since in the naval domain every ship is highly customized. Moreover, even if units are of the same kind (same technical features), the mission they have to fulfil as well as the environment in which they are evolving have a significant impact on the degradation behaviour.
- Signals are heterogeneous. Indeed, signals are heterogeneous in two ways. The first one is for the same kind of unit, since they are evolving in different environment, with different mission… monitored signals show some significant variations. The second deals with the technical differences among units. In that sense, units could have different number of sensors since they are not technically identical. For instance, if engines have different number of cylinders hence, the monitoring of cylinder temperature means that the number of signals is also different.
- Knowledge about degradation is application/technical oriented since it is mainly supported by FMECA/HAZOP. Hence in the corresponding monitoring databases, the structure of the fault/degradation tree might show some differences.
- The current situation to be prognosticated is partially defined. Indeed, predictive diagnosis aims at finding the running degradation at its early stage. Hence only partial knowledge is available and based on symptoms.

Moreover, in order to benefit from the latest information, since units are on-line monitored the proposed approach aims at integrating all the available information as soon as it is available through its integration at the fleet level. Such integration can be performed almost “on-line” through communication channel such as satellite or with some delay through USB hard disk for instance.

The proposed approach is dedicated not to work as a single tool but together with some experts of the corresponding field. For instance, in (Medina-Oliva et al., 2013), the authors show how experts can perform predictive diagnosis using the fleet knowledge. For this goal, the experts are using an iterative process in order to select a target sub-fleet that contains the proper information to solve the case under study (Medina-Oliva et al., 2012b).

Furthermore, one has to consider some constraint arising from the industrial context. Those constraints will help in the choice of well-fitted tools to support the fleet wide approach:

- The nature of the monitoring systems embedded in the ships. As for industrial systems, there exist several systems, which do not share a common conceptual data model (Umiliacchi et al., 2011).
- Monitored data is real data. The signal embed a part of randomness due to, for instance, measurement noise, singular behaviour…
- Due to unique service life of every unit, there exist some heterogeneity between the measured signals. Hence, the sensor signals of a degradation processes can be captured by the probabilistic nature of the prognostic tool.

3. PHM AT THE FLEET LEVEL

For PHM activities, one of the industrial realities is the lack of capitalization of knowledge and model reuse which represents high costs and efforts for the enterprises (Weber et al., 2012) (Medina-Oliva et al., 2012a). In some fields such as the naval one, units are very customized leading to heterogeneous units. This fact limits mainly data and knowledge capitalization and exploitation.

To tackle this issue the fleet dimension can provide enough information and data to improve/perform PHM activities. In that sense, when searching non-identical but similar units a higher volume of data becomes available to reduce uncertainty (e.g. more confidence on the hypothesis generation about the causes producing a drift or more information about the degradation trajectory of a unit). However, most of the existing fleet-wide approaches treat identical units either for the definition of thresholds based on the data of the fleet (Patrick et al., 2010), technical solution capitalization (Reymonet et al., 2009) or RUL estimation based on a similarity-based approach (Wang et al., 2008). The fact of comparing similar units has rarely been addressed as a whole in the literature. To deal with this issue, this paper is based on a methodology that leads to search non-identical but similar units. To do it, knowledge about different and general characteristics of units was formalized within an ontology (Medina-Oliva et al., 2012b), (Monnin et al., 2011a). This knowledge allows to group heterogeneous units based on shared common characteristics that are relevant for a given situation. Indeed, an expert determines the criteria (i.e. characteristics) to be matched in an iterative process. These criteria depend on the partial knowledge of the current situation, the unit under study, the goal of the expertise (here, predictive diagnosis and prognostic), and the expert itself.

Regarding the prognostic techniques to be used, it has to be defined according the constraints previously defined. First, as the units are on-line monitored, time series data of either sensors or indicators are available for processing. Hence, data-based techniques are well fitted. A review of these techniques has been proposed by (Jardine et al., 2006). Among them, we chose techniques that process the past degradation time series. This choice has been guided since the definition of the unit population used to solve the current case is iterative, i.e. some units are dynamically removed or added. Hence the chosen prognostic model has to be able to integrate quickly new information, in our case time series, to perform its computation. Two example of such techniques are available in (Liu et al., 2007; Wang et al., 2008). (Lui et al., 2007) propose to compute match matrixes that are images of the fitting between the multidimensional time series, the current and the past one, for every past time series. Then, in every image, the best similarity indexes are selected and an Auto Regressive Moving Average model (ARMA) (Box & Jenkins, 1976) predicts the time to failure. Finally, the global RUL is computed by combining the ARMA models according to their degree of similarity. (Wang at al., 2008) propose to compute, first, a health index from the multi dimensional current time series in order to get a mono-dimensional time series. Then, they use Relevance Vector Machine (RVM) and Sparse Bayes Learning (SBL) techniques (Tipping, 2001) in order to synthetize the mono-dimensional signal in a few numbers of kernels. The online prediction process employs the background health information for the health prognostic using the Similarity Based Interpolation (SBI) technique. Moreover, (Wang et al., 2008) mention: “This framework also enables the continuous update of the background health information through offline Sparse Bayes Learning and continuous update of the prognostic results in real-time with new sensory signals through SBI… The SBL process can be carried out individually for different training unit which enables the background health knowledge to be built sequentially without complicated retraining process and updated as more offline training units are gradually available.”

Secondly, the online update of the available knowledge can be satisfied. For that purpose, the operator computing the health index has to be determined for every unit in the fleet. Then, once a new event makes new knowledge available, the RVM learning through SBL make knowledge available for prognostic. The resulting information is very sparse and does not require too much space to be stored. Moreover, its use in the on-line phase does not require a huge amount of computing facilities.

Thirdly, since the considered units are heterogeneous, for instance they do not have the same number of cylinder, the monitored signals cannot be handle in the multi-dimensional signal space. Hence, the match matrix cannot handle this aspect. On the contrary, as (Wang et al., 2008) compute a mono-dimensional health index from the multidimensional time series, it is possible to compare the evolution of 2 health indexes even if the underlying units have different number of signals/indicators.

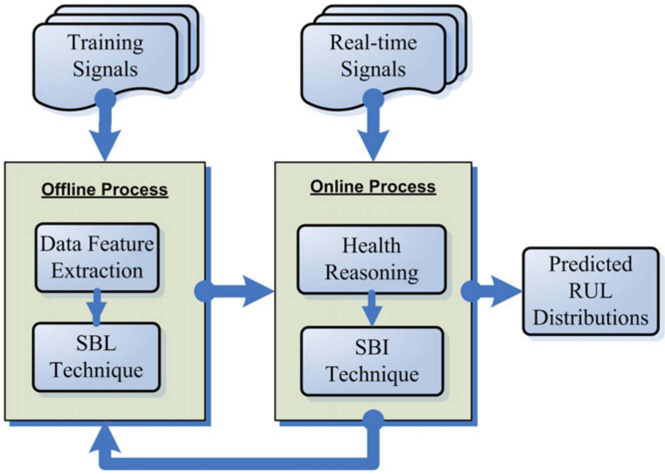


Figure 1. A generic framework for structural health prognostic (Wang et al., 2008)

4. PROPOSED PHM FLEET-WIDE APPROACH

The global fleet-wide approach (Figure 2) is performed in the same way as classical PHM historical based prognostic technics in two stages. The first stage (Figure 2a) consists in determining the hypothetical events causing the deviation (i.e. predictive diagnosis). The result of this first step is a set of solved event that are similar to the actual event under investigation. This set of event is then used in the second stage of the approach (Figure 2b) as historical data in order to performed prognostic.

4.1. Fleet-wide diagnostic approach

The proposed fleet-wide approach allowing case-reuse could bring benefits to almost all PHM activities (Monnin et al., 2011a), (Monnin et al., 2011b), (Medina-Oliva et al., 2012b). Some of them are: PHM solution engineering development/improvement, predictive diagnostic and prognostic model definition.

For the predictive diagnostic, the objective is to identify the causes that produce a drift on operations before failure occurs. To facilitate this task, information/data of past events is capitalized thanks to the semantic model (Monnin et al., 2011b), (Medina-Oliva et al., 2012b). This way it is possible to reuse all the historic data about the real causes producing the abnormal behaviour found among the selected population (Figure 3). As a matter of fact, every time an abnormal situation is studied the experiences such as the alerts detection and the operational context, the real root causes and past maintenance actions, could be capitalized allowing to establish an improvement feedback loop. In that sense feedback about all the individuals composing the selected fleet could be used to obtain more representative statistics based on fleet-wide past experiences, in order to solve a current situation (e.g. alert detection). This approach eases the identification of the real causes and reduces the downtime for a given situation. Furthermore, historic data about the real causes is used in order to build what we called a "fleet-wide populated causal tree" (Figure 9). This kind of tree shows statistics based on the capitalized data found in the fleet. Moreover, the user is guided by the thickness of the linking-lines to search of the most probable causes that produce an abnormal behaviour (e.g. degradation/deviation). The lines that link one degradation to another are thicker as the number of occurrence of events is higher. This way the user can explore different level of causalities in order to identify the most probable root-cause of an abnormal behaviour before the failure occurs and impacts the systems performances. Once the most probable cause has been identified, it is possible to identify the set of units that have presented this cause in order to reuse this population for prognostic purposes.

4.2. Fleet-wide prognostic approach

The prognostic process conforms to the one proposed by (Wang et al., 2008). We add a selection of the on-line stage using population selection according to the fleet wide approach.

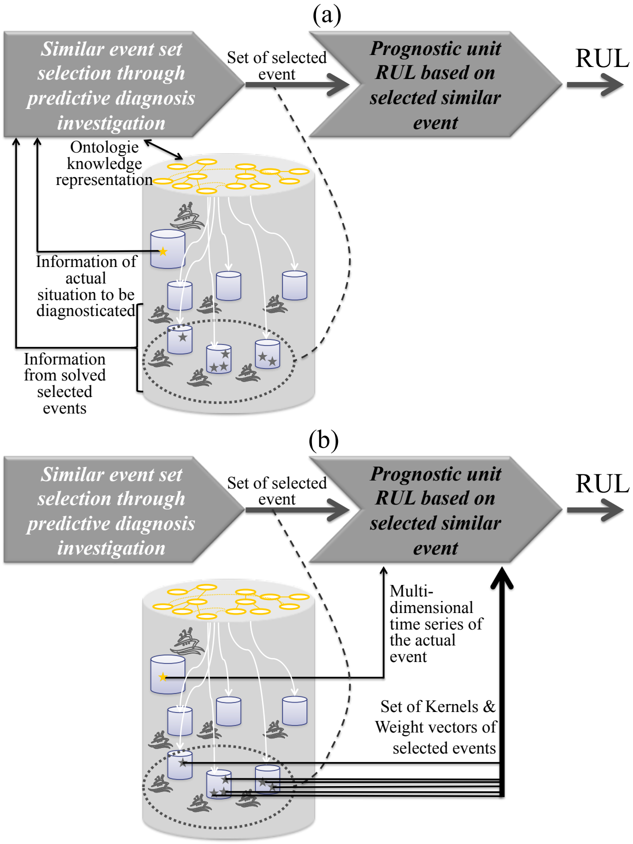

(a)
Similar event set selection through predictive diagnosis investigation
Set of selected event
Prognostic unit RUL based on selected similar event
RUL
Ontologie knowledge representation
Information of actual situation to be diagnosticated
Information from solved selected events
(b)
Similar event set selection through predictive diagnosis investigation
Set of selected event
Prognostic unit RUL based on selected similar event
RUL
Multi-dimensional time series of the actual event
Set of Kernels & Weight vectors of selected events

Figure 2. PHM fleet wide diagnostic (a) and prognostic (b) proposed approach process

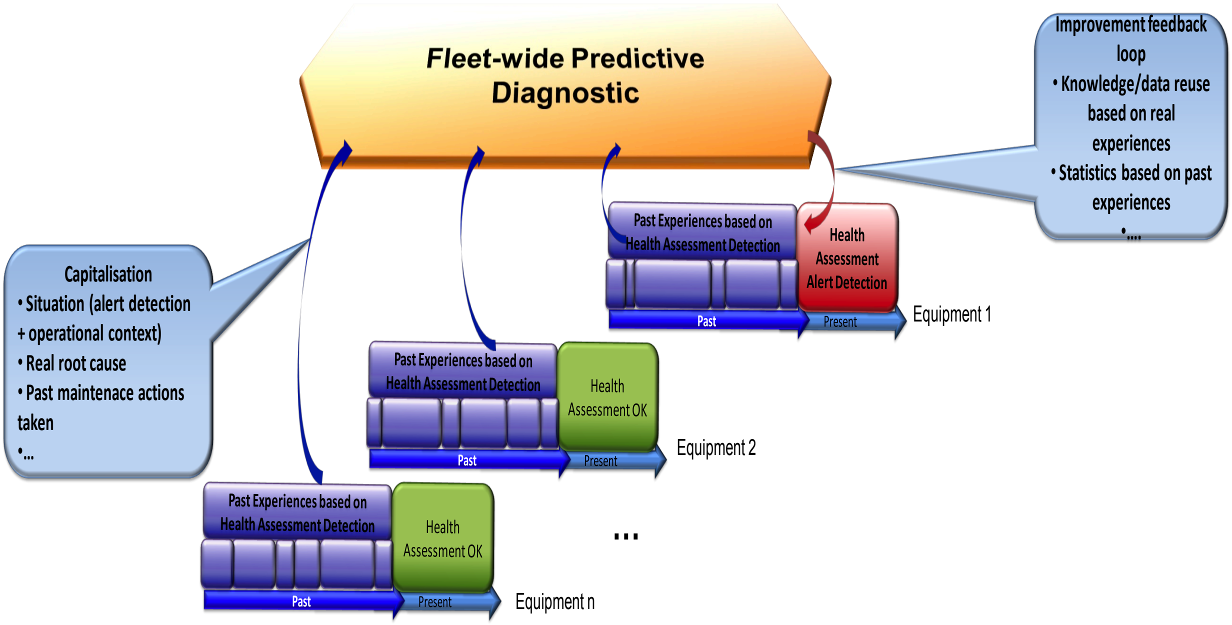

Fleet-wide Predictive Diagnostic
Improvement feedback loop
• Knowledge/data reuse based on real experiences
• Statistics based on past experiences
•....
Capitalisation
• Situation (alert detection + operational context)
• Real root cause
• Past maintenace actions taken
•...
Past Experiences based on Health Assessment Detection
Health Assessment Alert Detection
Past
Present
Equipment 1
Past Experiences based on Health Assessment Detection
Health Assessment OK
Past
Present
Equipment 2
...
Past Experiences based on Health Assessment Detection
Health Assessment OK
Past
Present
Equipment n

Figure 3. PHM Monitoring and Fleet-wide Diagnostic simplified process

4.2.1. Off line stage

The off line stage is composed (Figure 4) of (a) the determination of the aggregation function for health index and of (b) Sparse Bayes Learning (SBL) for health time series.

(a) Health Index computation

Wang et al. use a linear data transformation matrix T such that:

$$T = (Q^T Q)^{-1} Q S_H \qquad (1)$$

Where, Q is composed of both faulty (degraded) and nominal multi-dimensional signals and SH is a {0,1} matrix corresponding to every element of Q according to its state, i.e. 0 for degraded and 1 for healthy. T is able to transform any set of multi-dimensional signal into a mono dimensional signal of health index.

For the purpose of our approach, T has to be determined for every unit, event type and operating condition. On one hand, the computation of the matrix is not time consuming neither required complex data selection (2 sets of data: normal and degraded). On the other hand, this job has to be performed for every unit and every event type since degradation signals changes according to these 2 features. Both of them are easily identifiable. Moreover, operational conditions mode are influencing degradation signals as well, but are more hardly identifiable. Hence, some work is required for such a purpose. Then, normal and degraded signal are extracted as well as operational mode identification, and T matrixes are computed.

(b) off training scheme with SBL

For the sake of conciseness, we do not present the SBL. For more details, one can refer to (Wang et al. 2008) or to the original paper of Tipping (2001). The SBL is a generalized linear model in a Bayesian form and it shares the same functional form of the Support Vector Machine (SVM). Tipping has formulated this generalized linear model in a Bayesian form, named the Relevant Vector Machine (RVM). It achieves comparable machine learning accuracy to the SVM but provides a full predictive distribution with substantially fewer kernel functions.

The RVM is a special case of a sparse linear model:

$$h(t) = \sum_{i=1}^{N} \omega_i \phi(t, t_i) + \varepsilon(t) \qquad (2)$$

where $\varepsilon(t)$ is the measurement noise, $\omega = [\omega_1, \omega_2, \cdots, \omega_N]$ a weight vector and basis functions are formed of kernel functions $\phi(t, t_i)$ centered at the training point t_i. The sparseness property enables the automatic selection of a proper kernel at each location by pruning all irrelevant kernels. A sparse weight prior distribution can be assigned, in such a way that a different variance parameter is assigned to each weight. Moreover, SBL allows to integrate the uncertainty contained in the health index time series by using the statistics of the coefficients ω of the RVs.

In the fleet repository, the degradation time series, associated to a solved event, are summarized by the Gaussian kernels and the weight vectors (mean and covariance matrix). It represents available knowledge for prognostic purpose. SBL performed every time new event has been solved and a degradation time series has to be integrated.

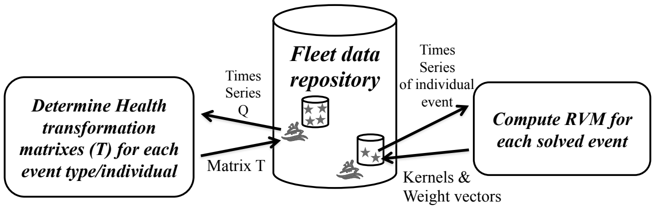

Fleet data repository
Times Series Q
Matrix T
Times Series of individual event
Kernels & Weight vectors
Determine Health transformation matrixes (T) for each event type/individual
Compute RVM for each solved event

Figure 4. Off line stages of prognostics process

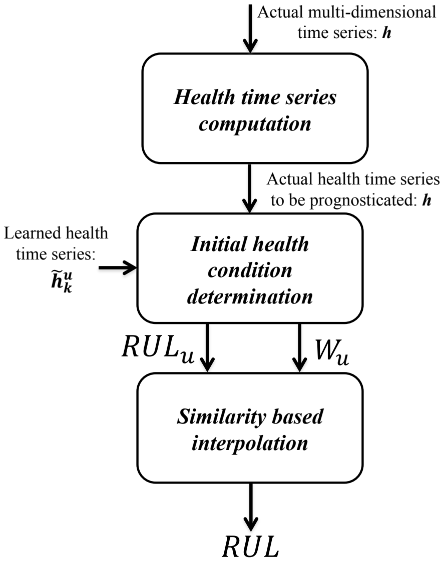

Actual multi-dimensional time series: h
Health time series computation
Actual health time series to be prognosticated: h
Learned health time series: $\tilde{h}_k^u$
Initial health condition determination
RUL_u
W_u
Similarity based interpolation
RUL

Figure 5. On line stages of prognostics process for one learned time series

4.2.2. On-line stage

The on line stage (Figure 5) is split into 3 steps: (a) Actual health time series computation, (b) initial health condition and (c) Similarity Based Interpolation

(a) Actual health time series computation

Based on the transformation matrix T (see eq. 1), the actual multi-dimensional time series is transformed into mono-dimension health time series h.

(b) Initial health condition determination

Lets consider $\tilde{h}$ a time series of health degradation from learned units with a length of $\tilde{l}$. $\tilde{h}$ and h the current one transformed using T with a length of l. Then, if both series represents the same degradation, first $l < \tilde{l}$ and second we supposed that h can be found in $\tilde{h}$. Indeed, as (Wang et al. 2008) explained h and $\tilde{h}$ may have different initial health index at the beginning of the time series due for instance to manufacturing variability or different service life. Hence the RUL estimation (see Figure 6), according to that single $\tilde{h}$, is:

$$RUL = \tilde{l} - l - T_0 \qquad (3)$$

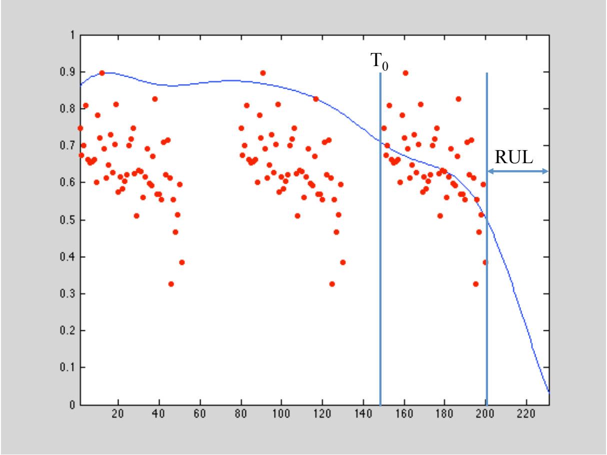


Figure 6. Initial health condition determination

with T_0 the initial time of matching between h and $\tilde{h}$ determined as:

$$T_0 = min_{T_0\in[0,\tilde{l}-l]} \sum_{i=1}^{l}\left(h(t_i) - \tilde{h}(T_0 + t_i)\right)^2 \qquad (4)$$

(c) Similarity based interpolation

Indeed, the computation of the RUL using step (a) is performed for every degradation time series selected using the fleet wide capability, i.e. $\tilde{h}^u$. Similarity based interpolation aims at combining these several RULs. Obviously, h does not match every $\tilde{h}^u$ with the same level of similarity. Hence, the combination of the RUL depends on the degree of matching. For a single unit u, its weight is issued from the matching step:

$$W_u = \left[\sum_{i=1}^{l}\left(h(t_i) - \tilde{h}(T_0^u + t_i)\right)^2\right]^{-1} \qquad (5)$$

The final RUL is computed as:

$$RUL = \frac{1}{W}\sum_u (W_u RUL_u) \quad where\ W = \sum_u W_u \qquad (6)$$

4.2.3. Uncertainty management

As explained earlier, the RVM approach allows to capture uncertainty contained in the data by means of the vector ω and the associated covariance matrix. Hence, for every unit u, instead of having only one $\tilde{h}^u$, for instance the mean curve, one has several $\tilde{h}_k^u$ corresponding to random realization of the weight ω^u. Hence eqs. (3) and (4) have to be computed for all the random realizations of the weight ω^u.

Finally, eq. (6) is re-written as:

$$RUL = \frac{1}{W}\sum_u \left(\frac{1}{W_u}\sum_k (W_k^u RUL_k^u)\right)$$
$$where\ W = \sum_u W_u\, , and\ W_u = \sum_k W_k^u \qquad (7)$$

5. Case-study

To illustrate the feasibility of the proposed approach as well as the added-value, a scenario was developed. This scenario shows how the fleet-wide approach is useful for experts during the decision making process for diagnosis and prognostic purposes. The scenario is developed using an ontology-based fleet-wide software application (Medina-Oliva et al., 2012 & 2013) and KASEM® (Knowledge and Advanced Service for E-Monitoring) e-maintenance software platform.

We consider first the predictive diagnosis of Volvo Penta D16 MG diesel engine. This engine presents one symptom: higher temperature of the turbo-compressor exhaust outlet gas. This symptom points out a degradation on the air-intake system.

5.1. Predictive diagnostic of a diesel engine

The air-intake system of this machine is a turbocharged system. A turbo-compressor consists of a turbine and a compressor connected by a shaft. The compressor draws in ambient air and compresses it. The compressor is connected to the turbine by a shaft and its outlet is routed to the engine cylinder air intake. Exhaust gas from the engine cylinders enters the turbine and expands, performing work on the turbine. The turbine spins the shaft connected to the compressor (Figure 7).

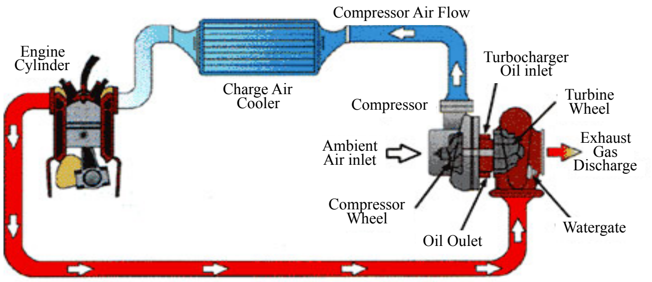


Figure 7. Turbo-charge system

The objective is first to help an expert to extract/retrieve data coming from the fleet in order to solve the diagnostic of this situation. In that sense, the expert should identify which are the most probable root-causes of degradation in the air-intake system (Figure 8) either internal or external causes to the turbo-compressor (Muller et al., 2008).

For the purpose of this example, the fleet is limited to diesel engines. Two hundred eighteen (218) events that occurred on diesel engines (Table 1) are considered. Table 1 presents an extract of the engine units of fleets and their technical features. It is possible to notice that units are heterogeneous, meaning they have different technical features. The ontology-based application aims at helping the expert in the research of similar cases among a heterogeneous fleet of engines that allows the identification of the root-causes. This way, to search the causes of the degradation the application guides the user and proposes different criteria such as the properties or technical features of units. For instance, since there is a degradation on the air-intake system for a turbo-charged engine, the embedded knowledge in the ontology (i.e. classification of engines) allows to select only turbo-charged engines. This criterion is essential to analyze the same type of degradation, for this reason it is necessary to integrate this criterion in the query. This kind of cluster could be relevant for the user since this criterion allows the definition of common and similar characteristics of engines behavior even though they are not identical.

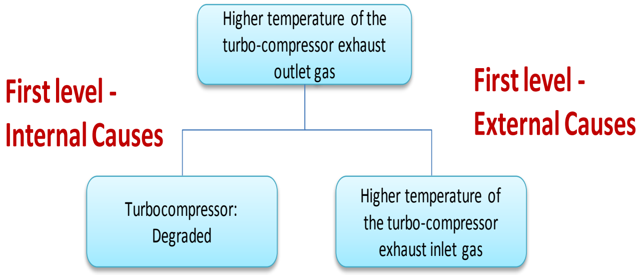

Higher temperature of the turbo-compressor exhaust outlet gas
First level - Internal Causes
First level - External Causes
Turbocompressor: Degraded
Higher temperature of the turbo-compressor exhaust inlet gas

Figure 8. Causality tree about the possible causes of degradation of the air-intake system

Table 1. Extract of engine fleet technical features stored in the data bases

Engine Ref	Output power (kW)	Nb. of Cylinder	Configuration	Engine Speed (rpm)	Tag related to the ontology	Engine cycle	Air admission	Total	Installation
Wärtsilä 12V38	8700	12	V	600	Fuel engine	4	Turbocharged	2	Propulsion engine
Wärtsilä RT-flex50	13960	8	L	124	Fuel oil	2	Turbocharged	2	Propulsion engine
Wärtsilä RT-FLEX82T	40680	9	L	80	Fuel oil	2	Turbocharged	1	Propulsion engine
Baudouin 12M26P1FR	357,94	12	V	1800	Fuel engine	4	Naturally-Aspirated	5	Generator engine
Wärtsilä 16V38	11600	16	V	600	Fuel engine	4	Turbocharged	3	Propulsion engine
Wärtsilä 9L38	6525	9	L	600	Fuel engine	4	Turbocharged	1	Propulsion engine
Wärtsilä 8L38	5800	8	L	600	Fuel engine	4	Turbocharged	1	Propulsion engine
Volvo Penta D16C – AMG	500	6	Ligne	1800	Fuel engine	4	Turbocharged	2	Generator engine
ABC 12VDZC	2652	12	V	1000	Fuel engine	4	Turbocharged	2	Propulsion engine
Baudouin 6 M26 SR P1	331	6	Ligne	1800	Fuel engine	4	Turbocharged	3	Generator engine
Baudouin 12 M26 SR	662	12	V	1800		4	Turbocharged	2	Propulsion engine

Once an ontology-based query is performed among the fleet, the user might be able to investigate the past events that have occurred in the fleet in order to reuse this information for example for predictive diagnostic purposes.

This way, information/data of past events is capitalized. The application allows to reuse all the historic data about the real causes producing the abnormal behavior found among the selected population. Furthermore, historical data about the real causes is used in order to build a "fleet-wide populated causal tree" (Figure 9). When exploring this tree which is based on statistics of fleet-wide past events (not on the signal of the events), it is possible to notice that the most probable cause producing the symptom is a degradation on the poppet valve of the outtake gas, which is delayed to open. Hence, the user can perform a predictive diagnostic guided by an ontology-based application that embeds useful knowledge about the marine domain and that allows the capitalization of data/knowledge within a fleet composed of heterogeneous units.

Then, based on the results of predictive diagnostic, a prognostic will be performed using the health state trajectory of the resulting 74 events that are presenting a problem with the poppet valve. This way it will be possible to estimate the remaining useful life of the Volvo Penta D16 MG diesel engine in order to plan a maintenance action before the engine performances are affected.

The on-line prognostic process in performed on the Volvo Penta D16 MG diesel engine. The first step is to compute the health time series of the engine.

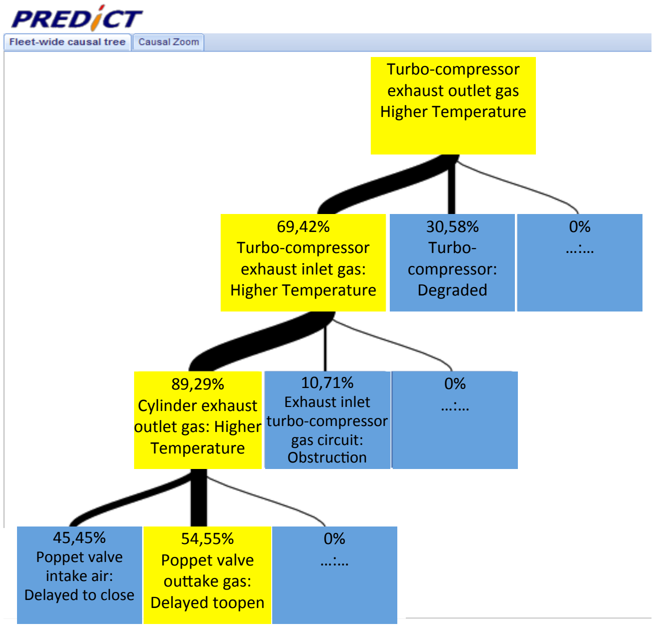

PREDICT
Fleet-wide causal tree
Causal Zoom
Turbo-compressor exhaust outlet gas Higher Temperature
69,42% Turbo-compressor exhaust inlet gas: Higher Temperature
30,58% Turbo-compressor: Degraded
0% ...:...
89,29% Cylinder exhaust outlet gas: Higher Temperature
10,71% Exhaust inlet turbo-compressor gas circuit: Obstruction
0% ...:...
45,45% Poppet valve intake air: Delayed to close
54,55% Poppet valve outtake gas: Delayed toopen
0% ...:...

Figure 9. Fleet-wide *populated* causal tree

5.2. Prognostic based on the obtained fleet-wide population

The second step is the computation of the RULs for every event time series. This part requires to get their degradation background knowledge, i.e. kernel vector, weights and covariance matrix. Then, several $\tilde{h}_k^u$ curves are generated. We show some curves in order to show different level of uncertainties capture by the RVs for k in {1…100} (Figure 11).

For every event u and $\tilde{h}_k^u$ curves, a RUL_k^u is computed. Figure 12 shows the histograms RULs for k in {1,…,10000}. One can notice that the relative dispersion of the histograms do not always correspond to a larger uncertainty in $\tilde{h}_k^u$. For instance, for event 41, $\tilde{h}_k^{41}$'s show some uncertainty (Figure 11c) while the RUL_k^{41}'s do not since a single value has been found. In the same way, RUL_k^{26} and RUL_k^{33} (Figure 12b and d) exhibit the same dispersion while $\tilde{h}_k^{26}$ and $\tilde{h}_k^{33}$ (Figure 11 b and d) do not. Moreover, the contribution of every RUL_k^u in the final RUL through the weight W_k^u, eq. (7), allows to draw histograms for every event as well (Figure 13).

Over all the events, only 68 gave a proper result to be used in the computation of the overall RUL. In such application, the computation of a single RUL does not seem of great relevance. Instead, the analysis of the histograms of the RUL^u would give better information. Such analysis could be performed with different number of histogram class (Figure 14). We take 4 numbers of class, N, between 6 and 13. One can notice between only one mode (Figure 14 b) for N=6, three modes ((Figure 14 a and c) for N=8 and 11 or even four modes ((Figure 14d) for N=13. Such differences could be further investigated by going down the cause-tree, investigating the population homogeneity according their service life, mission… and with the help of engine experts.

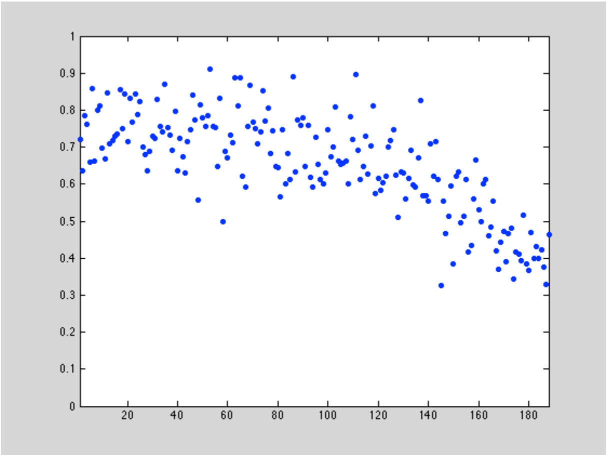


Figure 10: Health time series of the Volvo Penta D16 MG diesel engine with ill-defined running degradation

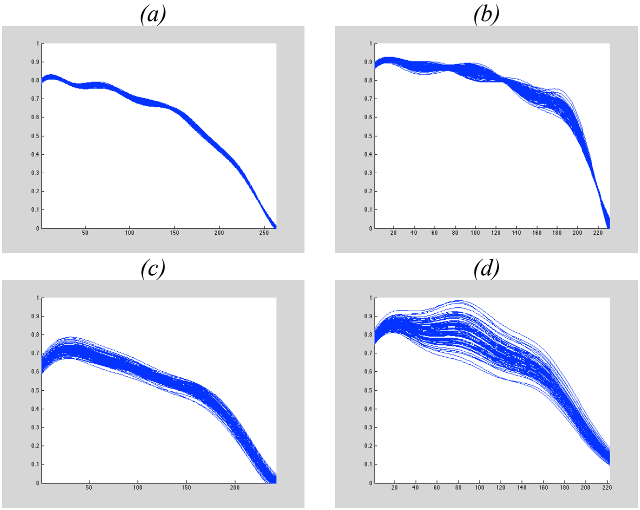


Figure 11: Several health curves for event 74 (a), 26 (b), 41 (c), 33 (d).

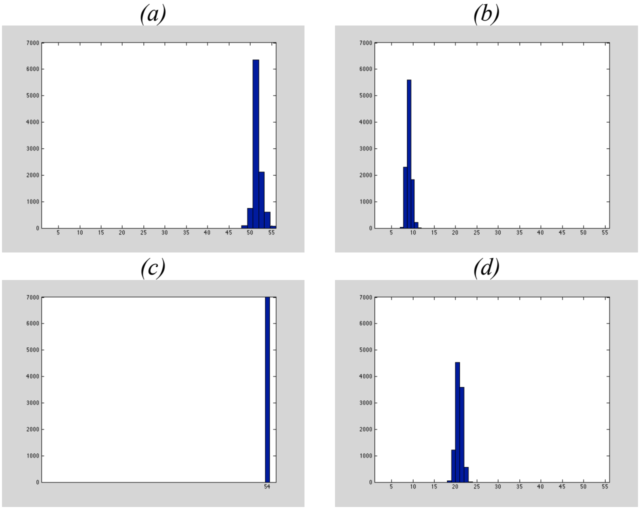


Figure 12: Histograms of RUL_k^u for the event 74 (a), 26 (b), 41 (c), 33 (d).

6. Conclusion

In this paper we proposed an approach taking advantage of all available knowledge at a fleet level for predictive diagnosis and prognosis. The originality of this work lies in the ability to make prognosis even if the degradation occurring is ill-defined, i.e. only partial knowledge about the degradation is available. Obviously, such a tool is clearly not self-sufficient. It is meant to work with experts whose knowledge helps to focus on how to solve a situation.

Despite the presented case study belongs to the naval domain, the proposed approach is general and can be applied to fields such as wind turbine farms, vehicle fleets… The presented work is a first step in that direction. Further steps can improve the usefulness of the proposed approach in several directions such as the computation of the units health which has to be homogenous over the entire fleet. An investigation on the operators to be used could be performed. The computation of the histograms of the RUL^u could be performed at every stage of the cause-tree.

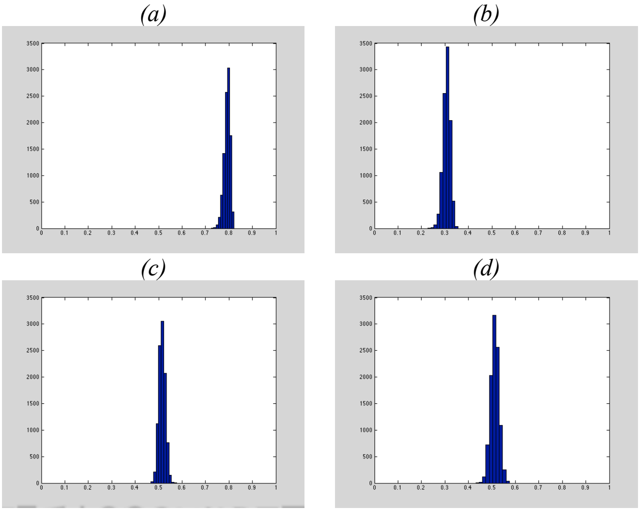


Figure 13: Histograms of W_k^u for the event 74 (a), 26 (b), 41 (c), 33 (d).

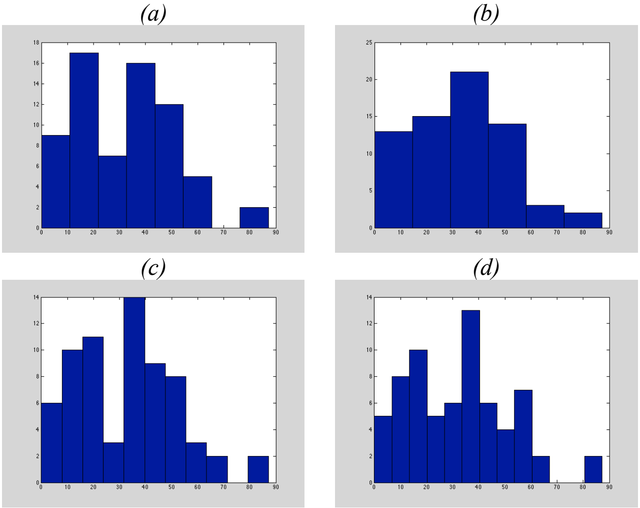


Figure 14: Histogram of the RUL^u of the Volvo Penta D16 MG diesel engine with ill-defined running degradation with different numbers of class: 8 (a), 6 (b), 11 (c) and 13 (d).

References

Bonissone, P.P., Varma, A. (2005). Predicting the Best Unit within a Fleet: Prognostic Capabilities Enabled by Peer Learning, Fuzzy Similarity, and Evolutionary Design Process. In Proceedings of the 14th IEEE International Conference on Fuzzy Systems, IEEE, pp. 312-318.

Byington C.S., M.J. Roemer, G.J. Kacprzynski et T. Galie (2002). Prognostic Enhancements to Diagnostic Systems for Improved Condition-Based Maintenance. 2002 IEEE Aerospace Conference, Big Sky, USA.

G.E.P. Box, G.M. Jenkins (1976), and Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco.

Heng A., S. Zhang, A.C.C. Tan et J. Mathew. (2009). Rotating machinery prognostic: State of the art, challenges and opportunities, Mechanical Systems and Signal Processing, vol. 23 (3), pp. 724-739.

Jardine A.K.S., D. Lin et D. Banjevic (2006). A review on machinery diagnostics and prognostic implementing condition-based maintenance. Mechanical Systems and Signal Processing, vol. 20, pp.1483-1510.

Liu J., Djurdjanovic D., Ni J., Casoetto N., Lee J. (2007) Similarity based method for manufacturing process performance prediction and diagnosis, Computers in Industry, 58, 558–566

Medina-Oliva G., Léger J-B, Voisin A., Monnin M., (2012) Predictive Diagnostic based on a Fleet-wide Ontology Approach, MFPT 2013, 13-17 May 2013 - Cleveland, Ohio, USA.

Medina-Oliva G., Weber P., Levrat E., Iung B. (2012a) Using probabilistic relational models for knowledge representation of production systems: A new approach to assessing maintenance strategies. CIRP Annals - Manufacturing Technology. in press. DOI: 10.1016/j.cirp.2012.03.059

Medina-Oliva G., Voisin A., Monnin M., Peysson F., Leger JB. (2012b). Prognostic Assessment Using Fleet-wide Ontology. PHM Conference 2012, Minneapolis, Minnesota, USA.

Medina-Oliva G., Peysson F., Voisin A., Monnin M., Léger J-B. (2013). Ships and marine diesel engines fleet-wide predictive diagnostic based on ontology, improvement feedback loop and continuous analytics. Proceedings of 25th International Congress on Condition Monitoring and Diagnostics Engineering Management, 11–13 June, 2013, Helsinki, Finland.

Monnin M., Voisin A., Leger JB., Iung B. (2011a). Fleet-wide health management architecture. Annual Conference of the Prognostic and Health Management Society. Montreal, Quebec, Canada.

Monnin M., Abichou B., Voisin A., Mozzati C. (2011b). Fleet historical cases for predictive maintenance. The International Conference Surveillance 6. October 25-26. Compiègne, France.

Muller A., Suhner M-C., Iung B. (2008). Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system. Reliability Engineering and System Safety. 93(2) 234-253.

Patrick, R., Smith, M J., Byington, C S., Vachtsevanos, G J., Tom, K., Ly, C. (2010). Integrated Software Platform for Fleet Data Analysis, Enhanced Diagnostics, and Safe Transition to Prognostic for Helicopter Component CBM, in Proceedings of Annual Conference of the Prognostic and Health Management Society, October 10-16, Portland, Oregon.

Reymonet, A., Thomas, J., Aussenac-Gilles, N. (2009). Ontology Based Information Retrieval: an application to automotive diagnosis, in Proceedings of International Workshop on Principles of Diagnosis, June 14-17, Stockholm, Sweden, pp. 9-14.

Rizzolo L., Abichou B., Voisin A., Kosayyer N. (2011), Aggregation of Health Assessment Indicators of Industrial Systems. In Proceedings of the 7th conference of the European Society for Fuzzy Logic and Technology, EUSFLAT-2011, Aix-Les-Bains, France.

Tipping M.E. (2001) Sparse Bayesian Learning and the Relevance Vector Machine, J ournal of Machine Learning Research 1, 211 244

Verma, A. K. and Srividya, A. and Ramesh, P. (2010). A systemic approach to integrated E-maintenance of large engineering plants, International Journal of Automation and Computing, vol. 7, pp. 173-179.

Wang P., Youn B., Hu C. (2012) A generic probabilistic framework for structural health prognostic and uncertainty management. Mechanical Systems and Signal Processing. 28, Pages 622–637

Wang T., Yu J., Siegel D., Lee J. (2008). A similarity-based prognostic approach for Remaining Useful Life estimation of engineered systems. International Conference on Prognostic and Health Management. Denver, USA.

Weber P., Medina-Oliva G., Simon C., Iung B. (2012). Overview on Bayesian networks Applications for Dependability, Risk Analysis and Maintenance areas. Engineering Applications of Artificial Intelligence, vol. 25 (4), (671-682).

.

Prognostics Health Management for Advanced Small Modular Reactor Passive Components

Ryan M. Meyer[1], Pradeep Ramuhalli[2], Jamie B. Coble[3], Mark R. Mitchell[4], David W. Wootan[5], Evelyn H. Hirt[6], Eric J. Berglin[7], Leonard J. Bond[8] and Chuck H. Henager, Jr.[9]

[1,2,4,5,6,7,9]*Pacific Northwest National Laboratory, Richland, WA, 99354, USA*

Ryan.Meyer@pnnl.gov
Pradeep.Ramuhalli@pnnl.gov
M.R.Mitchell@pnnl.gov
David.Wootan@pnnl.gov
Evelyn.Hirt@pnnl.gov
Eric.Berglin@pnnl.gov
Chuck.Henager@pnnl.gov

[3]*University of Tennessee, Knoxville, Knoxville, TN, 37996, USA*

jcoble1@utk.edu

[8]*Iowa State University, Ames, IA, 50011, USA*

bondlj@iastate.edu

ABSTRACT

In the United States, sustainable nuclear power to promote energy security is a key national energy priority. Advanced small modular reactors (AdvSMR), which are based on modularization of advanced reactor concepts using non-light-water reactor (LWR) coolants such as liquid metal, helium, or molten salt, may provide a longer-term alternative to more conventional LWR-based concepts. The economics of AdvSMRs will be impacted by the reduced economy-of-scale savings when compared to traditional LWRs and the controllable day-to-day costs of AdvSMRs are expected to be dominated by operations and maintenance costs. Therefore, achieving the full benefits of AdvSMR deployment requires a new paradigm for plant design and management. In this context, prognostic health management of passive components in AdvSMRs can play a key role in enabling the economic deployment of AdvSMRs. This paper discusses features of AdvSMR systems that are likely to influence PHM implementation for passive components and discusses some requirements based on those features. Further, a brief overview of the state-of-the-art in PHM relevant to AdvSMR passive components is provided followed by an illustration of prognostics for passive AdvSMR components.

1. INTRODUCTION

Nuclear energy currently contributes approximately 20% of baseload electrical needs in the United States and is considered a reliable generation source to meet future electricity needs. Sustainable nuclear power to promote energy security is a key national energy priority. The development of deployable small modular reactors (SMRs) is expected to support this priority by diversifying the available nuclear power alternatives for the country, and enhance U.S. economic competitiveness by ensuring a domestic capability to supply demonstrated reactor technology to a growing global market for clean and affordable energy sources.

Several concepts for SMRs have been proposed (Abu-Khader, 2009; Ingersol, 2009) with integral pressurized water reactor (iPWR) concepts the current front-runner for near-term licensing and deployment. Advanced small modular reactors (AdvSMRs), which are based on modularization of advanced reactor concepts using non-light-water reactor (LWR) coolants such as liquid metal, helium, or liquid salt may provide a longer-term alternative to LWRs and iPWRs.

The economics of small reactors (including AdvSMRs) will be impacted by the reduced economy-of-scale savings when compared to traditional LWRs, although the modular nature of such reactors can be advantageous in presenting lower initial capital costs. In addition, the controllable day-to-day

Meyer et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

costs of AdvSMRs are expected to be dominated by operations and maintenance (O&M) costs, and achieving the full benefits of AdvSMR deployment requires a new paradigm for plant design and management.

Components in nuclear power plants can be classified as active or passive. Passive components refer to those structures or components in a nuclear power plant that are functional without a power source. Examples of passive components include pipes, vessels, tanks, cables, etc. This is in contrast with active components which include pumps, valves, motors, etc. While proper maintenance of both active and passive components is important in the operation of nuclear power plants, the degradation in passive components, in particular, if not addressed in a timely fashion, is likely to result in unplanned plant shutdowns. Thus, PHM of passive components in AdvSMRs can play a key role in enabling the economic deployment of AdvSMRs.

A recent technical report describes several of the requirements for performing PHM of passive AdvSMR requirements and outlines several research gaps and technical needs to address these gaps (Meyer, Coble, Hirt, Ramuhalli, Mitchell, Wootan, Berglin, Bond, & Henager, 2013). This paper discusses features of AdvSMR systems that are likely to influence PHM implementation (Section 2) for passive components and discusses some requirements based on those features (Section 3). Further, a brief overview of the state-of-the-art in PHM relevant to AdvSMR passive components is provided (Section 4) followed by an illustration of prognostics for passive AdvSMR components (Section 5). Finally, some brief discussions and concluding remarks are provided in Section 6.

2. Advanced Small Modular Reactors

The evolution of nuclear power generating technology is organized by categorizing systems as Generation (Gen) I, II, III, III+, and IV technologies. Gen I includes the earliest prototype reactors while most commercial LWRs in operation today are considered Gen II technologies. Gen III and III+ reactors represent improvements over Gen II technologies with respect to increased reliance on passive safety mechanisms, increased use of digital instrumentation and control, and increased monitoring instrumentation. Gen IV represents a more significant leap in terms of technology advancements and concepts within Gen IV have expected deployments dates beyond 2030. The Gen IV International Forum (GIF) was created to help focus international resources and efforts to establish the feasibility and performance of future generation reactors. Improvements in safety and reliability, sustainability, proliferation resistance, and economics are among the key goals of the GIF efforts. AdvSMRs will be based on Gen IV concepts, such as those promoted by the GIF. Candidate technologies promoted by the GIF include (NERAC, 2002; Abram & Ion, 2008):

- Sodium Fast Reactors (SFRs)
- Very High Temperature Reactors (VHTRs)
- Gas-Cooled Fast Reactors (GFRs)
- Lead-Cooled Fast Reactors (LFRs)
- Molten Salt Reactors (MSRs)
- Supercritical Water-Cooled Reactors (SWCRs)

Like all nuclear reactors, heat is removed from the core in Gen IV reactors by a reactor coolant system that transfers the heat to a system of heat exchangers for power conversion. A depiction of a SFR in Figure 1 serves to illustrate many of the components that are basic to many nuclear power systems. In the case of the SFR, the primary sodium coolant and reactor core are contained within a reactor vessel. Penetrations in the reactor vessel allow the insertion and removal of control rods to manage the fission chain reaction. Pumps circulate the sodium through the reactor core and a secondary sodium loop transfers heat from a heat exchanger located in the reactor vessel to the steam generator. In the steam generator, heat is transferred from the sodium to water which is converted to steam. The steam is then converted to electricity through the turbine generator system.

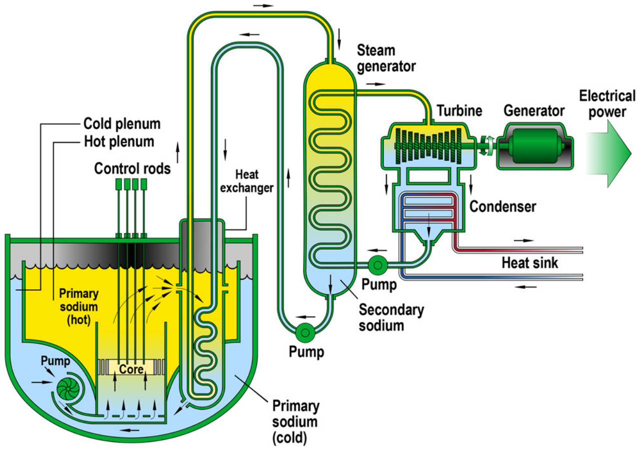


Figure 1. Depiction of a pool-type Sodium Fast Reactor.

There are many possible variations on the system discussed above for Gen IV technologies, including loop versus pool type designs for the primary systems or the elimination of the secondary heat exchange loop. In the case of gas-cooled reactor systems, it may even be possible to couple the primary coolant (i.e., He) directly to the gas turbine. In essence, the higher operating temperatures and exotic coolants of Gen IV systems enable many system configurations that cannot be realized with conventional technologies to achieve improved efficiencies. The following subsections briefly summarize features that will be generally applicable to AdvSMR systems and how these features will impact PHM system deployment for passive components.

2.1. Operating Environment and Materials Degradation

Passive components in AdvSMRs will be subject to relatively harsh operating environments in comparison to LWRs. This includes higher temperatures, fast neutron fluxes, and corrosive coolant conditions. Materials for advanced nuclear reactor applications generally consider radiation damage resistance, environmental stability, and high-temperature capability as paramount (Yvon & Carre, 2009; Zinkle & Busby, 2009). Volumetric swelling and dimensional stability, embrittlement, stress corrosion cracking, irradiation and thermal creep, and corrosion are critical materials degradation issues. Welds are problematic in nuclear structures as preferred sites for environmental degradation and stress-assisted degradation processes. Compatibility issues arise with regard to liquid metal coolants for liquid metal fast reactors (LFRs and SFRs) when metals and alloys in flowing coolant experience unwanted chemical reactions or leaching. In addition to driving the degradation issues, the harsh operating environment will negatively impact the performance of sensors for health monitoring and constrain their deployment.

2.2. Operations and Maintenance

Staffing and control room requirements have been identified as a significant technical and policy issue for multi-module SMR installations (Cetiner, Fugate, Kisner, & Wood, 2012). Key issues include determining appropriate staffing levels and how many units may be operated from a single control room. PHM systems can play an important role in reducing O&M costs and staffing needs by providing greater awareness of component and system conditions. In this case, to mitigate impending failure of a critical passive component of one module, the power level of that module may be decreased to reduce stresses and slow down the failure mechanisms. The power level of other modules may also be increased to compensate for the decrease in power to the first module. In this case, the role of a PHM system may be to determine appropriate stressor levels to achieve a desired remaining useful life (RUL). Also, compensation introduces coupling between modules and uncertainty that needs to be considered in the PHM implementation.

2.3. Concepts of Operation

In order to balance overall electricity generation and to meet fluctuating electrical demands, AdvSMRs may operate in a load-following mode, where the output of one or more reactor modules is adjusted (and thereby the electrical output of the plant). This type of operation has been studied for iPWR reactor designs (Hines, Upadhyaya, Doster, Edwards, Lewis, Turinsky, & Coble, 2011). Alternatively, electricity generation can be adjusted by using surplus heat for a secondary application. AdvSMRs may be required to operate in tandem with variable sources of renewable energy and/or supply electricity and process heat for industrial applications. One of the objectives of the Next Generation Nuclear Plant (NGNP) was to demonstrate cogeneration of electricity and hydrogen using high-temperature process heat (Southworth, MacDonald, Harrell, Shaber, Park, Holbrook, & Petti, 2003). Concepts for large-scale nuclear geothermal energy storage, shale oil extraction via nuclear and renewable energy, and symbiotic nuclear and renewable energy systems for electricity generation and hydrogen production have also been proposed (Haratyk & Forsberg, 2011; Forsberg, 2012; Forsberg, Lee, Kulhanek, & Driscoll, 2012). A key characteristic of many of these concepts is that they facilitate matching a constant nuclear energy source with variable electricity demand by distributing the nuclear production over multiple product streams (see Figure 2). In such scenarios, the distribution of load over components in the product streams will be subject to daily and seasonal load variations. Similar to the O&M, this introduces coupling and uncertainties that need to be considered in the PHM implementation.

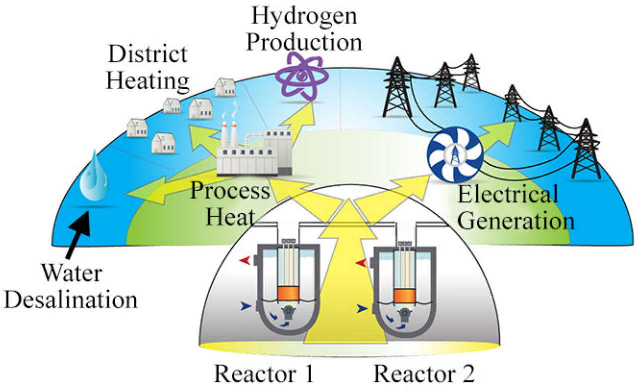


Figure 2. AdvSMR deployment concept illustrating multiple generation missions.

2.4. Refueling Schedules

Several advanced reactor concepts are intended to operate for extended periods between outages. For LWRs, outages are scheduled every 18–24 months for refueling but several advanced reactor concepts are intended to operate with much longer periods between refueling. The Toshiba 4S concept, for instance, is designed to operate up to 30 years without refueling (Tsuboi, Arie, Ueda, Grenci, & Yacout, 2012). The SSTAR is another advanced reactor concept with targeted operation periods of 15 to 30 years between refueling activities (Smith, Halsey, Brown, Sienicki, Moisseytsev, & Wade, 2008). Several other reactor concepts such as the liquid fuel MSRs and pebble bed-type VHTRs may have the capability to refuel while operating. Thus, it will be important that PHM systems for AdvSMRs are capable of utilizing data obtained from on-line measurements as well as data collected during outages.

3. Prognostic Health Monitoring Requirements

Based on AdvSMR features such as those discussed in Section 2, a requirements analysis for the application of PHM to AdvSMRs has been performed, identifying several important requirements to date (Meyer et al., 2013):

3.1. Sensors and Instrumentation for Condition Assessment of Passive Components

Because opportunities to perform inspections and maintenance of passive components when the plant is off-line will be limited in many designs, there is a need to monitor risk-significant passive components during plant operation for degradation. In addition, there is a need to monitor the stressors (time at temperature, fluence, mechanical loads, etc.) that are expected to contribute to degradation of these components. Requirements for sensors and instrumentation (whether for on-line or off-line condition assessment or for stressor monitoring) include:

- Ability to tolerate the harsh operating conditions in AdvSMRs.
- High sensitivity, to ensure that reliable measurements from earlier stages of degradation are possible.
- Capability to quantify the amount of degradation from the measurements.

3.2. Fusion of Measurement Data from Diverse Sources

Accessibility to some AdvSMR components may be restricted, particularly in pool-type reactors in which many of the primary system components will be submersed in coolant. Additionally, for concepts with infrequent refueling outages, opportunities to access components for periodic off-line inspection will be reduced. The fusion of data obtained from both online and offline measurements may enhance the performance of prognostics relative to relying on either type of measurement alone.

3.3. Address Coupling Between Components or Systems, and Across Modules

Compensating O&M strategies and concepts of operation that seek to distribute the output over multiple product streams will result in coupling effects between components, systems, and modules. This is likely to result in changing or time-varying load conditions that will introduce uncertainty in future stressor profiles.

3.4. Incorporation of Lifecycle Prognostics

An effective PHM system for AdvSMRs should be able to adapt or adjust its prognostics methodology to where the component or degradation is in its lifecycle. This helps to ensure accurate and timely determination of RUL based on the available information. Part of this requirement is determining the appropriate degradation models and updating these models in response to changes in operating conditions. Further, it will be necessary to transition between stressor-based prognostics and condition-based prognostics depending on the available data.

3.5. Integration with Risk Monitors for Real-time Risk Assessment

Given that it will likely be impractical to monitor or assess every component, a risk assessment will need to be performed to determine risk-significant components to ensure the highest return on investment. Such a risk assessment is in line with current practice for safety-significant components using risk-informed in-service inspection (RI-ISI). Also, the PHM system will be required to feed-back information on component condition and estimated RUL to the plant supervisory control algorithm for decision-making on O&M to manage and mitigate the impact of detected degradation. This feedback will have to flow through real-time risk monitors (Coble et al., 2013) that assess the risk associated with continued operation using the degraded component and contrast it with other options such as reactor-runbacks and shifting loads to other modules.

3.6. Interface with Plant Supervisory Control System

As already discussed, with compensating O&M strategies in a modular plant the potential exists to shift the power-generating burden among the units and/or modules to ensure component availability until the next scheduled maintenance opportunity. To accomplish this, PHM systems for passive components will require interfacing with the plant supervisory control system for AdvSMRs, to both obtain real-time information on operating conditions as well as feedback information that the control systems may use to adjust operating conditions to ensure a certain RUL.

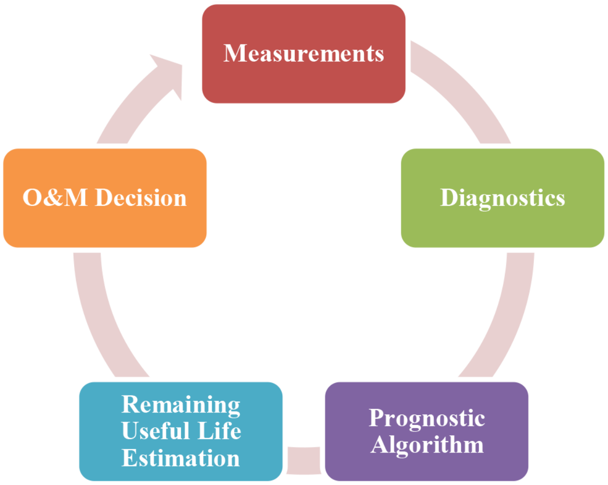


Figure 3. Depiction of the multiple components of a PHM system for passive AdvSMR components.

4. Relevant PHM State-Of-The-Art Overview

A PHM system of AdvSMR passive components will consist of several elements, as depicted in Figure 3. This section contains a brief overview of the state-of-the-art for PHM relevant passive AdvSMR components by considering these elements. The overview provided here is an abbreviated version of a state-of-the-art assessment provided in Meyer et al. (2013).

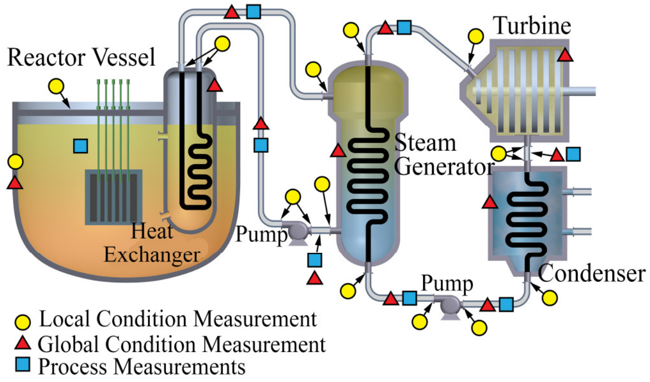


Figure 4. Conceptualization of candidate measurements and sensor locations for monitoring passive component degradation in AdvSMRs.

4.1. Measurements

Many different types of measurements can potentially be implemented in AdvSMRs to sample degradation and to input into prognostic models. Measurements can be categorized as local condition, global condition, and process/environmental measurements. Figure 4 illustrates several candidate measurements and sensor locations for monitoring passive component degradation in AdvSMRs. Local condition measurements refer to local nondestructive examination (NDE) measurements typically including various ultrasonic, eddy current, and visual testing techniques. These NDE measurements are currently limited to being performed while the reactor is off-line due to the operating environment. Although this limits the frequency at which these measurements can be performed, NDE measurements are generally more direct and descriptive than global condition or process/environmental measurements.

Global condition monitoring has also been deployed to monitor the status of passive components in nuclear reactors. As the name implies, these measurements relate to the overall health of a component or system and do not necessarily contain information about the nature of the fault or its precise location. Global condition measurements are sensitive to fairly advanced degradation such as cracks or the existence of loose parts. Although the measurements are less descriptive than local NDE measurements, global condition measurements are performed during reactor operation, and thus can be performed with greater frequency. In addition, global condition measurements can be used to monitor components that are not accessible to local NDE measurements due to physical access limitations. Examples of global condition monitoring methods in nuclear reactors include vibration analysis, neutron noise analysis, and acoustic emission. Guided ultrasonic wave techniques are also emerging in the nuclear power industry and have the potential to merge some of the benefits of global measurements (i.e., long range sampling) and local measurements (i.e., descriptiveness).

In addition to condition measurements, passive component health may indirectly be inferred from process/environmental measurements. These typically include measurements of temperature, flow rate, pressure, neutron flux, and coolant chemistry variables. Process/environmental conditions can be both contributors to passive component degradation and indicators of passive component degradation. In the former case, they represent stressors, and in the latter case, they are condition indicators. Like global condition measurements, process/environmental measurements are generally less descriptive or direct than local NDE measurements, but they are performed during reactor operation and can be performed with greater frequency.

4.2. Measurements in Harsh Environments

Multiple concepts exist for performing process/environmental and NDE measurements on-line at high temperatures and research in these technologies is ongoing. Examples of such efforts are provided by Ball, Holcomb, and Cetiner (2012) for measurements of temperature and neutron flux including gold-platinum (Au-Pt) thermocouples, Johnson Noise Thermometers (JNT), and high temperature fission chambers. In addition, there are several fiber optic and ultrasound based concepts for measuring temperature and pressure parameters. On the NDE side, there are efforts to develop piezoelectric based technologies for applications in SFRs (Bond, Griffin, Posakony, Harris, & Baldwin, 2012) and LFRs by Kažys, Voleisis, and Voleisiene (2008). A significant issue includes understanding how many proposed sensor types will hold-up to significant radiation fluxes and research efforts to address this gap with respect to in-pile instrumentation applications is ongoing (Rempe et al. 2011).

4.3. Diagnostics and Prognostics

Several approaches to diagnostics and prognostics are potentially available. Research towards addressing issues such as data fusion for diagnostics, prognostic models, lifecycle prognostics, uncertainty quantification, and prognostics in coupled systems, is ongoing. It is likely that research in these areas will require adaptation to address issues specific to AdvSMR passive component applications. With respect to data fusion for diagnostics, most efforts have focused on the fusion being performed at the signal level, using similar forms of measurements with less effort being expended on fusing dissimilar forms. Techniques for the latter efforts are largely data-driven and require data sets from known sources to determine the parameters of the fusion algorithm. Fusion using physics-based models, although not as widespread, has also been investigated.

Several state prediction techniques exist for potential application to passive components in AdvSMRs, many of them based on data-driven or probabilistic models of damage progression. Physics-of-failure models are increasingly being considered. Limited failure rate data or information related to many passive components in AdvSMRs will motivate the use of physics-of-failure models over historical data-driven models. Applicable models exist for many forms of relevant degradation such as Paris' Law for fatigue and Norton's Law for thermal creep. These models contain empirically derived constants that may not be fully known over the range of relevant operating conditions in AdvSMRs. Tracking algorithms (i.e., Kalman filtering, extended Kalman filtering, and particle filtering) provide a convenient framework for incorporating the latest information from measurements and facilitating the propagation of uncertainty to failure. Coupling the particle filter technique with physics-of-failure models for degradation modes can provide a versatile means for estimating the RUL of AdvSMR passive components.

5. Illustration–Prognostics for AdvSMR Passives

The PF technique is adequately described in the literature, including several tutorials for implementation (Arulampalam, Maskell, Gordon, & Clapp, 2002; An, Choi, & Kim, 2012). An application of PF to forecast mechanical fatigue degradation in passive components in LWRs is described by Ramuhalli, Bond, Griffin, Dixit, and Henager Jr. (2010). Here, we provide a simple illustration of the PF technique to predict the failure of AdvSMR components due to thermal creep. Additional functionality and complexity can then be demonstrated by stepwise expansions and modifications to this simple illustration.

The forecasting of thermal creep damage in He gas turbine blades fabricated from a Ni-based superalloy has recently been investigated by Baraldi, Mangili, and Zio (2012) using an ensemble of empirical models to improve performance. Here, Norton's Law is used with the PF technique to predict the RUL of AdvSMR passive components. To generate a sequence of states, Norton's Law [eq. (1)], is written as a state transition model:

$$\varepsilon_{k+1} = A\sigma^n \left(t_{k+1} - t_k\right) + \varepsilon_k . \tag{1}$$

Norton's Law parameters for 316L stainless steel weld material provided in Nassour, Bose, and Spinelli (2001) are used for the initial demonstration presented here assuming a temperature of T = 700°C. For now, the Norton's Law parameters are assumed to be Gaussian distributed variables and the values from Nassour et al. (2001) are interpreted as mean values although other distributions for these variables can be accommodated. The values of these parameters are provided in Table 1, along with assumed standard deviations.

Norton's Law is also used to generate simulated NDE measurement data. In this case, the model is developed in anticipation of accelerated aging studies that will provide data to validate the model illustrated here and potentially other models. The measurement uncertainties are assumed to have a Gaussian distribution. In this case, the uncertainty in the NDE measurements is assumed to be 0.1% of creep strain and the failure criterion is 3% creep strain. The actual failure time for these conditions according to Norton's Law is 10.8 hrs. The NDE measurements are simulated to be performed with a periodicity of 1 hr. This selection was made to approximate the relative frequency that offline NDE measurements may be performed on an AdvSMR, assuming the failure time in the accelerated studies is correlated with a plant lifetime.

Failure projections are included in Figures 5 through 7, for NDE measurements performed at 0 and 1 hours; 0, 1, and 2 hours; and 0, 1, 2, 3, and 4 hours. The distributions of RUL for each scenario are shown in Figures 8 through 10. The results were generated using a sample of 5000 particles.

Table 1. Summary of parameters and variables used in Norton's Law model to forecast thermal creep failure.

Parameter	Value (mean)	Std. Dev.
n	9.05	3.33%
A	2.93×10^{-22} $(\text{N m}^{-2})^{-n}$ h^{-1}	10%
σ	125 MPa	---

6. Conclusions and Discussions

PHM for passive components in AdvSMRs can play a key role in facilitating the deployment of AdvSMRs by minimizing controllable day-to-day costs associated with plant O&M. Although potential concepts and designs for AdvSMRs vary significantly, there are some general features that can help define the requirements of a PHM system for passive components. Degradation may be sampled in AdvSMRs through online and offline measurements. A PHM system is likely to be most effective if prognostics algorithms can use both types of measurements.

A basic illustration is provided of a prognostics method based on the PF technique for predicting passive component failure due to thermal creep degradation. The illustration simulates sampling of creep degradation with offline NDE measurements. The illustration only represents the start of prognostic algorithm development as additional functionality to address many the requirements in Section 3 will need to be demonstrated. The approach is to alternately add functionality and demonstrate that added functionality with accelerated aging studies.

ACKNOWLEDGMENT

The work described in this report was sponsored by the Small Modular Reactor Research and Development Program of the U.S. Department of Energy (DOE) Office of Nuclear Energy and performed at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated by Battelle Memorial Institute for the U.S. DOE under Contract DE-AC05-76RL01830.

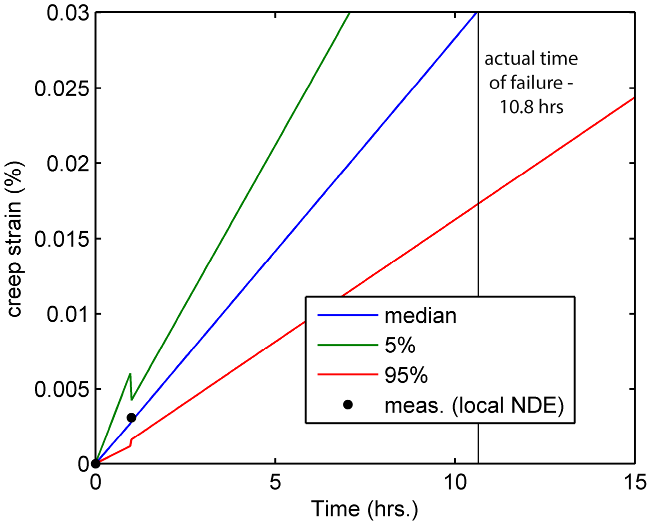


Figure 5. Failure projection for thermal creep based NDE measurements at 0 and 1 hours.

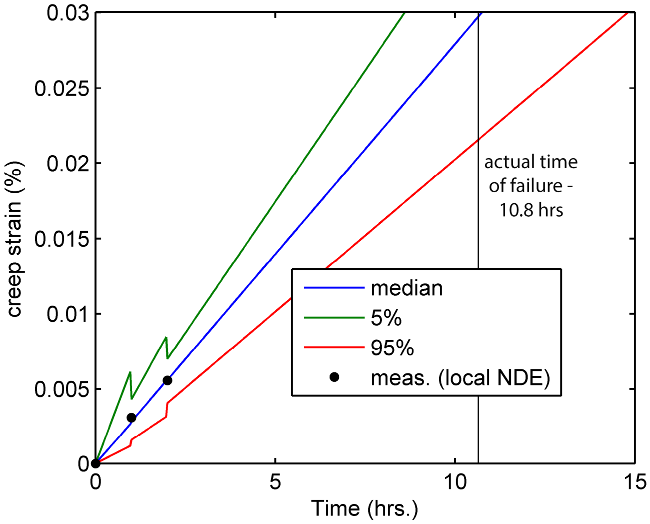


Figure 6. Failure projection for thermal creep-based NDE measurements at 0, 1, and 2 hours.

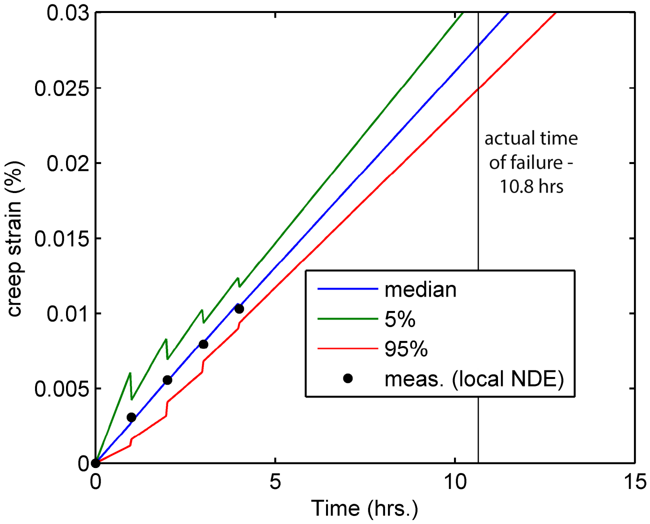


Figure 7. Failure projection for thermal creep-based NDE measurements at 0, 1, 2, 3, and 4 hours.

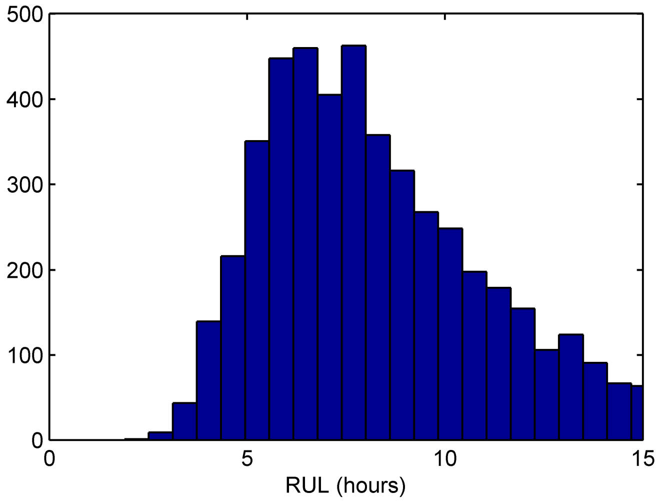


Figure 8. RUL distribution for NDE measurements performed at 0 and 1 hours (see Figure 6).

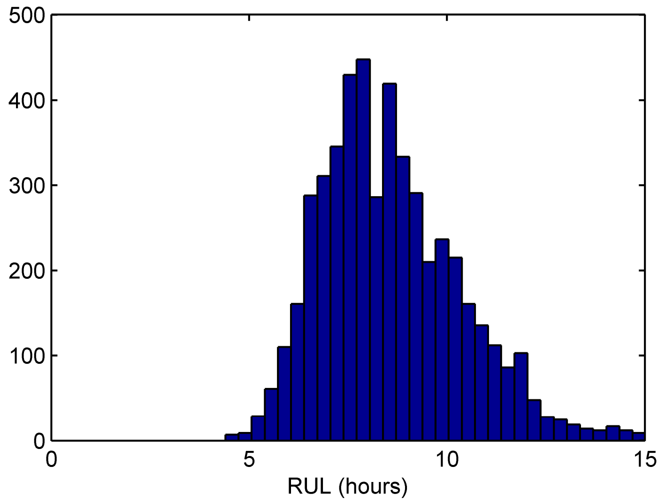


Figure 9. RUL distribution for NDE measurements performed at 0, 1, and 2 hours (see Figure 7).

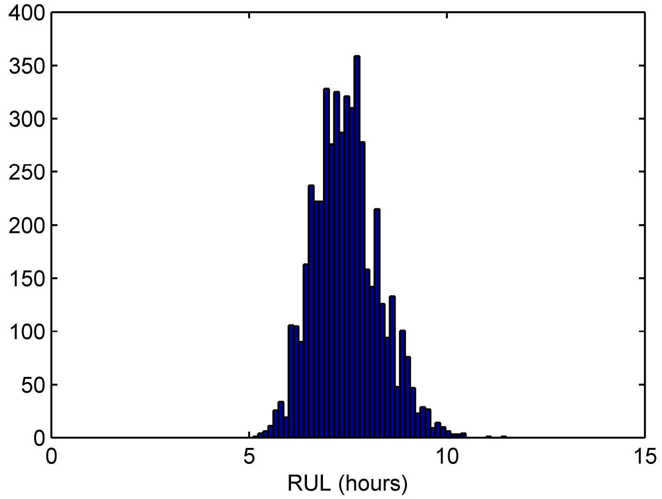


Figure 10. RUL distribution for NDE measurements performed at 0, 1, 2, 3, and 4 hours (see Figure 8).

REFERENCES

Abu-Khader, M. M. (2009). Recent advances in nuclear power: A review. *Progress in Nuclear Energy*, vol. 51, pp. 225-235. doi:10.1016/j.pnucene.2008.05.001.

Abram, T., & Ion, S. (2008). Generation-IV nuclear power: A review of the state of the science. *Energy Policy*, vol. 36 (12), pp. 4323-4330. doi:10.1016/j.enpol.2008. 09.059.

An, D., Choi, J.-H., & Kim, N. H. (2012). A tutorial for model-based prognostics algorithms based on Matlab code. *Annual Conference of Prognostics and Health Management Society 2012*, September 23-27, Minneapolis, MN. http://www.phmsociety.org/sites/ phmsociety.org/files/phm_submission/2012/phmc_12_ 122.pdf.

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, vol. 50 (2), pp. 174-188. doi:10.1109/78.978374.

Ball, S. J., Holcomb, D. E. & Cetiner, S. M. (2012). *HTGR measurements and instrumentation systems.* ORNL/TM-2012/107, Oak Ridge, TN: Oak Ridge National Laboratory.

Baraldi, P., Mangili, F., & Zio, E. (2012). Ensemble of bootstrapped models for the prediction of the remaining useful life of a creeping turbine blade. *IEEE International Conference on Prognostics and Health Management: Enhancing Safety, Efficiency, Availability, and Effectiveness of Systems Through PHM Technology and Application (PHM 2012)* (pp. 1-8), June 18-21, Denver, CO. doi: 10.1109/ICPHM.2012.6299506.

Bond, L. J., Griffin, J. W., Posakony, G. J., Harris, R. V. & Baldwin, D. L. (2012). Materials issues in high temperature ultrasonic transducers for under-sodium viewing. *Proceedings of 38th Annual Review of Progress in Quantitative Nondestructive Evaluation* (pp. 1617–1624), July 17-22, 2011, Burlington, VT. doi: 10.1063/1.4716407.

Cetiner, S. M., Fugate, D. L., Kisner, R. A., & Wood, R. T. (2012). *Functional requirements for supervisory control of advanced small modular reactors.* SMR/ICHMI/ORNL/TR-2013/03, Oak Ridge, TN: Oak Ridge National Laboratory.

Coble, J. B., Coles, G. A., Ramuhalli, P., Meyer, R. M., Berglin, E. J., Wootan, D. W. & Mitchell, M. R (2013). *Technical needs for enhancing risk monitors with equipment condition assessment for advanced small modular reactors.* PNNL-22377 Rev. 0; SMR/ICHMI/PNNL/TR-2013/02, Richland, WA: Pacific Northwest National Laboratory.

Forsberg, C. (2012). A nuclear wind/solar oil-shale system for variable electricity and liquid fuels production. *Proceedings of 2012 International Congress on Advances in National Power Plants (ICAPP '12)* (pp. 2758-2765), June 24-28, Chicago, IL.

Forsberg, C. W., Lee, Y., Kulhanek, M., & Driscoll, M. J. (2012). Gigawatt-year nuclear-geothermal energy storage for light-water and high-temperature reactors. *Proceedings of 2012 International Congress on Advances in National Power Plants (ICAPP '12)* (pp. 2728-2737), June 24-28, Chicago, IL.

Haratyk, G., & Forsberg, C. W. (2011). Nuclear-renewables energy system for hydrogen and electricity production. *Nuclear Technology*, vol. 178 (1), pp. 66-82 http://epubs.ans.org/download/?a=13548.

Hines, J. W., Upadhyaya, B. R., Doster, J. M., Edwards, R. M., Lewis, K. D., Turinsky, P., & Coble, J. (2011). *Advanced instrumentation and control methods for small and medium reactors with IRIS Demonstration.*

Report No. DE-FG07-07ID14895/UTNE/2011-3, Knoxville, TN: The University of Tennessee. doi:10.2172/1015813.

Ingersoll, D. T. (2009). Deliberately small reactors and the second nuclear era. *Progress in Nuclear Energy*, vol. 51 (4-5), pp. 589-603. doi:10.1016/j.pnucene. 2009.01.003.

Kazys, R., Voleisis, A. & Voleisiene, B. (2008). High temperature ultrasonic transducers: Review. *Ultragarsas* (*Ultrasound*), vol. 63 (2), pp. 7-17.

Meyer, R. M., Coble, J. B., Hirt, E. H., Ramuhalli, P., Mitchell, M. R., Wootan, D. W., Berglin, E. J., Bond, L. J., & Henager Jr., C. H. (2013). *Technical needs for prototypic prognostic technique demonstration for advanced small modular reactor passive components*. PNNL-22488 Rev. 0, SMR/ICHMI/PNNL/TR-2013/01, Richland, WA: Pacific Northwest National Laboratory.

Nassour, A., Bose, W. W., & Spinelli, D. (2001). Creep properties of austenitic stainless-steel weld metals. *Journal of Materials Engineering and Performance*, vol. 10 (6), pp. 693-698. doi:10.1361/105994901770344566.

NERAC. (2002). *A technology roadmap for generation iv nuclear energy systems - ten nations preparing today for tomorrow's energy needs*. GIF-002-00, Washington, DC: U.S. DOE Nuclear Energy Research Advisory Committee (NERAC) and the Generation IV International Forum (GIF). http://www.osti.gov/energycitations/servlets/purl/859029-304XRr/.

Ramuhalli, P., Bond, L. J., Griffin, J. W., Dixit, M., & Henager, C. H., Jr. (2010). A Bayesian prognostic algorithm for assessing remaining useful life of nuclear power components. *7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC&HMIT 2010)* (pp. 875-886, vol. 2), November 7-11, Las Vegas, NV.

Rempe, J. L., MacLean, H., Schley, R., Hurley, D., Daw, J., Taylor, S., Smith, J., Svoboda, J., Kotter, D., Knudson, D., Guers, M., Wilkins, S. C. & Bond, L. J. (2011). Strategy for developing new in-pile instrumentation to support fuel cycle research and development. INL/EXT-10-19149, Idaho Falls, ID: Idaho National Laboratory.

Smith, C. F., Halsey, W. G., Brown, N. W., Sienicki, J. J., Moisseytsev, A., & Wade, D. C. (2008). SSTAR: The US lead-cooled fast reactor (LFR). *Journal of Nuclear Materials*, vol. 376 (3), pp. 255-259. doi:10.1016/j.jnucmat.2008.02.049.

Southworth, F. H., MacDonald, P. E., Harrell, D. J., Shaber, E. L., Park, C. V., Holbrook, M. R., & Petti, D. A. (2003). The next generation nuclear plant (NGNP) project. *Global 2003: Atoms for Prosperity: Updating Eisenhowers Global Vision for Nuclear Energy* (pp. 276-287), November 16-20, New Orleans, LA. http://www.inl.gov/technicalpublications/Documents/2761729.pdf.

Tsuboi, Y., Arie, K., Ueda, N., Grenci, T., & Yacout, A. M. (2012). Design of the 4S reactor. *Nuclear Technology*, vol. 178 (2), pp. 201-217.

Yvon, P., & Carre, F. (2009). Structural materials challenges for advanced reactor systems. *Journal of Nuclear Materials*, vol. 385 (2), pp. 217-222. doi:10.1016/j.jnucmat.2008.11.026.

Zinkle, S. J., & Busby, J. T. (2009). Structural materials for fission & fusion energy. *Materials Today*, vol. 12 (11), pp. 12-19. doi:10.1016/s1369-7021(09)70294-9.

Detection of CH-53 swashplate bearing deformation - from a 3D dynamic model to diagnostics

Mor Battat[1], Gideon Kogan[1], Alex Kushnirsky[1], Renata Klein[2] and Jacob Bortman[1]

[1]*Pearlstone Center for Aeronautical Engineering Studies and Laboratory for Mechanical Health Monitoring, Department of Mechanical Engineering, Ben-Gurion University of the Negev, P.O. Box 653, Beer Sheva 84105, Israel*

morbat@post.bgu.ac.il
genak@bgu.ac.il
jacbort@bgu.ac.il

[2] *R.K. Diagnostics, P.O.Box 101, Gilon, D. N. Misgav 20103, Israel*

Renata.Klein@rkdiagnostics.co.il

ABSTRACT

The purpose of this paper is to present the research approach for the development of an algorithm for detection of a failure of the CH-53 swashplate bearing external spacer. The failure causes a lack of support of the swashplate bearing, thus creating a deformation of the outer ring. This study integrates the results of a new 3D dynamic model, developed for assessment of the defect pattern, and results from experiments. The research approach is planned in hierarchical phases. The experimental phases include a small scale specimen, full scale test rig, helicopter blades test facility and finally a CH-53 helicopter. The unique approach gradually simulates the real work environment of the swashplate bearing. The first two experimental phases and their results are presented. The first experimental phase is conducted on a small scale specimen and the second phase on a full scale test rig. Model results indicate that the lack of support has a defect pattern in both the radial and axial directions. These results are validated with the small scale specimen. In the future phases, the algorithm will be validated with data from the helicopter blades test facility and CH-53 helicopter.

1. INTRODUCTION

In 2008, the swashplate bearings of a CH-53 failed during a flight. The crew, subjected to a stroke followed by significant vibrations, managed to perform an emergency landing. Post-accident examination showed that a failure in the external spacer, separating the duplex ball bearing outer rings, led to collapse of the upper bearing and direct contact between the stationary and rotating swashplate.

The swashplate is a device used for transmitting control inputs to the blades (see Figure 1). Therefore the swashplate bearings are safety of flight critical components. The swashplate of the CH-53 is constructed of two angular contact ball bearings supported by two spacers- internal, between the static inner rings and external, between the rotating outer rings (see Figure 2).

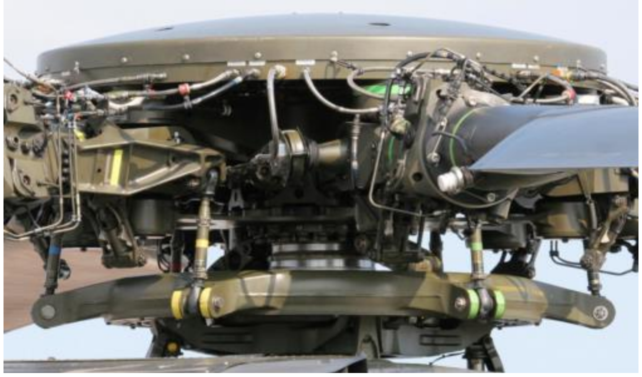


Figure 1. Sikorsky CH-53 swashplate; www.airliners.net

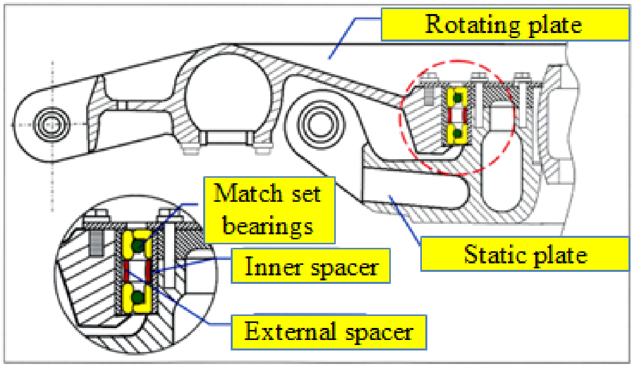


Figure 2. Swashplate cross section

Mor Battat et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The Laboratory for Mechanical Health Monitoring at Ben Gurion University of the Negev (BGU) is developing a method for early detection of failures caused by deformation of the bearing outer ring and is establishing prognostic tools that could prevent such catastrophic failures.

This article describes the methodology for development of a prognostic tool. The swashplate bearing cannot be efficiently examined before each flight, hence the need to develop a real-time diagnostic tool. The potential benefit of this research results is both in improving the aircraft safety and in reduction of maintenance costs. It should be noted, that the UH-60, as well as other Sikorsky helicopters, have a similar swashplate structure.

The post-accident investigation findings indicate local deflection of the external spacer, adjacent to the grease nipple, as the first stage of the failure. External spacer increased deformation, toward the inner spacer, resulted in abrasion of the two spacers, finally causing the external spacer fracture.

2. General approach

Pattern of bearings sub-components deformations are not thoroughly investigated, hence the need to study the defects signature. A new 3D dynamic model (Kogan et al., 2012) has been developed as a tool for investigation of faulty bearings behavior. As shown in Figure 3, time history data generated by the model was analyzed in order to define the fault expected pattern. Vibration signals generated by three test systems will be analyzed based on the model results in order to recognize the fault signature. At last, diagnostic algorithms will be applied to vibration data acquired from a CH-53 swashplate. The obtained signature will be compared to the validated dynamic model pattern to define the algorithm and the condition indicators of the fault.

Most bearings dynamic models known today simulate localized faults. Since the majority of used bearings are radial bearings a two dimensional model is sufficient for pattern and behavior simulations. The CH-53 accident examination presents a different case. The axial loading and the fact that the defect is at the supporting spacer rather than at the bearing components requires a new insight.

Constructing a new 3D model was the initial step. The model laid the foundations for the recognition of the defective pattern and the scale of vibrations at different levels of the failure progression. Knowing the expected pattern has a significant advantage when analyzing noisy data.

Comparing the dynamic model results with vibration signals generated by different test systems is planned in phases. It is assumed that when advancing from one phase to another the measured data will simulate more realistically the signal and the environment of the helicopter rotor head. As a result, the difficulty to recognize the pattern of the faulty bearing is expected to increase from one phase to another, demanding more complex algorithms. Progress in phases is performed to guarantee the recognition of the defect signature among the variety of signals generated by the helicopter during a flight.

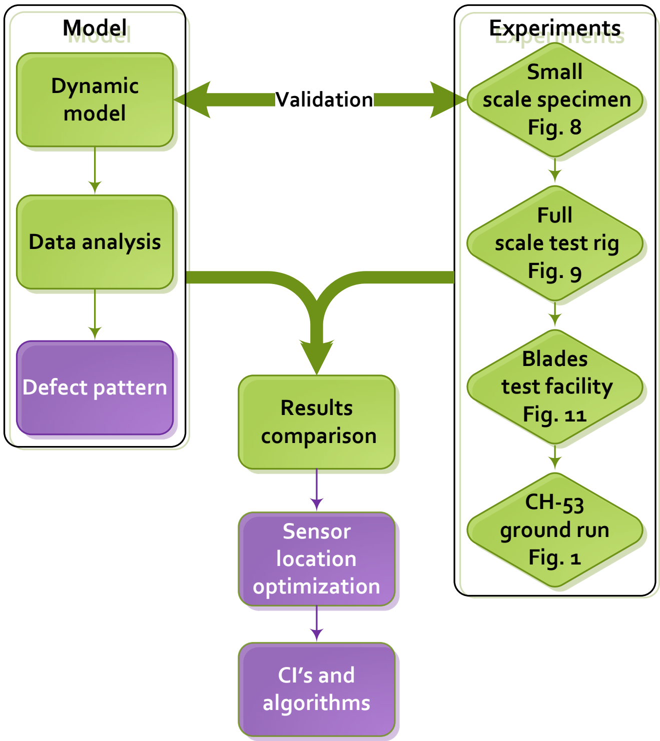


Figure 3. Research flow chart

Two designated test rigs were developed in the BGU lab. A small scale specimen is designed for fast experiments and easy simulation of a wide range of faults. A full scale test rig simulates better the real environment of the helicopter swashplate. The rig, constructed with an original swashplate, provides a good simulation under laboratory conditions of the real support structure. IAI helicopter blades test facility will be a following phase to the laboratory test rigs, simulating a full rotor head icluding transmission, main shaft and blades. The test systems will be installed and tested with a healthy swashplate at first. After all relevant data is acquired the healthy swashplate will be replaced by a faulty swashplate with a defective external spacer. In the final phase data acquisition will be made on an operational CH-53 ground run. The correlation of the signals between each test system is a primary question of the research and constantly being examined.

Research by Keller & Grabill (2005) examined the detection ability of a CH-47D faulted swashplate bearings by vibration monitoring. The tests were conducted on a swashplate test rig built in back-to-back design in which the upper swashplate assembly is the drive and the lower swashplate assembly houses the tested bearing. The work shows that spalled and corroded bearings are easily detected. Cage faults, however, were not detected by standard analysis methods.

A paper by Dempsey et al. (2010) compares faulty bearing data collected on a UH-60 Black Hawk helicopters and data collected on a test stand. Three Receiver Operating Characteristic (ROC) curves were used to compare the bearing performances. Variables affecting the ability to recognize a fault are presented. These variables include sensor type, mounting, location, signal processing, structural dynamics, flight regimes, and history of the component. It is concluded that development of diagnostic tools on a test stand is a complicated process due to environmental factors.

Our approach is designed to address the complexity of developing algorithms based on test rig data and implementing the algorithms on data from helicopter flights. The integration of a physical model and hierarchical test systems is a method designed to recognize the fault pattern in the different experimental systems.

Frequently, when analyzing data, appearance of irregular peaks indicates a fault, but one cannot associate the peaks due to lack of the fault pattern. The physical model is a tool aimed to address this uncertainty. The experiment systems are aimed to address the influence of the environmental factors.

3. Dynamic Model

A new 3D dynamic ball bearing model was developed to study the effect of anomalies in bearing sub-components on the bearing dynamic behavior. The aim of the model is to calculate the dynamic response of a bearing with a wide spectrum of faults. The algorithm was implemented numerically in MATLAB and is based on a model used in a study by Tkachuk and Strackeljan (2010).

The dynamics, for each bearings component, are based on the classical dynamic equations

$$\Sigma\overrightarrow{F_f} + \Sigma\overrightarrow{F_n} = m\vec{a},\ \Sigma\left(\vec{R}\times\overrightarrow{F_f}\right) = \boldsymbol{I}\dot{\vec{\omega}}_{xyz} + \vec{\Omega}\times(\boldsymbol{I}\vec{\omega}) \quad (1)$$

where F_f, F_n are respectively the friction and the normal forces that act on a body, with mass m and acceleration $\vec{a}$; and $\Sigma\left(\vec{R}\times\overrightarrow{F_f}\right)$ is the total moment of force acting on a body with a moment of inertia tensor $\boldsymbol{I}$, angular velocity $\vec{\omega}$; body system xyz, with angular velocity $\vec{\Omega}$; and rotational acceleration, within the body system, $\dot{\vec{\omega}}_{xyz}$.

The relative velocity equation

$$\overrightarrow{v_b} = \overrightarrow{v_a} + \vec{\omega}\times\overrightarrow{ab} \quad (2)$$

Where $\overrightarrow{v_x}$ is the velocity of the body at $\vec{x}$ and $\vec{\omega}$ is the angular velocity of $\overrightarrow{ab}$.

The presented equations describe the motion of all the modeled bodies and are solved using time steps (see Figure 4). In each time step, the solution of the equations is based on the previous time step solution, assuming a constant acceleration.

The dynamic model was validated by comparison to analytical solutions and known bearing response to local defects. After validation, the model was implemented for a structural deformation case that corresponds with the accident examination findings.

The failure investigation showed that the external spacer collapse separating the duplex ball bearing outer rings, led to damage of the upper bearing. The deformation of the outer ring was calculated using finite element analysis (see Figure 6). It was found that the lack of proper support causes local deflection in the axial direction of the outer ring of the top bearing. The deflected geometry was implemented in the dynamic model.

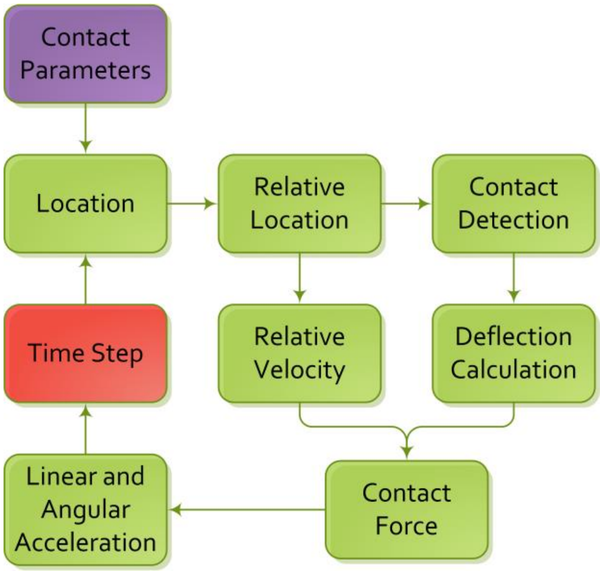


Figure 4. Simplified model algorithm

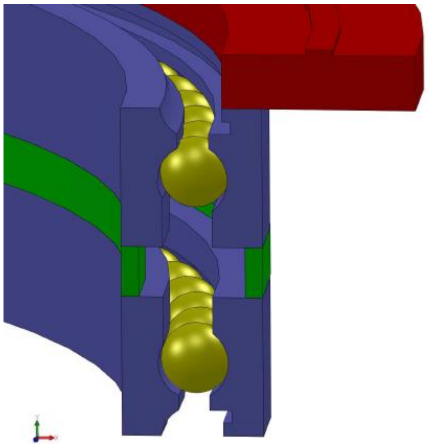


Figure 5. Model section

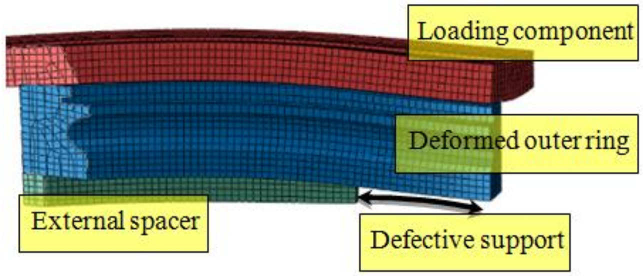


Figure 6. Structural defect finite element model

4. Small scale specimen

The small scale specimen is a kit of a generic test rig. The rig, shown in Figure 7, is a versatile platform built to study the signatures of common machinery faults. It has simply operated replaceable kits designed for fast experiments and simulation of a wide range of faults.

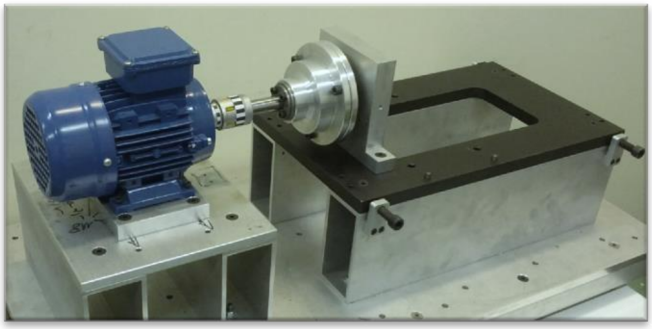
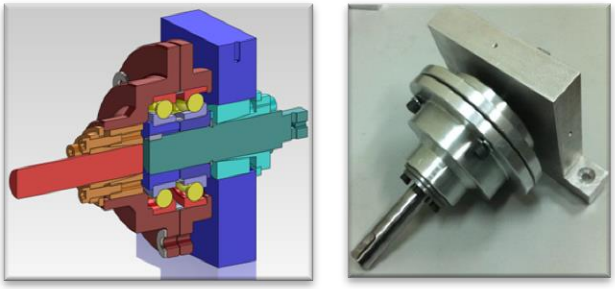


Figure 7. Generic test rig with the small scale specimen kit



a) cross-section b) image

Figure 8. Small scale specimen

The specimen, shown in Figure 8, is constructed of two axial unidirectional ball bearings (NSK 7208BWG) separated by the examined spacers (static internal spacer and dynamic external spacer). The bearings are tightly set into aluminum cups. Pressure, simulating the spacers axial load, is applied by tightening the cups screws, forcing the two outer races onto the spacer. Experiments included installation of the specimen with a healthy spacer and a number of spacers with faults of different sizes. Variation of tightening pressure resembled the spacers compression. A variable frequency drive controls the velocity of the electric motor and the shaft speed is measured by an optic encoder. Acceleration sensors can be mounted directly to the static shaft or onto the shaft housing, as shown in Figure 8a.

5. Full Scale Test Rig

The main purpose the full scale test rig was to simulate the original work environment of the swashplate bearings without the environmental noise. The reduction of noise will help in recognition and isolation of the searched CIs.

The rig, shown in Figure 9, is built around an original CH-53 swashplate. The rotating plate is set in motion by a transmission of gears and a belt driven by an electric motor. The motor is set to rotate the plate at 185 RPM, the CH-53 main rotor speed.

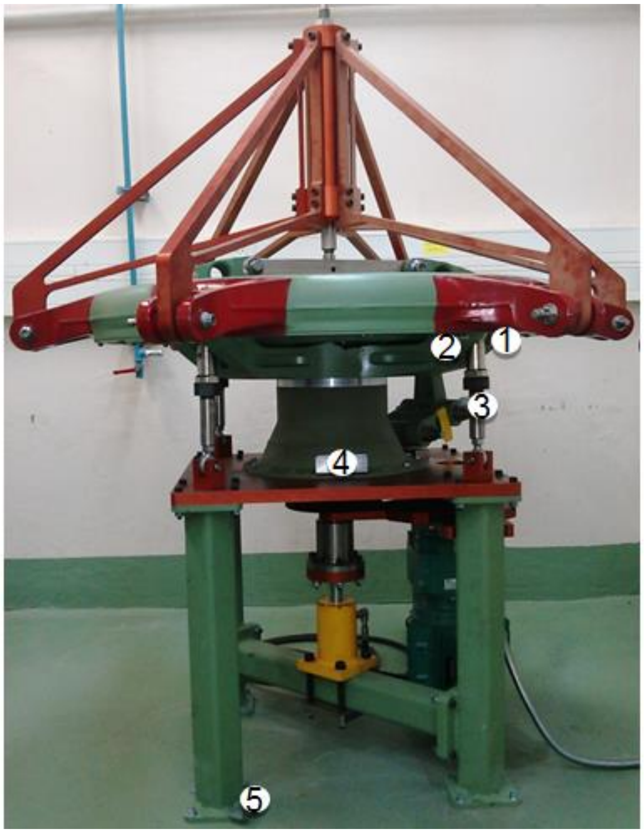


Figure 9. Full scale test rig and sensors locations

Simple and efficient change of swashplates was the main consideration in the design process. Already in-use loading configuration of such test rigs were examined (Bayoumi et al, 2008; Keller & Grabill, 2005). The chosen configuration, shown in Figure 9, has a reduced number of moving parts and lower weight due to reduction of the cyclic loading.

Figure 10 demonstrates the path of the load through the test rig. The simulation of the axial load on the swashplate is done using a hydraulic cylinder. The piston loads the main shaft that transmits the load to the rotating plate through the spider. The rotating plate transmits the load through the bearings to the stationary plate.

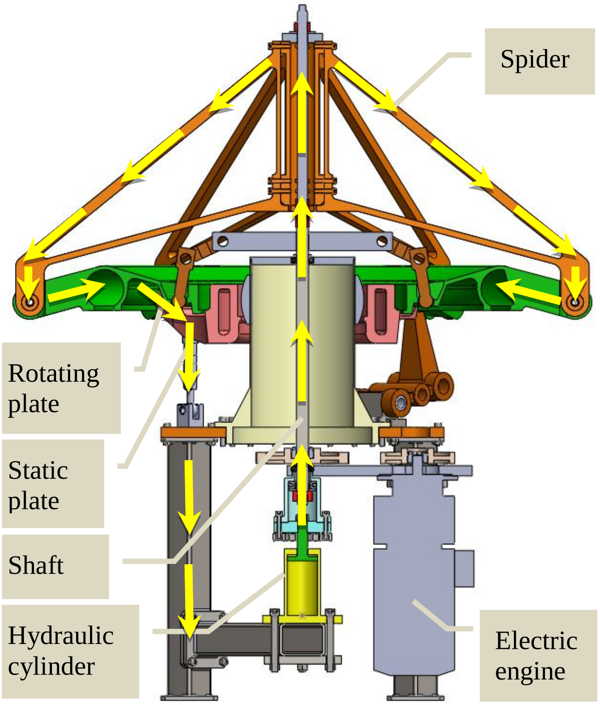


Figure 10. Load transformation over the test rig

In horizontal operating mode the rig simulates hovering due to the symmetric loading, therefore during the first installation, its balance was validated.

Figure 11. Helicopter blades test facility

The experimental program includes runs with healthy and faulty swashplates at a number of rotational speeds, piston pressures and sensor locations. Five different sensor locations, shown in Figure 9, have been selected considering both operational and scientific applications. The operational locations are derived from helicopter already-in-use sensor brackets and considering maintenance simplicity in case of malfunction. The scientific application considers sensor location optimization and variation of the damping in the rig. Furthermore, transfer function influences the acquired signal. Data from different sensor locations can be used as a tool of comparison or for further analysis.

Experiments at the helicopter blades test facility, shown in Figure 11, are planned to take place in the next phase. For facilitation of data comparison, the same acquisition system and sensors will be used.

6. RESULTS

6.1. Model results

This section describes the model implementation. The model was first designed with the small scale specimen bearing parameters followed by the full scale test rig bearing parameters.

6.1.1. Small scale specimen bearing implementation

The results of the standard bearing (NSK 7208BWG) dynamics under the influence of a defective support are presented. Table 1 presents the bearing parameters.

Table 1. Small scale specimen bearing parameters

Pitch diameter	$60.01 \cdot 10^{-3}[m]$
Ball diameter	$11.91 \cdot 10^{-3}[m]$
Number of balls	14
Outer rings rotation speed	$20[rps]$
Defect length (azimuth)	30°

During a ball passage through the outer ring deflected zone, the axial load acting on the ball drops and the ball support of the inner ring is reduced. In order to compensate for the support reduction, the neighbor balls are overloaded. The sudden interruption of the inner ring support causes a periodic deflection.

The inner ring center radial acceleration, shown in Figure 12, presents paired impulses that correspond to the impulses of the axial acceleration. However, the impulses that appear in the radial acceleration include shaft speed modulation, which is related to the defect location variation.

The angular distance between the balls is shorter than the angular length of the defect, therefore, between each two impacts corresponding to the interaction of a ball with the fault, appears a third impact that corresponds to next ball entrance into the deflected zone.

Examining the sensitivity of the bearing dynamics to variation of the defect geometry showed that the maximum acceleration of the inner ring is proportional to the maximum outer ring deflection.

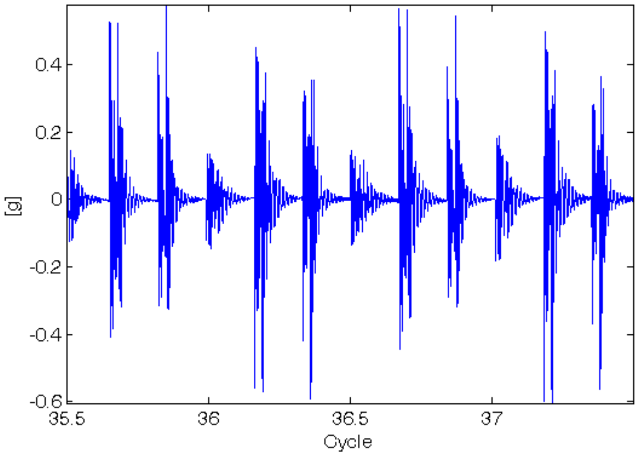


Figure 12. Small scale specimen bearing model results: inner ring radial acceleration

The relation between the outer ring deflected zone length and the bearing dynamics is more complicated. The deflected zone is longer than the angular distance between the balls. Therefore, part of the time, the inner ring is supported by 12 balls. This small period is not enough for stabilization. Therefore, the maximum inner ring center deflection is defined by the length of the period that it is supported by 12 balls. This means that the inner ring center deflection is proportional to the length of the outer ring deflected zone.

The model results show that the acceleration of the inner ring center is a sensitive indicator of the defect existence. Therefore, the power spectral density (PSD) of the simulated acceleration is evaluated for further comparison with the test results.

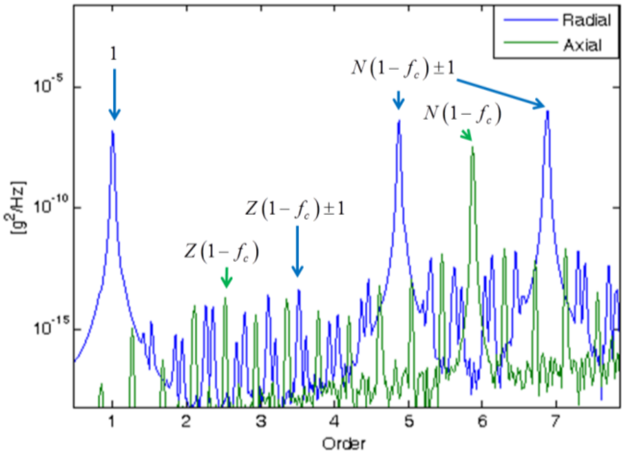


Figure 13. Small scale specimen bearing model results: PSD of the inner ring acceleration

The PSD of the axial acceleration of the inner ring center (see Figure 13) reveals peaks at the order:

$$P_h = Z(1 - f_c) \tag{3}$$

Where Z is an index and f_c is the cage order which is defined as:

$$f_c = \frac{1}{2}\left(1 + \frac{d}{D}\cos(a)\right) \tag{4}$$

The significant peaks of P_h appear when Z is a multiple of number of balls N, i.e. a harmonic of the Ball Pass Frequency Outer race (BPFO).

The spectrum of the acceleration in the radial direction reveals the first harmonic of the shaft speed and peaks at the order:

$$P_s = Z(1 - f_c) \pm 1 \tag{5}$$

The significant peaks of P_s appear when Z is a multiple of the number of the balls, i.e. sidebands of the shaft speed around a harmonic of the BPFO.

6.1.2. Full scale bearing implementation

Implementation of the dynamic model with a CH-53 swashplate bearing parameters was examined. Table 2 presents the bearing parameters.

Table 2. Full scale bearing parameters

Pitch diameter	$19.03[in]$
Ball diameter	$0.5[in]$
Number of balls	92
Outer race operational frequency	*3.16 [Hz]*
Contact angle	30°
Defect length (azimuth)	70°

In agreement with the small scale specimen bearing results the radial direction significant peaks of P_s indicate sidebands of the shaft speed around a harmonic of the BPFO. The axial direction peaks of P_h indicate harmonics of the BPFO. The axial acceleration shows peaks at the BPFO harmonics (44.95, 89.91 …).

The main parameters affecting the rings acceleration are the number of balls, the axial load and defect length. In order to clearly identify the fault, a relatively large primary defect was produced. As mentioned in Table 2, a defect length of 70° was inserted to the outer spacer.

The effect of the major increase in the number of balls can be seen when comparing the axial acceleration of the small scale specimen bearing (Figure 13) and the full scale bearing (Figure 14).

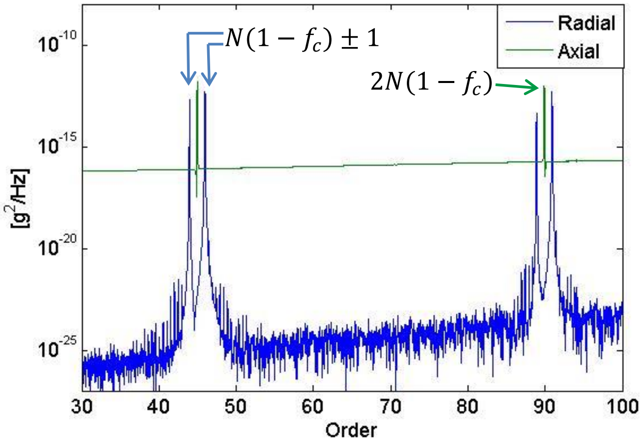


Figure 14. Full scale bearing model results: PSD of the inner ring acceleration

Examination of the model result for various defect lengths is needed. These results would be used for development of a generic failure algorithm independent of the bearing dimensions or the number of balls.

6.2. Small scale specimen experiments

The small scale specimen test results include examination of the fault influence on both radial and axial acceleration spectra (see Figure 15). The comparison between the spectra of the faulty supported bearing in the radial and axial directions reveals that the axial acceleration signal is dominant at the BPFO order harmonics and the radial acceleration signal is predominated by the shaft speed sidebands. These results correspond to the model results presented in Figure 13.

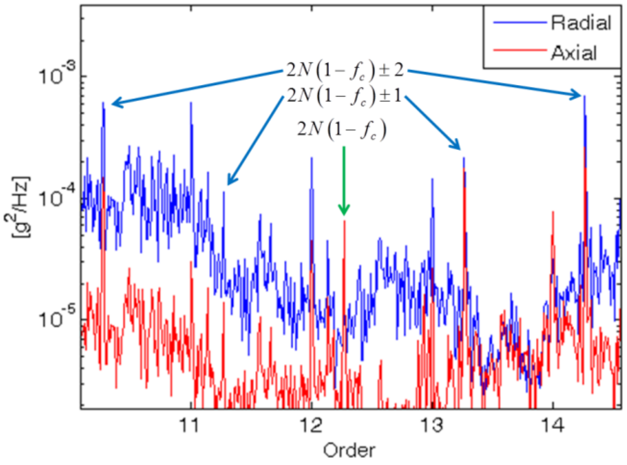


Figure 15. Small scale specimen results: PSD of the faulty supported bearing

The model results, presented in Figure 13, show peaks at orders of $Z(1-f_c)$ and of $Z(1-f_c)\pm1$. When Z is not a multiple of number of balls N, these peaks are low compared to the BPFO harmonics and to their shaft speed sidebands. Therefore, those peaks cannot be observed in the test results and are probably masked by other sources of vibrations.

6.3. Full scale test rig preliminary results

The rig was tested with a healthy bearing at 185 [RPM] without load. Transmission components frequencies were calculated and identified.

The acceleration PSD of the healthy bearing is presented in Figure 16. The PSD was calculated on the resampled acceleration acquired at sensor location 2 (as shown in Figure 9) in the bearing radial direction. As expected the bearing tones cannot be observed with a healthy swashplate bearing. Compared to the small scale specimen results, it can be seen that the vibration signature contains larger amount of information due to more inputs. This illustrates the expected difficulties in recognizing the pattern in a more realistic environment. Noises resulting from the transmission mask the bearing fundamental frequencies as shown in the multiple peaks appearing around the BPFO second harmonic (89.9). The model results point to the defect location but further analysis is needed in order to recognize the fault in complex mechanical systems.

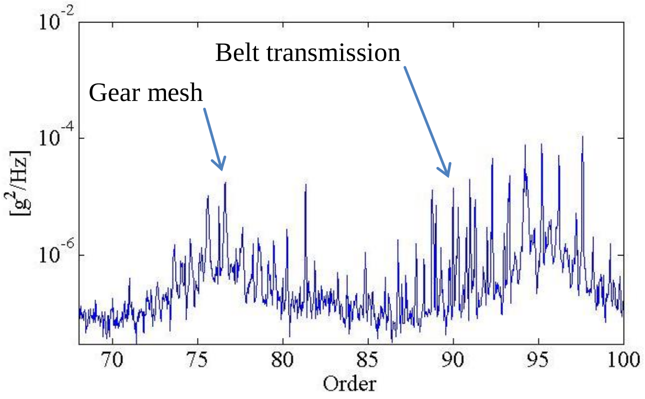


Figure 16. Full scale test rig results: order spectrum of the healthy swashplate acceleration

7. Conclusions

This paper presents the research approach for the development of an algorithm for detection of a failure of the CH-53 swashplate bearing external spacer. The failure causes a lack of support of the swashplate bearing, thus creating a deformation of the outer ring. This study integrates the results of a 3D dynamic model, developed for assessment of the defect pattern, and results from experimental systems.

The new dynamic model predicts a defect pattern for the unique, newly-researched failure. The results present the defect pattern in the radial and axial directions. A similar schematic pattern was obtained in both the small scale specimen bearing and the full scale bearing. This pattern can be used in the future to develop a generic algorithm for detection of the failure independent of the bearing dimensions or the number of balls.

The small scale specimen is designed for fast experiments and easy simulation of a wide range of faults. Experiments of the small scale specimen simulating the fault are in good agreement with the model results.

A full scale test rig was constructed to simulate more realistically the signals generated by the rotating swashplate. The test rig configuration allows swashplates to be easily replaced and operated. The test rig was constructed to function under a static load and rotational speed of an operational main rotor while keeping a low number of moving parts and using a small electric motor. Preliminary experiments conducted on the test rig confirm the increase in noise sources. The appearance of multiple peaks around the BPFO second harmonic observed in the test rig experiment illustrates the need to integrate the model results with experiments.

In the next steps it is planned to complete the seeded fault tests on the full scale test rig and examine the algorithm with data from the helicopter blades test facility and the CH-53 helicopter.

ACKNOWLEDGEMENT

We gratefully acknowledge the invaluable support of Dr. Avi Weinreb and Tal Yehoshua.

We would also like to thank the Perlston Foundation.

REFERENCES

Kogan, G., Bortman, J., Kushnirsky, A. & Klein, R. (2012). Ball bearing modeling for faults simulation. *The Ninth International Conference on Condition Monitoring and Machinery Failure Prevention Technologies* (no. 1, pp. 1–8).

Tkachuk, P. & Strackeljan, J. (2010). A 3D-ball bearing model for simulation of axial load variations. *The Seventh International Conference on Condition Monitoring and Machinery Failure Prevention Technologies* (pp. 1–10).

Keller, J. & Grabill, P. (2005). Inserted fault vibration monitoring tests for a CH-47D Aft swashplate bearing. *The American helicopter society 61st Annial Forum.*

Bayoumi, A., Goodman, N., Shah, R., Roebuck, T., Jarvie, A. & Eisner, L. (2008). Conditioned-Based Maintenance at USC - Part III: Aircraft Components Mapping and Testing for CBM. *The American Helicopter Society Specialists' Meeting on Condition Based Maintenance. Huntsville*.

Blechertas, V., Bayoumi, A., Goodman, N., Shah, R. & Shin, Y. J., (2009). CBM Fundamental Research at the University of South Carolina: A Systematic Approach to U.S. Army Rotorcraft CBM and the Resulting Tangible Benefits. *The American Helicopter Society Technical Specialists' Meeting on Condition Based Maintenance. Huntsville*.

Chin, H. H., Mayhew, E. & Green, D. L. (2005). Assessing Bearing Health for Helicopter Power Train Systems. *American Helicopter Society 61st Annual Forum. Grapevine*.

Blunt, D. M. & Keller, J. A. (2006). Detection of a fatigue crack in a UH-60A planet gear carrier using vibration analysis. *Mechanical Systems and Signal Processing*, 20(8), 2095–2111.

Dempsey, P., Branning, J. & Arsenal, R. (2010). Comparison of Test Stand and Helicopter Oil Cooler Bearing Condition Indicators. *AHS 66th Annual Forum and Technology*.

Budynas, R. & Nisbett, K. (2006). Shigley's Mechanical Engineering Design (*Mcgraw-Hill Series in Mechanical Engineering*). McGraw-Hill Science/Engineering/Math.

Multivariate Fault Detection using Vector Autoregressive Moving Average and Orthogonal Transformation in Residual Space

Francisco Serdio[1], Edwin Lughofer[2], Kurt Pichler[3], Thomas Buchegger[4], Markus Pichler[5], and Hajrudin Efendic[6]

[1,2] *Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, Austria*
francisco.serdio@jku.at edwin.lughofer@jku.at

[3,4,5] *Austrian Competence Center of Mechatronics, Linz, Austria*
kurt.pichler@lcm.at thomas.buchegger@lcm.at markus.pichler@accm.co.at

[6] *Institute for Design and Control of Mechatronical Systems, Johannes Kepler University Linz, Austria*
hajrudin.efendic@jku.at

Abstract

We propose the use of multivariate orthogonal space transformations and Vector Autoregressive Moving-Average (VARMA) models in combination with data-driven system identification models to improve residual-based approaches to fault detection in rolling mills. Introducing VARMA models allows us to build k-step ahead multi-dimensional prediction models including the time lags that best explain the target. Multivariate orthogonal space transformations provide estimates for the dynamical parameters by rewriting the equation set of the system at hand, decomposing the measured data into process and residuals spaces. Modeling in the process space then produces much more accurate models due to dimensionality (noise) reduction. Since we use an unsupervised scheme that requires a priori neither annotated samples nor fault patterns/models, both model identification and fault detection are based solely on the on-line recorded data streams. Our experimental results demonstrate that our approach yields improved Receiver Operating Characteristic (ROC) curves than methods that do not employ vector autoregressive moving-average models and multivariate orthogonal space transformations.

1. INTRODUCTION

Unscheduled machine downtime could be reduced significantly by accurate condition monitoring and early detection of faults. Thus, expensive repair costs could be minimized and production efficiency increased. The concept of *fault detection* was formally defined by the ”SAFEPROCESS” IFAC Technical Committee as the ”Determination of the faults present in a system and the time of detection”. Further, they also defined a *fault* as ”An unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition” (Isermann & Ballé, 1997). At the time when the Committee introduced the terminology to the fields of supervision, fault detection and diagnosis, most applications supported fault detection by simple threshold logic or hypothesis testing, but the number of publications describing much more complex techniques (such as fuzzy logic or neural networks) was steadily growing. The main challenge in our application is the *detection of faults*, without (i) an analytical description of faults and process models and (ii) a collection of typical fault patterns. Approaches using model-based techniques that rely on analytical process (system) models (Dong, Liu, & Li, 2010) or employ models deduced from the physical definition of the appearance of a fault (Bolt et al., 2010) or take advantage of a robust observer design (Theilliol, Mahfouf, Ponsart, Sauter, & Gama, 2010) are unsuited to our application. Neither are time-series analysis (Chandola, Banerjee, & Kumar, 2009) and vibration monitoring in frequency spaces (spectrograms, etc.) (Pichler, Lughofer, Buchegger, Klement, & Huschenbett, 2012) because the measurement signals are not completely smooth and continuous in their time line appearance but may show jumping patterns even in fault-free states due to, for instance, varying systems states. Pattern recognition and classification approaches (Bishop, 2006) cannot be employed (i) due to the absence of fault patterns and (ii) because annotated samples are not available since simulating real faults directly at the system involves high costs and the risk of component breakdown.

For fault detection in rolling mills, we propose a residual-based approach that builds upon data-driven techniques, combined with vector autoregressive moving-average models and

Francisco Serdio et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

multivariate orthogonal space transformations. Further, we use a dynamic threshold based on a tolerance band that tracks the residuals of the models over time after being normalized by a local error bar. The model-based FD approach takes advantage of the nominal model of the system to generate residuals (deviations between model predictions and measured targets) containing information about the faults. Clearly, the quality of the model is essential for fault detectability and isolability and the avoidance of "false alarms" (Frank, Alcorta, & Köppen-Seliger, 2000).

We conducted our experiments using *process history*-based methods which only require large amounts of historical process data. There are different ways in which this data can be transformed and presented as a priori knowledge to a diagnostic system. Principal Component Analysis (PCA) and Partial Least Squares (PLS) together with statistical pattern classifiers form a major component of statistical feature extraction methods (Venkatasubramanian, Rengaswamy, Kavuri, & Yin, 2003). Section 2 explains how we handle VARMA models and describes both PCA and PLS as preprocessing and dimension reduction tools and how to combine them with linear or fuzzy methods. Section 4 comments briefly on preliminary results which show that the proposed techniques yield better Receiver Operating Characteristic (ROC) curves than previous fault detection methods. Section 5 concludes the paper and presents future research directions.

2. MODELING

2.1. Methodology step by step

Our methodology at a glance is shown in Figure 1. In the workflow, four different pathways can be follow, thus obtaining the different combinations of models, i.e. the models without any additional issue (raw models), the VARMA models (including the lags of the channels), the models trained on the projected spaces and the VARMA models trained on the projected spaces. Along these pathways, how a dataset is spanned to include lags and how a model can be trained in a projected space is also depicted. The methodology also include a preprocessing step cleaning the data.

2.2. VARMA (Multi-regressive) Models

Inspired by Auto-Regressive Moving Average (ARMA) models, which model a channel by its own lags, we use lags in our multidimensional and multichannel approach. Thus, the dataset containing the candidate channels to explain a target is first spanned with the candidate channels' lags (Figure 1). To this end, each channel in the dataset is delayed several times by different amounts (1 to L in our experiments). Then, the model of each channel including all the channels' lags is learned by means of a modified variant of forward selection (Miller, 2002) over the spanned dataset. Thus, we obtain separate and independent multivariate time series models, $f_i, i = 1, ..., M$.

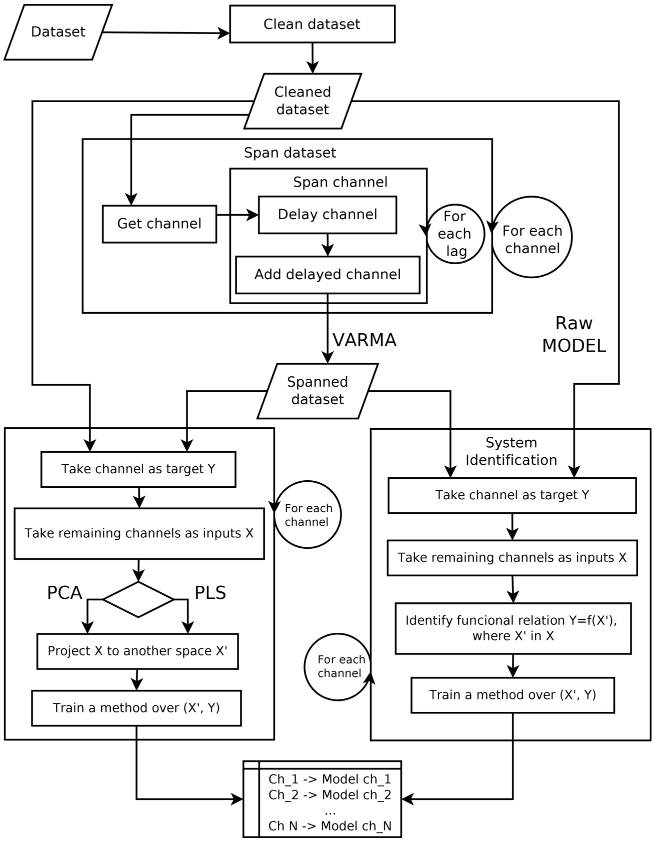


Figure 1. Methodology workflow. All combinations of methods can be obtained following the four different pathways.

Starting from the general definition of the model of a channel ch_i as a function f_i of other channels (where no lags are used), and using the notation [t] to express the lag operator (also known as backshift operator, *at time t*), it becomes straightforward to pass from the particular model definition where the set of candidate channels for explaining the target is obtained from the spanned dataset including lags (cf. Figure 1). The functional relation without lags is then

$$ch_i[t] = f_i(ch_1[t], ..., ch_{i-1}[t], ch_{i+1}[t], ..., ch_M[t]) \,, \quad (1)$$

while the functional relation with lags is

$$ch_i[t] = f_i(ch_p[t-k], ..., ch_q[t-l], ..., ch_r[t-m]) \,, \quad (2)$$

where k varies from 0 (i.e., allowing channels without lags to participate in the model definition) to L, and $p, q, r \neq i$ (lags of ch_i are not included) but not necessarily $p \neq q \neq r$. Thus, models are potentially obtained where no lags, different lags from the same channel, or different lags from different channels appear as input. To do not allow lagged versions of ch_i will avoid introducing the information of the (potentially faulty) channel as an input of the final model.

Previous work using autoregressive models in the fault detection area can be found in Schöener, Moser, and Lughofer (2008), M. Yang and Makis (2010) and T. Yang (2006). These use AR, ARX and ARMA models respectively, but we have not found evidence of VARMA models applied for the purpose of fault detection. However, our results demonstrate that this new functional relation enriches the model set since they yield better ROC curves (see section 4). A detailed description of ARMA models and its variations is provided in Holan, Lund, and Davis (2010).

2.3. (Fuzzy) Principal Components Regression

Principal component analysis (PCA) (Jolliffe, 2002) is a vector space transformation that identifies the most meaningful basis to re-express the original space while preserving maximum variance in a minimum number of dimensions and filtering out the noise. When starting from correlated data, PCA is a good technique to transform the set of original process variables into a new set of uncorrelated variables that explain the main trends of the process. The PCA decomposition is

$$X = \sum_{h=1}^{m} v_h r_h^T + G = VR^T + G\,, \tag{3}$$

where V, R and G are scores, loadings and residuals, respectively. Since PCA was first introduced as a tool (Jackson & Mudholkar, 1979), numerous publications have described fault detection using PCA, comprising both practical applications (Odgaard, Lin, & Jørgensen, 2008) and theoretical studies (Tamura & Tsujita, 2007). We, however, do not use PCA directly as a fault detection method; instead we go a step further and use a Principal Component Regression (PCR) technique –see Jolliffe (2002). PCR exploits the PCA capabilities as a dimensionality reduction tool in order to produce a new set of regressors to train a linear method with them (see Figure 1). Since the model trained with the PCA scores can be changed (a linear model is used in PCR), several variants are possible. We used both PCR and Fuzzy PCR (FPCR) in our experiments. For FPCR we used a Takagi-Sugeno fuzzy model (TSK) trained with an extended version of *SparseFIS* as explained in Lughofer and Kindermann (2010).

SparseFIS relies on a top-down approach that takes a maximum number of (allowed, feasible) rules as inputs and discards out as many rules as possible to find a compact rule set through an enhanced numerical optimization process which is a non-linear version of the *projected gradient descent algorithm* (Daubechies, Defrise, & Mol, 2004). SparseFIS also uses an extended form of TS fuzzy models, since it integrates rule weights $\rho \in [0,1]$ as rule importance levels, thus preferring some local regions over others.

$$\hat{f}(\vec{x}) = \hat{y} = \sum_{i=1}^{C} \Psi_i(\vec{x}) \cdot l_i(\vec{x}) \quad \Psi_i(\vec{x}) = \frac{\rho_i \varphi_i(\vec{x})}{\sum_{j=1}^{C} \rho_j \varphi_j(\vec{x})}\,, \tag{4}$$

where l_i are the linear consequent functions and $\varphi(\vec{x})$ denotes the activation degree of the i-th rule. The analytical model of FPCR results from substituting the inputs in (4) by those transformed (and usually reduced) by the PCA orthogonal projection in (3).

2.4. (Fuzzy) Partial Least Squares

Partial Least Squares (PLS) handles correlated inputs, reducing the dimensionality of the input and target variables by projecting them to the directions maximizing the covariance between them. The projection combines highly correlated variables into new one-dimensional variables. The PLS projection explained in terms of scores and loadings corresponds to Equations (5) and (6).

$$X = \sum_{h=1}^{m} t_h p_h^T + E = TP^T + E\,. \tag{5}$$

$$Y = \sum_{h=1}^{m} u_h q_h^T + F = UQ^T + F\,. \tag{6}$$

These equations are called PLS outer relation, where T and U are input and target scores, P and Q are input and target loadings, and E and F are input and target residuals. The relation between score vectors t_h and u_h is called inner relation.

Some previous work combined PLS with fuzzy systems in what is called Fuzzy Partial Least Squares (FPLS) (Bang, Yoo, & Lee, 2002), a subset of Nonlinear Partial Least Squares (NPLS) techniques. This FPLS approach takes the PLS outer relation as a reduction tool to remove collinearity and then applies a Takagi fuzzy model to capture and model the nonlinearity in the projected latent space. To the best of our knowledge, FPLS has not been used in fault detection, though there is published literature on PLS for process monitoring describing its theoretical properties (Li, Qin, & Zhou, 2010), on PLS and its variants with practical applications in fault detection (Muradore & Fiorini, 2012), Wang, Kruger, and Lennox (2003), and on fuzzy systems successfully applied in process monitoring tasks (Angelov, Giglio, Guardiola, Lughofer, & Luján, 2006). Our results demonstrate that it is also feasible to use the combination of PLS+TSK (i.e., FPLS) in the fault detection domain. As described in Section 2.3, we used a TSK model trained with an extended version of *SparseFIS*. Analogously, the analytical model of FLSR results from substituting the inputs in (4) by those resulting from the PLS projection (5).

2.5. Dynamic Residual Analysis (On-line)

The online fault detection uses a dynamic analysis of the *residuals* of the system. Figure 2 shows how the residual signal may be affected by a fault in a monitored channel. The (real) example illustrates how the dynamic band *mimics* the fault when the fault is too small and how the tolerance band is exceeded and a fault is signaled when the fault intensity is sufficiently great to affect the residual signal. This strategy provides a flexible way of handling the residuals, reacting more dynamically to their basic trend.

In order to formulate the tolerance band for each new incoming data sample $\vec{x}(k)$ at time instance k, its affected residuals $res_1(k),...,res_{m^*}(k)$ to all m^* models $f_1,...,f_{m^*}$ are calculated and normalized according to the models' confidence

$$res_m(k) = \frac{|f_m(\vec{x}(k)) - y_m(k)|}{conf_m(k)} \tag{7}$$

where f_m is the model prediction, y_m is the observed target and $conf_m$ is a *local error bar* capturing different certainty levels (confidence) for different regions of the input space according to distribution, noise level, density, etc. of our training data. The inverse covariance matrix of the model inputs X^TX provides a reliable representation of dense and sparse data regions (Nelles, 2001), and by multiplying it with the noise level σ^2, the noise intensity is incorporated

$$cov\{par_m\} = \sigma^2(X^TX)^{-1}\,. \tag{8}$$

Since the noise level of the training data is usually unknown, we use an unbiased estimator (Söderström & Stoica, 1988)

$$\widehat{\sigma}^2 = \frac{2\sum_{k=1}^{N}(y(k) - \widehat{y}(k))^2}{N - deg}, \tag{9}$$

where N is the number of data samples in the training data and deg is the degrees of freedom. For new test data samples X_{test}, the covariance matrix of the model outputs $\widehat{y}_{test}$ becomes

$$cov\{\widehat{y}_{test}\} = \vec{x}_{test}\widehat{\sigma}^2(X^TX)^{-1}\vec{x}_{test}^T = conf_m\,, \tag{10}$$

and is equivalent to the confidence. Thus, Equation (10) express the local error bars in the case of Linear Regression.

As long as the linear parameters of the consequent functions in the Takagi-Sugeno fuzzy systems are estimated by a local learning approach, the formula can also be exploited for its partial local linear models. Then, a nestling of the consequent hyper-planes to the models surface can be observed as analyzed in Angelov, Lughofer, and Zhou (2008). This yields a good approximation of the global model consisting of local linear pieces, which can be used to calculate error bars for each rule consequent function separately and to connect them with weights, thus forming an overall error bar for the whole fuzzy model. The error bar of an evolving (TS) fuzzy model with C rules, for a specific sample $\vec{x}_{act}$ can be calculated by

$$\widehat{y}_m \pm \sqrt{cov\{\widehat{y}_m\}} = \widehat{y}_m \pm \frac{\sum_{i=1}^{C}\varphi_i(\vec{x}_{act})\sqrt{cov\{\widehat{y}_{m,i}\}}}{\sum_{i=1}^{C}\varphi_i(\vec{x}_{act})}, \tag{11}$$

where $\widehat{y}_i$ is the estimated value of the i^{th} rule consequent function, for which $cov\{\widehat{y}_{m,i}\}$ is calculated as in (10) by using the inverse weighted matrix $(X_i^TQ_iX_i)^{-1}$ corresponding to the i^{th} rule and the noise variance as in (9). Hence,

$$cov\{par_{m,i}\} = \widehat{\sigma}^2(X_i^TQ_iX_i)^{-1}\,. \tag{12}$$

The symbol $\varphi_i(\vec{x}_{act})$ denotes the membership degree of the current point to the i^{th} rule and $\widehat{y}_m$ the output value from the TS fuzzy model for the current input point $\vec{x}_{act}$.

The tolerance band equation is completed with the equations of the incremental/decremental tracking along the timeline, where the mean (13) and standard deviation (14) of the residuals are computed.

$$\mu_i(k) = \frac{N_1\mu_i(k-1) + res_i(k) - res_i(k-T)}{N_2} \tag{13}$$

$$\begin{aligned}\sigma_i(k) =& \frac{1}{N_2}(N_1\sigma_i(N-1) + N_2\Delta\mu_i(k)^2 + \\ & (\mu_i(k) - res_i(k))^2 - N_2\Delta\mu_i(k-T)^2 - \\ & (\mu_i(k-T) - res_i(k-T))^2)\end{aligned} \tag{14}$$

where $res_i(k-T) = 0, \mu_i(k-T) = 0, N_1 = k-1, N_2 = k$ for all $k < T$ and $N_1 = N_2 = T$ for $k \geq T$, and $\Delta\mu_i(k) = \mu_i(k) - \mu_i(k-1)$. Combining (13) and (14) yields the tolerance band around the residuals of a given model

$$tolband_i(k) = \mu_i(k) + n * \sigma_i(k)\,, \tag{15}$$

where n is the parameter in the fault detection method and produces the ROC curve when varied.

Finally, our FD system is composed of M identified models, thus producing M residuals signals that are evaluated in parallel. The OR condition is used to identify faults: if a tolerance band in (15) is exceeded by at least one model, a fault alarm is triggered.

3. EXPERIMENTAL SETUP

We have not real faults in our datasets, so we were required to introduce them artificially. This was done by an automatic process, selecting 10 channels at random and introducing a fault in each one. The fault parameters, i.e. type (abrupt of incipient), shape (only for incipient faults) and position were also chosen randomly. The process intentionally skips introducing faults in the warm-up of the detection system, which is an initial stage (20 samples) where faults are neither assumed nor signaled. It also skips faults in regions where the nominal signal is constantly zero. Keeping the random-chosen-parameters of a fault, it is replicated with 5 different intensities (5%, 10%, 20%, 50%, 100%). Since each dataset is composed by 5 different files to test, this led to 50 faults to be detected in each tested dataset at each fault intensity.

This automatic process produces the test files of a run. A run therefore is the execution of a method over these test files, one time per fault intensity.

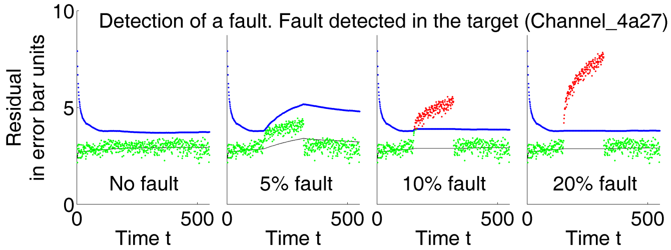


Figure 2. Detection of a fault (drift with exponential shape) in the residual space using a tolerance band. Different fault intensities are shown. The upper line (blue) is a dynamic threshold, whereas the lower line (green when below the threshold, red otherwise) are the residuals. The thin line (black) belongs to the trend of the residuals, i.e. its mean.

The datasets were composed by 240 channels, that after the cleaning process were cut down to 51 and 45 for Dataset 1 and Dataset 2 respectively. There were 9676 samples to train and 8959 samples to test in Dataset 1, whereas Dataset 2 had 11462 samples to train and 12716 samples to test.

4. Results

We base our results on the average of 10 runs, leading to 500 candidate faults to be detected in each tested dataset at each fault intensity. The different fault intensities determine the minimum deviation from the nominal state that a method can detect. Since the results are expressed in terms of ROC curves, the outputs of the experiments are comparable. The sensitivity parameter is the width n in the $n*\sigma$ tolerance band (15); it was increased by steps of 1, from 1 to 10. As we were especially interested in low false alarm rates ($< 10\%$), the X axis of the ROC curves (Figure 3) ranges from 0 to 10 (%). The Y axis denotes the detection rate, and the larger the Area Under the Curve (AUC), the better the method.

Figure 3, up, left, shows how Linear Regression using the spanned datasets (thus obtaining VARMA models) and transforming the dataset using PCA improves ROC curves compared to those of purely linear methods. When the dataset is not spanned but only transformed, the ROC curve produced by the test is even better (recall that *better* means that the Area Under the Curve (AUC) is greater). In this case, both the expansion of the dataset using lags and the orthogonal transformation help to improve the fault detection capabilities, but when combined, the result is not as good as when training the method only in the transformed (i.e., PCA) space. This result seems to be consistent with the analogous case using SparseFIS instead of Linear Regression. Figure 3, up, right, illustrates a behaviour similar to that described previously. SparseFIS produces an improved ROC curve when the method is trained (and tested) on the spanned and transformed dataset, that is, using VARMA models and PCA. When the method is trained with the transformed dataset without lags (i.e. without VARMA models), the ROC is better for one dataset but worse for the other. Therefore, PCA as a preprocessing stage helps to improve the fault detection capabilities of our approach either with Linear Regression or with SparseFIS, but to spanning the dataset with lags seems to yield no further improvement.

The results are different when PLS is used. Figure 3, down, left, shows how Linear Regression trained in a transformed space using PLS without spanning the datasets (i.e. without VARMA models) produces better ROC curves than the linear method itself. With the PLS transformation, the lags also play a role. When the Linear Regression is trained in both transformed and spanned space, the VARMA models yield a greater AUC, as seen in the ROC curves. In Figure 3, down, right, where Linear Regression is substituted by SparseFIS, the result is consistent: the method trained (and tested) with VARMA models in the transformed space using PLS produces better ROC curves than the model alone. Then, the use of VARMA models on the transformed space still produce a gain in the ROC curves. This is particularly noticeable for 'Dataset 1', where the use of lags clearly increases the AUC.

Tables **??** and **??** show the detection rates for each method when establishing thresholds for the false alarm rates in 3%, 5% and 10%, The highest rate per column is highlighted in bold font and 'n/a' denotes that the method is not able to achieve a false alarm rate of $X\%$ or lower.

As can be seen, the methods trained in the transformed space yield better results than methods trained in the original space. As for transformation, PLS is clearly preferable to PCA; although PCA produces better results than methods using the original space, it is outperformed by PLS. VARMA models and PLS appears to be a good combination, since the winning approaches use a transformed space and a VARMA model. The results do not indicate which model is preferable, since Linear Regression and SparseFIS are winners the same number of times, i.e. 9 out of 18. Apparently, SparseFIS better detects low-intensity faults, whereas Linear Regression is best suited for high-intensity ones.

5. Conclusions and Future Works

We have introduced vector autoregressive moving-average models in combination with multivariate orthogonal space transformations to the fault detection domain. Using ROC curves, we have demonstrated that, in terms of detection rates, our approach outperforms approaches in which either the datasets are not spanned with lags (time delays) or orthogonal transformations are not performed before the modeling stage. When PCA transformation is used, the expansion of the datasets with the lags produces no clear improvement in fault detection capability, and the VARMA models can be ignored in this case. In contrast, when the datasets are transformed using PLS, VARMA models help to improve the ROC curves, and therefore their fault detection capabilities.

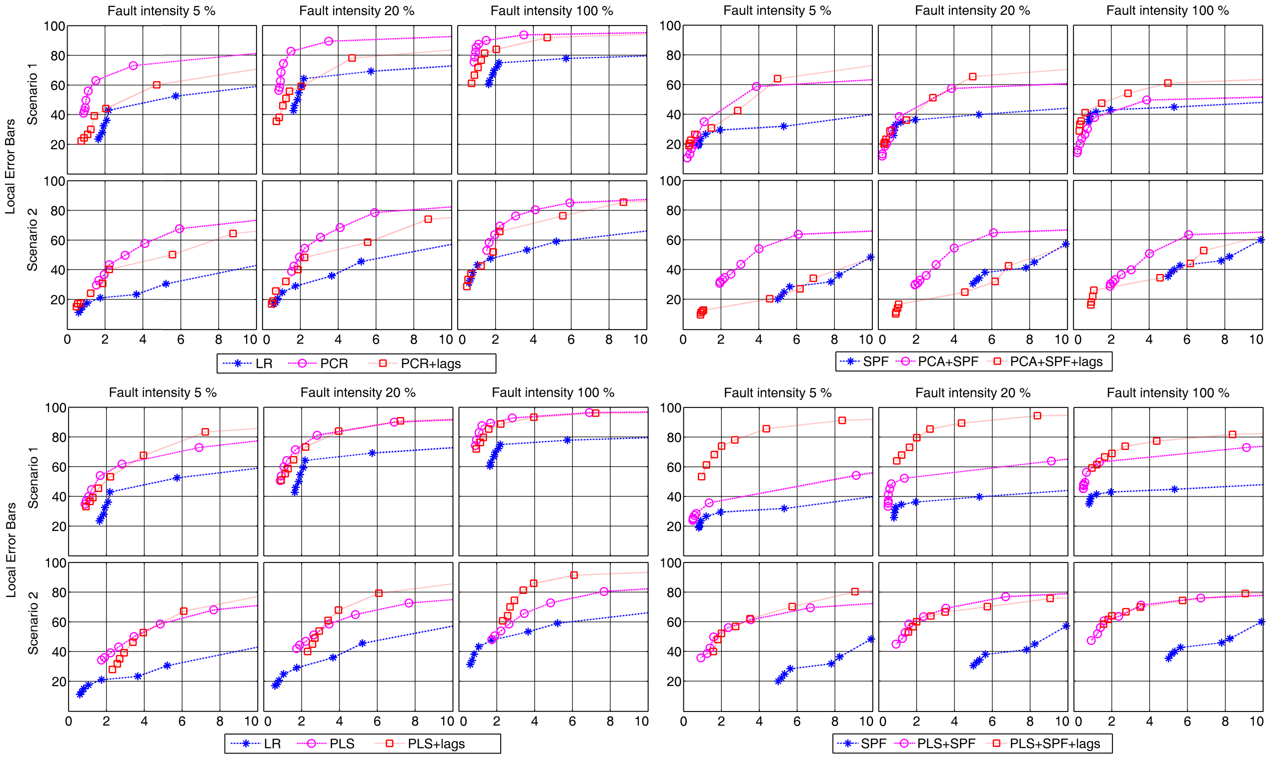


Figure 3. ROC curve (% detection (Y) vs % false alarm (X)) improved. Initial (worst) curves belong to Linear Regression (LR); curves are improved when using PCR with lags; best curves use PCR without lags.

Table 1. Dataset 1. Highest detection rates with different fault and false alarm levels

Method	5% Fault			20% Fault			100% Fault		
	Max. False Alarms			Max. False Alarms			Max. False Alarms		
	< 3%	< 5%	< 10%	< 3%	< 5%	< 10%	< 3%	< 5%	< 10%
Linear Reg.	42.95	42.95	52.53	64.24	64.24	69.24	74.89	74.89	77.86
PCR	63.05	73.15	73.15	82.70	89.40	89.40	89.95	93.72	93.72
PCR + lags	47.10	64.10	64.10	63.39	79.95	79.95	86.41	94.40	94.40
PLS	61.77	61.77	72.91	81.22	81.22	89.98	92.84	92.84	96.43
PLS + lags	73.54	81.67	88.79	85.36	89.16	93.15	**93.09**	**94.99**	**96.86**
SparseFIS	34.45	34.45	39.87	45.08	45.08	49.65	51.59	51.59	53.67
PCA + SparseFIS	35.10	58.78	58.78	38.56	57.38	57.38	37.91	49.65	49.65
PCA + SparseFIS + lags	42.50	64.04	64.04	51.16	65.39	65.39	54.17	61.06	61.06
PLS + SparseFIS	35.78	35.78	54.26	52.37	52.37	63.92	63.24	63.24	73.05
PLS + SparseFIS + lags	**78.11**	**85.65**	**91.30**	**85.44**	**89.42**	**94.41**	73.81	77.33	81.86

Table 2. Dataset 2. Highest detection rates with different fault and false alarm levels

Method	5% Fault			20% Fault			100% Fault		
	Max. False Alarms			Max. False Alarms			Max. False Alarms		
	< 3%	< 5%	< 10%	< 3%	< 5%	< 10%	< 3%	< 5%	< 10%
Linear Reg.	21.08	23.36	30.57	29.06	35.97	45.63	47.72	53.40	59.07
PCR	43.49	57.78	67.69	54.57	68.43	78.58	69.49	80.46	85.11
PCR + lags	40.13	40.13	50.21	48.31	48.31	58.59	65.76	65.76	76.46
PLS	43.15	58.62	68.09	51.28	64.90	72.69	58.54	72.84	80.44
PLS + lags	39.13	52.79	67.20	53.78	67.87	**79.24**	**74.45**	**85.93**	**91.42**
SparseFIS	n/a	14.51	29.68	n/a	20.43	36.55	n/a	32.79	47.88
PCA + SparseFIS	36.96	54.00	63.72	35.96	54.48	64.75	36.66	50.76	63.42
PCA + SparseFIS + lags	12.63	20.30	34.04	16.60	24.77	42.49	26.04	34.40	52.77
PLS + SparseFIS	56.08	61.34	69.55	63.49	**69.15**	76.96	63.65	71.26	76.13
PLS + SparseFIS + lags	**56.76**	**62.05**	**80.37**	**63.92**	66.68	75.75	66.76	69.92	79.05

Testing our approach with more datasets and performing statistical tests on the results will be part of future work. Future work will also concentrate on Fault Identification (FI). In FI, the Fault Detection system goes further, providing additional indications, with confidence measurements, of signaled faults and their locations; this should equip the operator with more information for the decision making process. How a fault deforms a model is a potential starting point for future research.

Acknowledgment

This work formed part of the research program at the Austrian Center of Competence in Mechatronics (ACCM), a part of the COMET K2 program of the Austrian Government. The work-related projects are kindly supported by the Austrian Government and Johannes Kepler University Linz. This publication reflects only the authors' views.

References

Angelov, P., Giglio, V., Guardiola, C., Lughofer, E., & Luján, J. (2006). An approach to model-based fault detection in industrial measurement systems with application to engine test benches. *Measurement Science and Technology*, *17*(7), 1809–1818.

Angelov, P., Lughofer, E., & Zhou, X. (2008). Evolving fuzzy classifiers using different model architectures. *Fuzzy Sets and Systems*, *159*(23), 3160–3182.

Bang, Y., Yoo, C., & Lee, I. (2002). Nonlinear pls modeling with fuzzy inference system. *Chemometrics and Intelligent Laboratory Systems*, *64*, 137–155.

Bishop, C. (2006). *Pattern recognition and machine learning*. New York, NY, USA: Springer.

Bolt, P., Batazzi, D., Belfiore, N., Gaspard, C., Goiset, L., Laugier, M., et al. (2010). Damage resistance and roughness retention of work rolls in cold rolling mills. *Revue de Métallurgie*, *107*(6), 245–255.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3).

Daubechies, I., Defrise, M., & Mol, C. D. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, *57*(11), 1413–1457.

Dong, M., Liu, C., & Li, G. (2010). Robust fault diagnosis based on nonlinear model of hydraulic gauge control system on rolling mill. *IEEE Transactions on Control Systems Technology*, *18*(2), 510–515.

Frank, P., Alcorta, E., & Köppen-Seliger, B. (2000). Modelling for fault detection and isolation versus modelling for control. *Mathematics and Computers in Simulation*, *53*(33), 259 - 271.

Holan, S., Lund, R., & Davis, G. (2010). The arma alphabet soup: A tour of arma model variants. *Statistical Surveys*, *4*, 232–274.

Isermann, R., & Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, *5*(5), 709–719.

Jackson, J., & Mudholkar, G. (1979). Control procedures for residuals associated with principal component analysis. *Technometrics*, *21*(3), 341–349.

Jolliffe, I. (2002). *Principal component analysis* (2nd ed.). New York, NY, USA: Springer.

Li, G., Qin, S., & Zhou, D. (2010). Geometric properties of partial least squares for process monitoring. *Automatica*, 204–210.

Lughofer, E., & Kindermann, S. (2010). SparseFIS: Data-driven learning of fuzzy systems with sparsity constraints. *IEEE Transactions on Fuzzy Systems*, *18*(2), 396–411.

Miller, A. (2002). *Subset selection in regression second edition*. Boca Raton, FL, USA: Chapman and Hall/CRC.

Muradore, R., & Fiorini, P. (2012). A pls-based statistical approach for fault detection and isolation of robotic manipulators. *Industrial Electronics, IEEE Transactions on*, *59*, 3167–3175.

Nelles, O. (2001). *Nonlinear system identification*. Berlin: Springer.

Odgaard, P., Lin, B., & Jørgensen, S. (2008). Observer and data-driven-model-based fault detection in power plant coal mills. *IEEE Transactions on Energy Conversion*, *23*(2), 659–668.

Pichler, K., Lughofer, E., Buchegger, T., Klement, E., & Huschenbett, M. (2012). A visual method to detect broken reciprocating compressor valves under varying load conditions. In *Proceedings of the asme 2012 international mechanical engineering congress & exposition* (p. to appear). Houston, TX, USA: ASME.

Schöener, H., Moser, B., & Lughofer, E. (2008). On preprocessing multi-channel sensor data for online process monitoring. In *Proceedings of the international conference on computational intelligence for modelling, control and automation (cimca)* (pp. 414–419). Vienna, Austria.

Söderström, T., & Stoica, P. (1988). *System identification*. Upper Saddle River, NJ, USA: Prentice Hall.

Tamura, M., & Tsujita, S. (2007). A study on the number of principal components and sensitivity of fault detection using pca. *Computers & Chemical Engineering*, *31*(9), 1035–1046.

Theilliol, D., Mahfouf, M., Ponsart, J., Sauter, D., & Gama, M. (2010). Design of a fault diagnosis system based on a bank of filter-observers with application to a hot rolling mill. *Transactions of the Institute of Measurement and Control*, *32*(3), 265–285.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S., & Yin, K. (2003). A review of process fault detection and diagnosis: Part iii: Process history based methods. *Com-*

puters & Chemical Engineering, *27*(3), 327–346.

Wang, X., Kruger, U., & Lennox, B. (2003). Recursive partial least squares algorithms for monitoring complex industrial processes. *Control Engineering Practice*, *6*(11), 613–632.

Yang, M., & Makis, V. (2010). ARX model-based gearbox fault detection and localization under varying load conditions. *Journal of Sound and Vibration*, *329*(24), 5209–5221.

Yang, T. (2006). A method of fast fault detection based on ARMA and neural network. In *Proceedings of the sixth world congress on intelligent control and automation (wcica)* (pp. 5438–5441). Dalian, China.

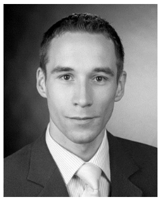
BIOGRAPHIES

Francisco Serdio was born in Asturias, Spain, in 1979. He received the Dipl.-Ing. degree in Computer Science and the MSc. degree in Soft Computing and Intelligent Data Analysis from the University of Oviedo, Asturias, Spain, in 2002 and 2011, respectively. He was involved in commercial software development companies from 2003 to 2010, including banking, stock market, CRM, and others. He is currently pursuing the Ph.D. degree with the Department of Knowledge-Based Mathematical Systems at the Johannes Kepler University, Linz. He received best paper awards at the IFAC Conference on Manufacturing Modelling, Management and Control, 2013. His research interests mainly include data-driven modeling and fault detection.

Edwin Lughofer was born in Ried im Innkreis, Austria, in 1972. He received his PhD. degree from the Department of Knowledge-Based Mathematical Systems at the Johannes Kepler University, Linz, where he is now employed as post-doctoral fellow. He has published more than 90 journal and conference papers in the fields of evolving fuzzy systems, machine learning and vision, data-driven modeling, clustering, fault detection, quality control and human-machine interaction, including a monograph on 'Evolving Fuzzy Systems' (Springer, 2011) and an edited book on 'Learning in Non-stationary Environments' (Springer, 2012). He received best paper awards at the 2006 International Symposium on Evolving Fuzzy Systems, at the 2008 International Workshop on Genetic and Evolving Fuzzy Systems and at the IFAC Conference on Manufacturing Modeling, Management and Control 2013, as well as the Royal Society Grant for knowledge exchange with Lancaster University in 2007.

Kurt Pichler was born in Grieskirchen, Austria, in 1981. He received the Dipl.-Ing. degree in technical mathematics from the Johannes Kepler University Linz (JKU), Austria, in 2005. He was with the Institute for Design and Control of Mechatronical Systems at the JKU and joined the Linz Center of Mechatronics (LCM) in 2007. He is currently pursuing the Ph.D. degree with the Department of Knowledge-Based Mathematical Systems at the JKU. His research interests include fault detection, parameter estimation, data-driven modeling and pattern recognition.

Thomas Buchegger was born in Linz, Austria, in 1974. He received the Dipl.-Ing. degree in mechatronical engineering in 2002 and his Dr. techn. (Ph.D.) degree in 2005 from the Johannes Kepler University, Linz, Austria. He is employed at the Linz Center of Mechatronics GmbH as Business Unit Manager for Sensors and Communication and Information Analysis. In the Comet K2 Center Austrian Center of Competence in Mechatronics GmbH he works as Vice Area Coordinator for Wireless Systems.



Markus Pichler was born in Linz, Austria, in 1976. He received the Dipl.-Ing. degree in mechatronics and the Dr.techn. degree from the Johannes Kepler University, Linz, Austria, in 2002 and 2007, respectively. He was with the Institute for Communications and Information Engineering at the University of Linz, and joined the Linz Center of Mechatronics (LCM) in 2002. His research interests include systems design, signal generation, digital signal processing and parameter estimation for radar and positioning systems as well as fault detection and error analysis. He has authored or co-authored over 25 publications, was the recipient of the 2004 European Microwave Association (EuMA) Radar Prize, and is a member of IEEE.

Hajrudin Efendic was born in Gorazde, Bosnia and Herzegovina, in 1970. He received the Dipl.-Ing. degree in automatic control and electronics and the MSc. degree in telecommunications from the University of Sarajevo, Bosnia and Herzegovina, in 1997 and 2001 respectively, and the Ph.D. degree in technical sciences from Johannes Kepler University (JKU), Linz, Austria, in 2012. He also received the MBA degree (summa cum laude) with specializations in management, international management and quantitative analysis from the University of Cincinnati, OH, U.S., in 2011. He is with the Institute for Design and Control of Mechatronical Systems at the JKU. His research is focused on data-based structural analysis, modeling and data-based modeling, monitoring and fault diagnosis in complex technical systems and in physiological and medical systems.

Simple Metrics for Evaluating and Conveying Prognostic Model Performance To Users With Varied Backgrounds

Michael E. Sharp

University of Tennessee Nuclear Engineering Department, Knoxville TN
msharp6@utk.edu

ABSTRACT

The need for standardized methods for comparison and evaluation of new models and algorithms has been known for nearly as long as there has been models and algorithms to evaluate. Conveying the results of these comparative algorithms to people not intimately familiar with the methods and systems can also present many challenges as nomenclature and relative representative values may vary from case to case. Many predictive models rely primarily on the minimization of simplistic error calculation techniques such as the Mean Squared Error (MSE) for their performance evaluation. This, however, may not provide the total necessary information when the criticality, or importance of a model's predictions changes over time. Such is the case with prognostic models; predictions early in life can have relatively larger errors with lower impact on the operations of a system than a similar error near the end of life. For example, an error of 10 hours in the prediction of Remaining Useful Life (RUL) when the predicted value is 1000 hours is far less significant than when the predicted value is 25 hours. This temporality of prognostic predictions in relation to the query unit's lifetime means that any evaluation metrics should capture and reflect this evolution of importance.

This work briefly explores some of the existing metrics and algorithms for evaluation of prognostic models, and then offers a series of alternative metrics that provide clear and intuitive measures that fully represent the quality of the model performance on a scale that is independent of the application. This provides a method for relating performance to users and evaluators with a wide range of backgrounds and expertise without the need for specific knowledge of the system in question, helping to aid in collaboration and cross-field use of prognostic methodologies. Four primary evaluation metrics can be used to capture information regarding both timely precision and accuracy for any series or set of prognostic predictions of RUL. These metrics, the Weighted Error Bias, the Weighted Prediction Spread, the Confidence Interval Coverage, and the Confidence Convergence Horizon are all detailed in this work and are designed such that they can easily be combined into a single representative "score" of the overall performance of a prediction set and by extension, the prognostic model that produced it. Designed to be separately informative or used as a group, this set of performance evaluation metrics can be used to quickly compare different prognostic prediction sets not only for the same corresponding query set, but just as simply from differing query data sets by scaling all predictions and metrics to relative values based on the individual query cases.

1. INTRODUCTION

The need for standardization in the area of evaluation for prognostics research has been well documented [Uckun et al 2008]. Work has even been presented on the evaluation of individual features or parameters as to their suitability for use in prognostic modeling [Coble 2010]. Recent effort has been focused on the standardization of prognostic model performance evaluation based on meaningful criteria that can be used to compare the output of prognostic models not only within given application, but across the field of predictive engineering [Saxena 2008]. Unfortunately, despite this large step forward in the evaluation of prognostic models, the existing metrics have yet to see wide spread acceptance and use. This may in part be due to both the seemingly and occasionally complicated nature of evaluating and interpreting these metrics.

This work seeks to incorporate group-based comparison into the offline prognostic algorithm evaluation process, and presents variants on some well-known performance metrics that are built upon a multitude of known cases to which the prognostic model has been applied.

Michael Sharp. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Specifically, four separate updated scalar metrics have been identified that sufficiently characterize and convey meaningful, intuitive information about the output predictions of a prognostic model: Weighted Error Bias (WEB), Weighted Prediction Spread (WPS), Confidence Interval Coverage (CIC), and the Confidence Convergence Horizon (CCH). Each one, detailed below, captures a key aspect and desirable quality of prognostic predictions that can be quickly, easily, and intuitively compared amongst separately developed models to rank and rate output performance. These metrics are built upon the errors and uncertainty associated with each prediction set, rewarding the minimization of both. To calculate both the errors and uncertainty of a prediction set, another descriptive series of values known as the Binned Percent Error is also defined and demonstrated in both use and interpretation in regards to the scalar metrics.

2. Background and Motivation

It has been suggested that depending on the needs of the end user, many different types of effective comparison algorithms could be employed such as cost/benefit analysis [Banks 2007]. However, given that many models may provide comparable results at similar costs, what are robust and useable methods for effectively ranking and expressing their relative effectiveness? Or more generally, what is the best way to convey results of a comparative analysis to someone that is not necessarily well versed in the science of prognostics or to a large audience with varied backgrounds and expertise? A standardized set of evaluation metrics that is both simple to calculate and intuitive to understand is possibly the best answer to this question. Many metrics for determining model error and even prognostic error have been proposed in the past. Building upon these metrics to update the evaluation of prognostic prediction set metrics, the addition of standardization in the formats and values reported can promote the use of the more intuitively descriptive metrics for a more wide spread understanding and standardization of the field of prognostics beyond it's traditional set of core users. Simply, and accurately conveying the capabilities of any prognostic algorithm is key to gaining acceptance and application in real world, and industry scenarios.

2.1. Standard Model Evaluation Metrics

The most basic of metrics are often overlooked in regards to their usefulness for evaluation prognostic predictions. It is true that in many ways these simple error metrics are inadequate to completely and appropriately characterize the type of information pertinent to prognostic performance. However, when conveying information to potential users of a prognostic model or scientists and engineers from other fields, it is often convenient to at first convey information in a manner both simple and familiar to them.

Many of these type metrics exist, but the Mean Absolute Error (MAE) is a fairly easy metric to compute and in many ways, the most intuitive to understand. Unfortunately, this metric could also be argued to be the least informative about the overall performance of the model compared to those presented in later sections of this paper. Defined in Equation 1, MAE is the average absolute difference between the model prediction P_i and the true Remaining Useful Life (RUL_i) at all times t and for all historic query cases i.

Equation 1

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T_i}\sum_{t}^{T_i}\left(\left|\hat{P}_i(t) - RUL_i(t)\right|\right)$$

In other words, the MAE can be thought of as the average error in prediction for each unit, i, that has run to failure and for each time T, that a prediction is made. The primary attraction of this metric is that it quantifies the average expected value any estimate will be from the true value in real units directly comparable with the system lifetime. Similarly, one could also calculate the standard deviation of the prediction error for a measure of the spread of these errors.

These metrics are useful for comparing separate models built upon similar data, or data from systems with comparable lifetime scales, but give no clear indication of prediction performance without some context to the data. Additionally, these standard formula metrics are inflexible to individual requirements about the specifications of the predictions and can be largely susceptible to outliers.

Although MAE has existed in some implicit form for as long as there have been errors to calculate, the direct usefulness of this intuitive error metric to the evaluation of prognostic predictions performance is severely limited. Conceptually, this error metric provides clear and meaningful indications of the expected error of the total lifetime of a system. Unfortunately as far as prognostics is concerned predictions near the end of a unit or system's lifetime are much more critical than those near the beginning of life. The remaining metrics introduced and described in this work help to overcome and fill in the gaps left by MAE and similar standard metrics.

2.2. Traditional Hierarchical Based Metrics

Saxena et al proposed a hierarchical system of prognostic evaluation, which includes four primary metrics that rely on one another to provide meaning [Saxena et al 2009]. The hierarchy includes in order: the Prognostic Horizon, the Alpha – Lambda Performance, the Relative Accuracy, and Convergence. Briefly, these metrics describe in order, the first instance a prediction becomes within acceptable bounds, if predictions remain in the acceptable bounds at a given time, one minus the percent difference of the prediction to the true value, and finally how quickly the predictions arrive at the final answer.

This system is a good step in establishing a coherent and consistent method for evaluating the performance of a prognostic model. However, these metrics rely on several case specific terms and concepts that do not always lend well to conveying performance to individuals not previously well versed in their application.

As presented, the hierarchy is largely self-reliant. The groups of metrics must be evaluated together, the interpretation of the results from one test have effect on others. For example, a model could have an early lifetime Prognostic Horizon, but this is only truly useful if the model also passes all the Alpha-Lambda tests at every subsequent time instance. Similarly, a rapid convergence should be coupled with a good Relative Accuracy. Understanding the passing of these tests and their significance requires an understanding of the relationships between the metrics that may not be instantly intuitive to experts of other fields.

Also, some of these metrics are not presented in a scalar value manner, making it difficult to assign an overall quantitative value of prediction quality. When presenting the results of a model analysis to prospective users, often a simple and intuitive scalar value comparison is much cleaner and easier to understand than a series of mixed visually and numerically descriptive values. In other words, for certain audiences the hierarchy may unintentionally obscure model evaluation when trying to compare separate models.

In the papers presented by Saxena, the metrics are used to evaluate the prognostic estimations of a single failed unit query case. These metrics each take into consideration only a single query case and only report aspects of that case. The obvious extension of this is to create an average of these individual query based metrics over a large set of query cases. However, this may not always translate well, particularly given the interdependency of the interpretation of the metrics as described above. Metrics built to collectively analyze a group predictions across several failed query cases can help to avoid such skewing of the reported values.

Group based metrics can also help to better estimate a level of uncertainty associated with each predictive model under evaluation. Saxena et al suggest methods for incorporating singular case uncertainties into their metric evaluations, but do not suggest a simple, effective way to propagate these uncertainties [Saxena 2010]. Other recent works have also focused on the evaluation of uncertainty in regards to prognostics. Many interesting ideas concerning both the quantification and evaluation of uncertainty and uncertainty-based metrics have been presented [Orchard et al. 2008], [Leao et al. 2010]. Metrics presented in later sections seek to provide intuitive estimations on model uncertainty based on the set of estimations themselves.

Additional considerations about the interpretation and presentation of some of the metrics should also be mentioned. Prognostic Horizon was originally designed to report the first instance in life where predictions fall within a certain bound, regardless of if the predictions later leave that bound. In later work, Saxena suggests corrections to this by allowing the user to instead quantify the last time it falls in bound without going back out [Saxena 2009]. This practice makes much more sense and should become standard, but again because Prognostic Horizon is calculated over a series of individual cases, there is no standard way to define the value for a set of prediction cases. Should the average value be reported, or would a minimum or maximum be more representative?

Similarly, the convergence metric has the potential to give the same quantification of convergence for vastly different evolutions in the predictions, potentially misleading any blind interpretation of the metric. For example, a prediction set that contains a large outlier early in life (which may be considered trivial) followed quickly by consistent near correct values can show the exact same convergence as one without an early outlier the never quite settles on a consistent prediction value, depending on the application these could be effectively very different results. This work seeks to build upon the initial successes of these metrics, by creating and presenting metrics and methods that are more easily interpreted on a common scale without need for intense understanding of the methods behind them.

2.3. Additional Quantitative Evaluations

Other works have also tried to build upon or propose other standardized metrics. Some of these works, taking a cue from the fields of meteorology and climatology have adapted the concepts of “value” and “skill into the prognostic predictions evaluation vernacular [Tang et al 2011]. Skill, simply put, is the percent improvement of any singular evaluation metric of one prediction set versus some reference prediction set. This can be convenient as a concept for comparing two different models, but provides no additional information not obtained for the original metric itself.

Conversely the concept of “value” is a quantitative metric that can directly be used to evaluate a prediction set. Value is the total set of some user-defined error versus consequence values for a particular application. This allows a user to capture important aspects of low probability but high cost errors that may be of particular interest, such as extreme early life failures. This is very useful for high level decision making and internal review processes; however it lacks the intuitive ability to be conveyed without some form of reference context for the associated costs, whether they be in safety related hours, repair/downtime costs, or some other arbitrary unit. The value of a system is a wonderful internally created metric and can be used to great effect when properly applied and calculated in the industrial setting. Unfortunately, "value" does not translate well across different systems and industries. A standardized prognostic evaluation system should be expected to be instantly interpretable by people of many different backgrounds. The work presented below proposes solutions to this and other problems inherent in the standard set of prognostic evaluation metrics.

3. Proposed Updated Prognostic Prediction Metrics

To promote the wide spread usage of a set of standardized evaluation metrics for prognostic predictions, this work presents set of prognostic prediction evaluation metrics that are designed to be both intuitive and informative to users and reviewers with various backgrounds and levels of expertise. These metrics are also designed to be evaluated on and capture pertinent aspects of entire sets of prognostic predictions over many query cases. Each metric captures key aspects of accuracy, precision, and timeliness. For any prediction, there is both an expected error and an associated uncertainty, these metrics help to report the evolution of these values with special regards to the importance of the relative lifetime of the failed system or equipment, also referred to as query units.

3.1. Weighted Error Bias

The Weighted Error Bias (WEB) is the first of the lifetime percentage based metrics. WEB, as defined in Equation 2, is a measure indicating the effective bias in all predictions as a percentage of total unit lifetime.

Equation 2

$$WEB = \frac{100}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} w_i(t) * \frac{\left(\hat{P}_i(t) - RUL_i(t)\right)}{TotalUnitLifeTime_i}$$

where $w_i(t)$ is the importance weighting for unit i at time t. In this equation, negative values indicate that the prognostic predictions tend to be lower than the true RUL where positive means the opposite. Additional metrics, such as the Weighted Prediction Spread (WPS) presented below, can be combined with the WEB help to capture the average absolute deviation or uncertainty of a prediction set.

From this equation, it becomes evident that WEB is very similar to MAE except in two important respects. First, it is tallied and reported as a percentage of the total lifetime of the individual failed query unit, i. This allows for the intuitive inspection of the performance of a series of predictions without the need for some contextual setting. A model whose predictions yield a 10%WEB would be expected to be better than one with a 25%WEB regardless of the systems, equipment, or time scales involved. This also has the added benefit of implicitly scaling the errors such that similar deviations from the true Remaining Useful Life (RUL) values for short-lived components would be weighed heavier than those in longer-lived units, even within the same historic data set. This is intuitively important, as an error of 20 time cycles is less important if the unit in question survives 300 cycles as opposed to if it only survives 100 cycles.

The second difference is in the explicit importance weighting, w_i, of the different errors based on their time in the lifecycle of the historic unit. This importance weighting can easily be tailored to the specific needs or desires of the end user, but in most cases an emphasis on the end of lifetime is the most meaningful towards prognostic predictions. A 10% error near the beginning of unit life when there is 85% of life remaining gives plenty of time to act an take corrective actions, where a 10% error with only 5% of life remaining could result in an unexpected failure if the unit were expected to life through those remaining cycles. A weighting function that accurately reflects this end of life importance is the Gaussian Kernel Function with a mean value set to the lifetime of the unit and a standard deviation, or bandwidth, set to 50% of that lifetime. Although this metric is built with weightings in mind, a weighting function equal to 1/T for all t can easily turn this metric into a simple average of percent difference between the true and estimated values. For this and all weighted metric, comparisons between algorithms using these metrics would only be meaningful if standard weighting functions are used. Additional work and investigation into what the most appropriate standardized weighting function could prove beneficial. However, regardless of the weighting function, the standardized scaling of the metrics can help it be more relatable to generic audiences.

The optimal value for the WEB is zero, indicating that the average prediction value is centered on the true RUL. Positive and negative values simply express the direction of the bias, otherwise this metric can be presented as a representation of averaged percent error, a concept that is widely utilized and accessible to many academic and industry backgrounds. The weighting function can be tailored to any specific need or application, but the fundamental metric remains an easily interpreted percentage of system lifetime.

3.2. Percent Error Value Binning

The final three prognostic prediction performance metrics rely on estimating or inferring the uncertainty of prognostic predictions throughout the total lifetime of a query unit. In order to do this effectively, the 95%

confidence interval (or some similar level of confidence interval) needs to be calculated at various points throughout the unit lifetime. One of the more straight forward methods for doing this is to create a set of bins evenly divided between 0 and 100% of system lifetime, and placing each calculated percent error in the bin corresponding to the true percent of unit life corresponding to that error. In other words, first calculate the percent error for a given historic prediction, $P_i(t)$, such that the percent error is the difference between the predicted RUL and the actual RUL divided by the query unit ,i's, total lifetime.

Equation 3

$$\%Er = 100 * \frac{\hat{P}_i(t) - RUL_i(t)}{TotalLifeTime_i}$$

Next note the corresponding percentage of actual lifetime (POL), defined by the current time, t, divided by the current unit's total lifetime. Finally place the calculated percent error into the POL bin whose edges, B, are defined as: $B_{LOWER} < POL_i(t) < B_{UPPER}$

Repeat for all historic predictions over all query cases, placing them in to the same series of corresponding bins. Converting the numbers into percentages allows for the direct comparison and inclusion of these similarly located values with proper importance weightings applied as based on their lifetime.

Once this series of regular serial bins is populated, a 95% confidence interval around the mean value can be calculated from the 2.5% and 97.5% percentiles of the error set for each bin. Much like the weightings presented with the metrics presented in this paper, these percentages can be altered to suit the specific application requirements. Additionally, the expected value for each individual bin can be calculated, creating an expected error bias that maps throughout the lifetime of a unit as a more detailed representation of the WEB if such is required. This binning is primarily an intermediary form for the metrics presented in this paper, but as will be shown later, it can also be used to create clear visualizations of the evolution of predictions and how they relate to the true values of RUL. Visualizations such as these can be a great aid in communicating a prediction algorithm's performance to an audience not intimately familiar with the algorithm or system in question.

3.3. Weighted Prediction Spread

Uncertainty estimations, though not always straightforward, are a crucial part of evaluating any prediction value. Thus it follows that the quality of any prediction model should also be defined by its' associated uncertainty. Additionally, much like the model prediction error and bias, not all points during the lifetime of the query system should necessarily be treated with equal importance. The predictions of Remaining Useful Life (RUL) made by a model are typically more important near the end of the system's life than they are at the beginning of life, as near the beginning of life there is comparatively much more time to react and compensate, or mitigate any impending faults or failure inferred from the prognostic model.

The spread of the model predictions at various points in life are an important factor in the total considerations of the uncertainty of a series of predictions. The prediction spread for each binned point of system life, is calculated as the difference between the upper and lower bounds of the corresponding 95% confidence intervals from the binned error values discussed previously. Using the same importance weighting function as the Weighted Error Bias (WEB), the Weighted Prediction Spread (WPS) can be defined as by Equation 4.

Equation 4

$$WPS = 100 * \frac{\sum_{bi=1}^{\#Bins} W_{bi} * CI_{bi}}{\sum_{bi=1}^{\#Bins} W_{bi}}$$

In this equation, the weighting function is based on the center value for each reference bin, such that each bin importance weighting, W_{bi}, is defined by the Gaussian kernel in Equation 5.

Equation 5

$$W_{bi} = \exp\left(-\left(\frac{Bin_{bi} - 100\%}{50\%}\right)^2\right)$$

Notice that the typical normalization factor associated with Gaussian kernels is rendered unnecessary due to the inherent normalization factor included in the definition of WUS. Although a kernel bandwidth of 50% is shown, other bandwidths or even a uniform weighting function can easily be substituted to accommodate specific needs. All the factors and values associated in the metrics based on the binned interval error values are listed and manipulated as percentages allowing for quick intuitive evaluation of the effective important uncertainty of any given prediction set.

With this metric, a 0% WPS alone would seem to indicate absolute certainty in all predicted values, but this may be misleading. In fact, all this would indicate is that all predictions made are exactly the same based exclusively on the percent RUL of the system in question. This is why uncertainty is comprised of both a spread and a bias. The WPS metric can be used in conjunction with the WEB to infer the level of model uncertainty according to the equation: $Uncer \approx \sqrt{WPS + WEB^2}$. This modification of the traditional equation for analytic uncertainty allows for more flexibility in defining what an appropriate value of the spread should be.

Another useful criteria to think of is if the predictions do in fact have enough spread to cover the true RUL (i.e. WPS>=WEB). A more explicit and useful metric evaluating this coverage is the Confidence Interval Coverage (CIC) should also be calculated, and is discussed in the next section.

3.4. Confidence Interval Coverage

Another important indication of the quality of a prediction set generated by any model is whether or not the confidence interval of the prediction spread covers the true Remaining Useful Life (RUL). This effectively incorporates information relating to both the error bias and the error variance at given points in life. This metric is simply defined by the total percentage of binned error sets whose 95% confidence interval contains the true RUL. This is more rigorously defined in Equation 6.

Equation 6

$$CIC = 100 * \frac{\sum_{bi=1} \%RUL_{bi} \in B_{bi}}{\# Bins}$$

This equation is interpreted as the sum number of true percent RUL values that are contained within their corresponding error bin set, and divided by the total number of bins and multiplied by 100 to convert to a percentage. This additional metric verifies the total accuracy of the prediction set. An optimal coverage of 100% shows that the true value of any prediction is contained within the prediction spread or approximate confidence interval of the prognostic model's predictions. This when coupled with the previously detailed metrics gives a solid expectation of the accuracy and expected effective error over the total of system life predictions. The final vital element not conveyed by these metrics is the explicit end of life accuracy and precision. The Confidence Convergence Horizon fills this void.

3.5. Confidence Convergence Horizon

This final standalone metric captures and quantifies the end of life quality of both the precision and accuracy of a prediction set. A 10% Confidence Convergence Horizon (CCH), or simply the Convergence Horizon (CH), identifies the percentage of system Remaining Useful Life (RUL) beyond which, all prediction confidence intervals are both less than 10% of the total system life and contain the true RUL. In other words, the CCH identifies a RUL prediction value that once reached, it and all remaining predictions of RUL can trusted to be no more than 10% from the true RUL 95% of the time (assuming a 95% confidence interval was selected as described above). Obviously a CCH of 100% would be optimal, showing that all predictions within the query set are within less than 10% of the true values. Much like the other metrics, the percentage of this metric can be adjusted to suit the specific needs and requirements of any particular application.

Although this seems to be a rather stringent criterion to meet, it nonetheless, is very important. This horizon is a quick and intuitive identifier of the region of most confidence for a particular prediction set. Unfortunately, like any single descriptive metric, the CCH has the potential to be misleading if it is not considered along with the other metrics defined in this section. As an example, consider a model which predicts the RUL of a system within 10% during most of the system life, but due to an artifact of the data, exhibits an 11% bias at the very end of life. This model would produce a CCH of 0% as there is no point in time which you can trust all following predictions to be less than 10%. This does not however mean that the model produces unusable or even inaccurate results.

Information from each of the listed metrics contains and expresses vital information required to develop a full understanding of the models performance, but when relating to potential users of an algorithm it is often convenient to assign a single quantitative value of "goodness" to a particular model and prediction set. Described in the following section is a method for developing such a single metric.

3.6. Total Score Metric

There has been proposed a sort of hierarchical ranking of some of the previously developed metrics [Saxena 2009]. To some degree, this work is able to eliminate the explicit need for this hierarchical system and in its place supplies a single aggregate scoring metric to rank the overall performance of a particular prognostic model's output predictions. Of the metrics detailed in this paper, four in particular can be merged to give a singular quantitative value of "goodness" for a prognostic model prediction set. These metrics, Weighted Error Bias (WEB), Weighted Prediction Spread (WPS), Confidence Interval Coverage (CIC), and the Confidence Convergence Horizon (CCH) each detail a particular yet vital aspect of the total historic prediction produced by a given model. With this in mind, and given that each of these metrics have been constructed to be listed in similar units of percent Remaining Useful Life (%RUL), a simple composite of these metrics can yield a meaningful, accessible, and direct measure of the quality of a model prediction set. Equation 7 can easily be applied for quick quantitative comparison of multiple models' prediction sets.

Equation 7

$$TotalScore = \vec{N} * \begin{bmatrix} 100 - |WEB| \\ 100 - WPS \\ CIC \\ CCH \end{bmatrix}$$

Note that in this equation, both the absolute value of the WEB and the WPS are subtracted from 100 to reflect that the minimums of these values are the desired quantities. The original WEB metric can be negative to indicate direction of bias, but when combining into an overall score, it is the absolute value that is of more interest. N is any normalized vector weighting the importance of the four metrics. For both simplicity, and intuitive interpretation of the resulting number, a simple average of the four modified metrics can be taken, (mathematically this results from a vector of [.25 .25 .25 .25]). This combined metric can easily be used to present the performance of any predictive model out of a perfect score of 100%. Much like other standardized academic testing, this ideal score is ranked based on ideal performance. For nearly all real systems, 100% accurate predictions 100% of the time is essentially impossible, but this still can help to provide an intuitive ranking system familiar to a wide audience. Some of the model metrics contain similar information, this is not useless redundancy, but instead reflects the increased importance of these aspects when the metrics are combined. For example, if a set of model predictions exhibit 0% CIC, that prediction set would also by definition exhibit a 0% CCH. Coverage of the correct RUL within a confidence interval is one of the most important criteria any prognostic model should meet, so with the standard weighting set, the best total score the model could produce would be less than 50%, reflecting that the model has never produced correct answer.

4. Prediction Metric Example Cases Studies

To help further clarify and explain the prediction metrics, consider a standard pump and motor system with a mean failure time of about 275 operating hours with two common modes of failure with different mean failure times. Three separate simulated models were built to predict the Remaining Useful Life (RUL) of these motors. The first is based strictly on statistical conditional time based probability of failure. The second two are built to simulate more effects based modeling types. In order to compare the three models, each one uses a set of 100 predictions about similar sets of query cases and has the metrics detailed above applied to those prediction sets.

Shown in Figure 1, the Model 1 prediction set for all 100 cases completely overlay one another. This is expected and due to the fact that this model's output is based exclusively on the current lifetime of the queried system.

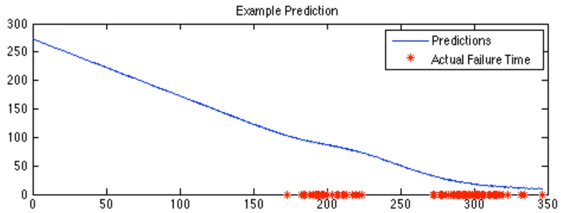


Figure 1 - Model 1 RUL Predictions

Despite the fact that each of the predictions for each individual case are all exactly the same, they represent varying percentages based on the true queried system's lifetime. This is accounted for in the calculations of the performance metrics shown in Figure 2.

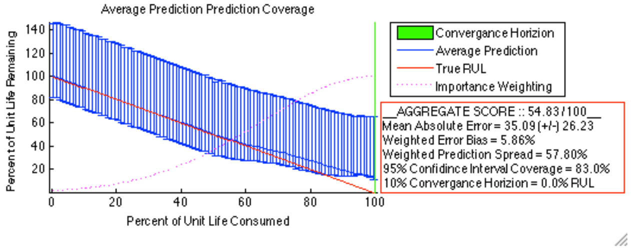


Figure 2 - Model 1 Prediction Performance Metrics

The most intuitive and easily understood metric on this figure is the Mean Absolute Error (MAE), listed as 35.1 hours with an associated standard deviation of 26.2 hours. Considering that the average lifetime is 275 hours, these numbers present values which would easily allow for the rescheduling of duty cycles to accommodate maintenance or similar mitigating actions before the units would be expected to fail. The MAE gives a good basic understanding of how much error to expect out of the model, and is good for comparing models that are run against the same data set. However, the three example models presented here are run with differing query data sets. The sets are taken from similar sets of pump systems, but the individual units and their true total lifetimes are different. Although MAE could be used to compare these models and prediction sets because time units and expected average lifetimes are the same, the percentage-based metrics are more appropriate and generally informative.

The most prominent prediction evaluation tool in this figure is the binned error average estimate and their associated 95% confidence intervals represented by the blue error bars. This contains the most total and useful information about the prediction set. These bind also are used to represent the other metrics as they evolve through time. The solid blue line is the bias at a given time; the error-bars represent the spread; the total number of bars which contain the red (true) RUL represent the confidence coverage. Finally, the Convergence Horizon will be represented as a green box in the following figures, but is not present in this on due to CH being equal to 0.

From this chart it is obvious that early in life the model predicts the correct percentage of RUL on average, but also has high uncertainty, meaning it may in fact never predict the exact true RUL for a particular unit. This inference is confirmed by examination of the end of life binned error as the average model prediction value departs from the true RUL line at around 62% of life consumed (38% RUL) and loses even the 95% prediction interval coverage at around 85% of life consumed (15% RUL). Because of the fact that this is a strictly time based model, this helps to confirm that the model is unable precisely predict individual systems' RUL, instead only calculating the average RUL over all historic systems. Although this chart of binned error is useful and contains a wealth of information, it does require some degree of examination and analysis in order to compare different model sets. The other percentage based prediction metrics provide that analysis.

The effective bias for this model, as calculated by the Weighted Error Bias (WEB) from Equation 2 is 5.86%. Again, this can be seen in the binned error analysis as the average estimation line begins to deviate from the true RUL line particularly near the end of life. For this system, that means that there is an effective average bias of about 16 hours on average, but this does not mean that the expected error is 16 hours. This value, as well as the Weighted Prediction Spread (WPS), is considered an effective value because of their applied weighting function shown in the figure as a magenta dotted line, which allows them to be more effective at ranking the predictions. If for some reason, the more literal average values are needed, the same equations and metrics can be applied with a simple adjustment of the weighting function. This prediction set's WPS is listed as 58.07% of life, reflecting the fact that there is a considerable amount of uncertainty associated with the predictions.

The final two metrics listed are the Confidence Interval Coverage (CIC) and the Convergence Horizon (CH). Reported at 83% and 0% respectively, these indicate that although the model uncertainty covers the true RUL 83% of the time, it never continuously falls within 10 of that true value towards the end of the unit's life.

All these metrics can be combined according to Equation 7 in order to give this model's prediction set a total ranking of 54.83% out of a possible total score of 100%. This should not be read as an indication that the model's total accuracy is around 50% or that only 50% of the model's estimations are trust worthy. Instead this metric shows a quantitative evaluation of the model's performance for this prediction set. It is a quick and relatable evaluation of the model's "goodness" which can easily be used to compare against other models, or other prediction sets. For example, if Model 1 is compared to Model 2 shown in Figure 3, one can quickly see that Model 2 has a total performance score of 75.02%, much better than Model 1's 54.83%.

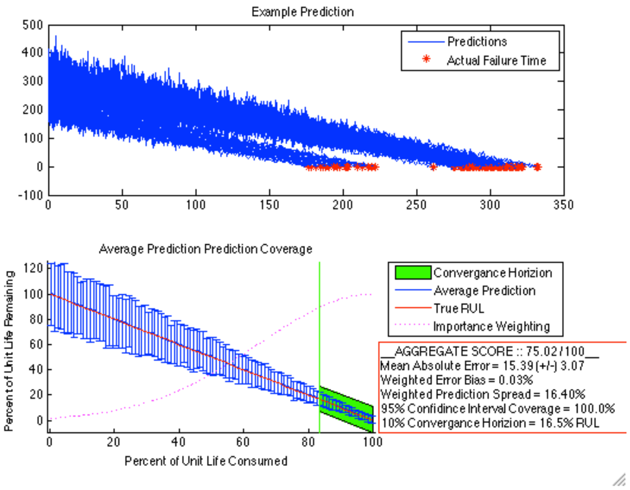


Figure 3 - Model 2 Predictions and Metrics Evaluation

Looking at the individual metrics, it becomes clear why this model is ranked better. First, it has 100% CIC with a 16.5% CH meaning that not only is the model more accurate overall, but it also shows that the accuracy improves near end of life. Next the effective prediction spread is 16.4% of life, much lower than Model 1's WPS. Finally Model 2 has virtually 0% effective bias, meaning that all the predictions are centered on the true RUL.

Clearly, these metrics give a quick, effective, and qualitative method for comparing two different models, and if that were the only end goal the analysis could stop there. However, if there is opportunity to change and improve the models which created the prediction sets, then the scalar metrics alone may not give the complete picture. Consider the prediction set developed by Model 3 in Figure 4.

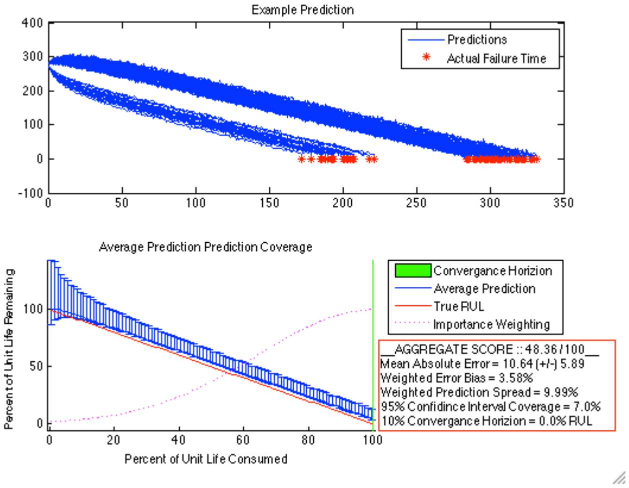


Figure 4 - Model 3 Predictions and Metrics Evaluation

Model 3 has a total performance score of 48.36%, indicating its' performance is worse than either of the two previous models. In fact, the only metrics which it out performs both of the other models are MAE and the WPS. Unfortunately, these alone would not necessarily merit further investigations into the development of this model. However, when the total binned prediction value map is investigated, it becomes instantly clear that by removing a small bias in this model, these predictions would be expected to out perform both of the previous models. This same conclusion could be inferred from the scalar metrics, but a graphical examination of the binned values map is both more expedient and informative.

5. Summary and Conclusion

The scalar metrics presented in this work help to provide clear and concise evaluations of the performance of prognostic models in a manner easily accessible and largely intuitive to audiences with various backgrounds and expertise. In order to demonstrate and visualize the underlying meanings of each of the metrics, three separate sets of predictions made from three separate simulated prognostic models were compared. From the results listed in Table 1 it is clear that Model 2 is the best performing model by a large margin.

Table 1 – Summary of Model Comparison Results

	Total Score	MAE	WEB	WPS	CIC	CCH
M1	54.83%	35.09 Hrs	5.06%	57.80%	83.0%	0%
M2	**75.02%**	15.39 Hrs	0.03%	16.40%	100%	16.5%
M3	46.36%	10.64 Hrs	3.58%	9.99%	7.0%	0%

Further, Model 3 shows great potential for improvement via a simple bias removal as can be inferred from the low Weighted Prediction Spread (WPS) coupled with the results of the binned prediction value map. A quick summary of each metric is listed below in Table 2.

Table 2 - Metrics Summary

Metric Name	Quality Aspect Reflected	Units
Mean Absolute Error (MAE)	Precision *Average distance from true value*	Real Time Units
Weighted Error Bias (WEB)	Timely Precision *Scaled expected distance from true value*	Percent of Unit Life *Weighted by Lifetime Importance*
Weighted Prediction Spread (WPS)	Timely Accuracy *Scaled uncertainty estimate associated with each prediction*	Percent of Unit Life *Weighted by Lifetime Importance*
Confidence Interval Coverage (CIC)	Accuracy *How often the estimated uncertainty contains the true value*	Percent of Unit Life
Confidence Convergence Horizon (CCH)	Timely Accuracy & Precision *What part of life can all remaining estimates be trusted to within 10%*	Percent of Unit Remaining Useful Life
Binned Prediction Value Map	Timely Accuracy & Precision *Detailed visualization of the evolution of the prognostic predictions. Used to calculate other metrics*	Percent of Unit Life

These novel metrics build upon natural aspects of the prediction data itself to create meaningful and intuitive representations of performance. The goal of this work is to learn from previously introduced metrics and create a set of generic metrics that can be widely used and understood in both academic and industrial settings. All of the metrics detailed in this work can be easily calculated and widely applied and interpreted across many cases allowing for un-obscured, evaluation of predictions from a wide variety of algorithms and methodologies. The balance between case specific adaptability and overall standardization is an area of continual interest and research. This work seeks to provide a set of metrics that provide a level of both in a manner that is accessible and relatable to a wide audience to help promote investigation and collaboration on prognostic projects across many fields.

References

Banks, J., J.Merenich. "Cost Benefit Analysis for Asset Health Management Technology". Reliability and Maintainablity Symposium (RAMS), Orlando, Florida. 2007

Coble, Jamie, "Merging Data Sources to Predict Remaining Useful Life – An Automated Method to Identify Prognostic Parameters," Doctorial Dissertation, University of Tennessee, Knoxville TN. 2010

Leao, B.P., J.P.P.Gomes, R.K.H.Galvaro, and T.Yoneyama. "How to Tell the Good from The Bad in Failure Prognostics". IEEE Aerospace Conference Proceedings. 2010

Orchard, M., G.Kacprzynski, K.Goebel, B.Saha, and G.Vachtservanos. "Advances in Uncertainty Representation and Management for Particle Filtering Applied to Prognostics". International Conference on Prognostics and Health Management, 2008.

Saxena, Abhinav, Jose Celaya, E. Balaban, B. Saha, S. Saha, and K. Goebel, "Metrics for evaluating performance of prognostic techniques”. International Conference on Prognostics and Health Management (PHM08), Denver CO, pp. 1-17, 2008

Saxena, Abhinav, Jose Celaya, Bhaskar Saha, Sankalita Saha, and Kai Goebel. "On Applying the Prognostic Performance Metrics." Annual Conference of the Prognostics and Health Management Society (2009)

Saxena, Abhinav, Jose Celaya, Bhaskar Saha, Sankalita Saha, and Kai Goebel. “Metrics for Offline Evaluation of Prognostic Performance”. International Journal of Prognostics and Health Management. ISSN 2153-2648, 2010 001. April 2010.

Tang, Liang, Marcos E.Orchard, Kai Gobel, George Vachtevanos, “Novel Metrics for the Verification and Validation of Prognostic Algorithms”. Aerospace Conference 2011 IEEE, Big Sky, MT. 5 -12 March 2011.

Uckun,S., K.Goebel, and P.J.F.Lucus. “Standardizing Research Methods for Prognostics. International Conference on Prognostics and Health Management (PHM08). Denver CO. 2008

PHM for Astronauts – A New Application

Alexandre Popov[1], Wolfgang Fink[2], and Andrew Hess[3]

[1]*Canadian Space Agency, Saint-Hubert, QC, J3Y 8Y9, Canada*
alexandre.popov@asc-csa.gc.ca

[2]*Visual and Autonomous Exploration Systems Research Laboratory, Division of Physics, Mathematics and Astronomy, California Institute of Technology, Pasadena, CA 91125, USA*
wfink@autonomy.caltech.edu

[3]*The Hess PHM Group, Inc., Rockville, MD, 20850, USA*
andrew_hess@comcast.net

ABSTRACT

This paper introduces a concept and approach on bridging Prognostics and Health Management (PHM), an engineering discipline, to Space Medicine (SM) in order to mitigate the Human Health and Performance (HH&P) risks of exploration-class space missions by focusing on efforts to reduce countermeasure mass and volume and drive the risks down to an acceptable level. The paper also discusses main risks of missions such as autonomous medical care risk (i.e., mission and long-term health risk due to the inability to provide adequate medical care throughout the mission) and Behavioral Health and Performance (BH&P) risk (i.e., mission and long-term behavioral health risk). The main objective of the HH&P technologies being developed for exploration-class missions is to maintain the health of the crew and support optimal and sustained performance throughout the duration of a mission. A PHM-based technology solution augmented with predictive diagnostics capability could be the one that meets the main objective. In discussing the similarities of and differences between the PHM and SM domains, the paper explores available solutions on crew health maintenance in terms of predictive diagnostics providing early and actionable real-time warnings of impending health problems that otherwise would have gone undetected. The paper discusses the use of PHM principles and techniques with data mining capabilities to assess the value of Electronic Health Records (EHR) augmented with real-time monitoring of data for accurate predictive diagnostics on manned space exploration programs. The proposed technology concept with predictive diagnostics capability and a pilot implementation of the technology on the International Space Station (ISS) includes evaluation and augmented research/testing of the technology, which will regularly and efficiently provide advancements during the development phases.

1. INTRODUCTION

Long duration missions present numerous risks to crew health and performance. The international space community is actively studying these effects and possible mitigation techniques, but much work remains to be done. As such the space community and space agencies are increasingly cooperating to enable timely answers in support of exploration mission needs (2013 "Global Exploration Roadmap" report). This is very important because with a common understanding of risks and effective mitigation approaches, the space community has the opportunity to leverage investments in the research and technology development to mitigate risks.

Crew health and performance are critical to successful human exploration. Long-duration missions bring numerous risks that must be understood and mitigated in order to keep astronauts healthy, rather than treat a diagnosed health disorder. Crewed missions venturing beyond Low Earth Orbit (LEO) will require technology solutions for crew health care to address physiological, psychological, performance, and other needs in-situ, e.g., self-sufficiency, as an emergency or quick-return option will not be feasible. Therefore, onboard capabilities that would allow for early self-diagnosis of impending health issues, and autonomous identification of proper responses on negative trends to keep astronauts healthy are critical. With the absence of real-time medical ground support, personal health-tracking tools for health monitoring, health risk assessment and management

A. Popov et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

are required for any crew member to predict her/his future health condition if no preventive measures are taken.

Per the 2013 "Global Exploration Roadmap" report, published by the International Space Exploration Coordination Group (ISECG) in August 2013, a key supporting objective to develop exploration technologies and capabilities is the following: test concepts, approaches, countermeasures and techniques to maintain crew health and performance. This paper suggests a concept using PHM-based technologies, such as real-time health monitoring and condition-based health maintenance in terms of predictive diagnostics. Discussing similarities of PHM vs. Space Medicine, the paper introduces a predictive diagnostics concept for crew health maintenance. Furthermore, it explores PHM solutions based on real-time monitoring, which could be applicable to crew health risk assessment and management.

While the International Space Station is an excellent platform and currently the only "test bed" on which to prepare for future manned exploration missions, the exploration beyond low-Earth orbit will require a new generation of capabilities and systems, which build on existing capabilities and incorporate technologies yet to be developed.

It becomes necessary to develop alternative, evidence-based, effective methods and tools to predict and prevent health problems in a timely manner, rather than to follow reactive approaches, which are inherent to conventional medicine, but largely prohibitive in the operational environment of space because of lack of accessibility of health problem resolutions.

Interdisciplinary research is underway to develop computer-based, self-diagnosis and self-directed treatment programs for astronauts to autonomously predict, prevent, and manage potential health problems (e.g., Fink, Clark, Reisman, and Tarbell, 2013). In the 2010 Interim Report "Life and Physical Sciences Research for a New Era of Space Exploration" the National Research Council emphasizes a priority on bringing the programs to the required technology readiness level (TRL), i.e., corresponding to a representative laboratory environment for exploration-class missions (TRL 6 per NASA designation), so that they can be systematically evaluated in comparative treatment outcome studies.

2. Understanding the Technology

Prognostics and Health Management (PHM) is an engineering discipline that focuses on the fundamental principles of system failures in an attempt to predict when they might fail, and links the principles to system life cycle management. Sometimes this engineering discipline is also referred to as System Health Management (SHM) (Uckun, Goebel, and Lucas, 2008). In recent years, PHM has emerged as a key enabling technology to provide early warning of failure and assess the potential for life extension, thereby leading to potential monetary and downtime savings.

Prognostics is about predicting the future performance of a component by assessing the extent of deviation or degradation of a system from its expected normal operating conditions. The science of prognostics is based on the analysis of failure modes, detection of early signs of wear and aging, and fault conditions. Technical approaches to building models in prognostics can be categorized broadly into data-driven approaches, model-based approaches, and hybrid approaches.

As an engineering discipline PHM includes the following:

- Health monitoring (i.e., monitoring the extent of degradation or deviation from an expected normal condition);
- Methods for in-situ monitoring;
- Sensors for prognostics;
- Data collection, pre-processing, reduction, and feature extraction;
- Methods for identifying and analyzing precursors based on failure mechanisms;
- Damage assessment;
- Anomaly detection;
- Diagnostics;
- Prognostics;
- Risk and uncertainty analysis;
- Software tools for diagnostics and prognostics.

PHM concept implementation is now a required design feature for space systems (Uckun et al., 2008). Space systems have built-in PHM elements such as failure tracking. In the future, PHM will enable systems to assess their own real-time performance (self-cognizant health management and diagnostics) under actual usage conditions and adaptively enhance life cycle sustainment with risk-mitigation actions.

Human health is one of the application areas of PHM, while health records and health care delivery are going digital (see, e.g., Health Information Technology (Health IT): "Policymaking, Regulation, & Strategy" on the HealthIT.gov website of the U.S. Department of Health and Human Services). As multiple intersecting platforms evolve to form a novel operational foundation for health and health care – the digital health utility – the stage is set for fundamental and unprecedented transformation. Progress in computational science, information technology (IT), and biomedical and health research methods have made it possible to foresee the emergence of a learning health system that enables both the seamless and efficient delivery of best care practices and the real-time generation and application of new knowledge (Grossman, Powers, and McGinnis, Rapporteurs and Editors (2011)).

Prognostics as an engineering discipline is focusing on predicting the time at which a system or a component will no longer perform its intended function, while *predictive diagnostics* is built on the powerful foundation of predictive analytics. But whereas predictive analytics and PHM methods are to identify what is going to fail and when a particular element is going to fail, in-flight predictive diagnostics also tells the cause(s) of the failure as well as potential factors contributing to and the priority of the impending failure. Direct, contributing, and root causes as well as priority of the impending failure with a corresponding probability are other notions introduced by predictive diagnostics. That makes predictive diagnostics different from predictive analytics. The terminology is not commonly adopted yet.

Like PHM, predictive diagnostics provides early and actionable real-time warnings of impending health problems that otherwise would have gone undetected. Based upon the differences between real-time health status and predefined normal status, predictive diagnostics detects and isolates abnormal dynamics and negative trends in the context of operating conditions. An underlying concept in predictive diagnostics in space missions is that every crew member is unique. This requires the development of a unique data set ("set of fingerprints") for each individual in a number of areas: medical history, genetic predisposition, recent medical events, baseline health assessments including vital signs in terms of operational (e.g., extra-vehicular activity) and emotional contexts (e.g., anxiety (2010 Interim Report "Life and Physical Sciences Research for a New Era of Space Exploration")).

A PHM-based system augmented with predictive diagnostics capability would be required to perform real-time health assessment followed by evaluating the assessment results against a crew member health baseline, i.e., a health pattern corresponding to a "normal" health state in which the crew member is identified as a physically and mentally healthy person meeting in-flight specific requirements. Based upon the differences between real-time assessment and normal health state, predictive analytics would detect negative trends and isolate abnormal dynamics in the context of the current operational environment.

PHM technologies augmented with predictive diagnostics capability on manned space exploration programs include, but are not limited to the following:

- Proven engineering techniques, data analysis, and statistical methods to astronaut health maintenance in order to translate complex data into accurate knowledge and informed actions;
- Methods for in-situ monitoring of astronaut health using unobtrusive and non-invasive sensors/devices;
- Implementation of telemetry and data processing concepts to improve health care delivery;
- Data-driven approaches, algorithms and models for large-scale health data processing and extraction of features of interest;
- Health damage assessment;
- Identification and analysis of precursors on health compromise;
- Statistical techniques and machine learning methods for diagnostics and prognostics;
- Anomaly detection.

The absence of real-time medical ground support requires a shift in health care delivery on manned exploration-class space programs from a telemedicine paradigm to that of medical autonomy (i.e., onboard health care). It used to be that all the information on crew member health and all the controls were residing with the medical ground support team (Integrated Medical Group (IMG) or MED Ops Team) and on-board health care professional (Crew Medical Officer (CMO)). This paradigm may have to shift to where the consumer, i.e., the crew member, is gathering his/her own supplemental data through various means, and decides whether he/she wants to share these data with the medical ground support team and when. These data are additional to those data the medical ground support team receives on a regular basis as the routine part of the space program. Having accepted the inherent risk of autonomous medical care (2013 "Global Exploration Roadmap" report) the crew should be in control until a disorder symptom is identified or a disease is diagnosed. Given that predictive diagnostics is the key to keep the crew healthy, it appears that in addition to the current responsibilities, which the IMG usually has on space programs, the new role of the ground support team is to provide the crew with more health assessment software applications rather than more pharmaceuticals. Yet, this could/should be largely done at the mission design stage though. The paradigm shift could yield solutions to known issues related to health care delivery on manned space programs, such as underreporting, reluctance to discuss health status, etc.

The technology implementation could bridge PHM with the space medicine domain by introducing proven engineering techniques coupled with advanced information technologies that could help the space medicine community to build scientific- and evidence-based health care delivery in terms of individualized medicine and autonomy paradigms.

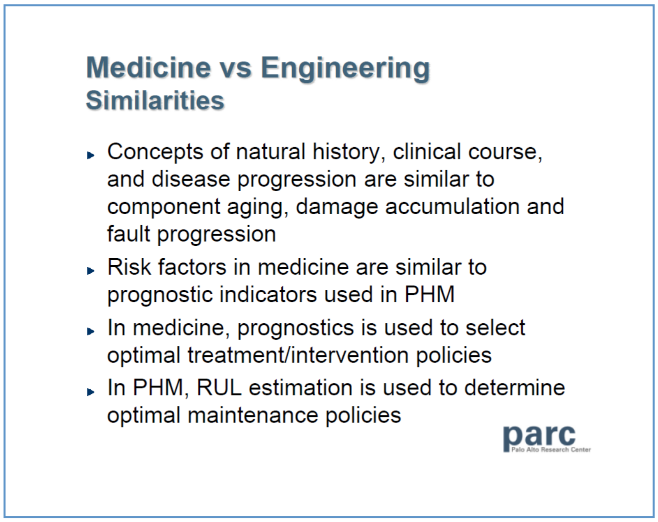
3. Similarities between Space Medicine and Engineering

There has been a growing interest in monitoring the "health" of both the operational environment and astronauts in order to predict failures and provide early warning to avoid health compromise. Here, health is defined as the degree of normal condition. The following are the PHM techniques used in real-time health monitoring:

- Built-in-test (BIT);
- Usage of "canary devices"[1] and/or (bio-) markers;
- Monitoring of and reasoning over failure precursors;
- Modeling of accumulated damage.

All of these techniques could be successfully employed to astronaut health real-time monitoring as well. For example, like certain biomarkers, the "canary devices" (Pecht, 2008), which are usually integrated into a system, have incorporated failure mechanisms that occur first in the embedded device.

Thus, it is possible to make continuously updated predictions based on the actual environmental and operational condition monitoring of astronaut health (see also Figure 1).



Medicine vs Engineering
Similarities

- Concepts of natural history, clinical course, and disease progression are similar to component aging, damage accumulation and fault progression
- Risk factors in medicine are similar to prognostic indicators used in PHM
- In medicine, prognostics is used to select optimal treatment/intervention policies
- In PHM, RUL estimation is used to determine optimal maintenance policies

parc
Palo Alto Research Center

Figure 1. Comparison of Health Care vs. Engineering (Uckun et al., 2008)

Health management in both engineering and medicine domains requires that considerations of the appropriateness of interventions are based on scientific evidence. Given the similarities of the two domains focusing on prognostics, a common scientific foundation for both of the domains could be established. However, to ensure maturity of this foundation, a common language, singular methodology, and benchmarking are required to be implemented. For more details, Uckun et al. (2008) provides an excellent summary on PHM methods and techniques, whereas Lucas and Abu-Hanna (1999) do the same for prognostic methods and techniques in medicine.

It should be stressed though that a fundamental difference exists between components of complex machinery/processes and the human body or organs/processes within. While machine/process components may have well-defined and well-understood failure modes, the failure modes of a human body or organs/processes within are far less predictable: (1) the human body is not a machine, and (2) it is characterized by far more complex (and often unknown) interactions of failure modes. Rapid, unpredicted, and unforeseen changes in the health status of a patient can occur within seconds.

4. Managing Health and Human Performance Risks for Space Exploration: The Value of Predictive Diagnostics

The promise of data-driven decision-making is now being recognized broadly. Decisions that previously were based on guesswork, or on painstakingly constructed models of reality, can now be made based on the data itself. Decision-making in the areas of health and human performance management is not any different.

It is widely believed that the use of the particular data-driven information technology can reduce the cost of healthcare while improving its quality, by making care more preventive and personalized, and by basing it on continuous monitoring.

To understand the value of new technologies a differentiation has to be made between two things that are often confused by analysts: *capabilities* and *functions*. Capabilities are derived from combinations of functions. Functions are the basic tasks or activities that can be performed with a new technology. Broadly speaking, a capability is what can be achieved with the technology, i.e., "what it is for", whereas a function is what the technology does.

Predictive analytics is a new information management approach and set of capabilities for uncovering additional value from health information. Within the health care sector it provides new insights that have the potential to advance personalized care, improve patient outcomes, and avoid unnecessary costs.

Predictive analytics is the process of examining large amounts of data, from a variety of data sources and in different formats, to deliver insights that can enable decisions in real or near real time. Various analytical concepts such as data mining, natural language processing, artificial intelligence (e.g., expert systems), machine learning, and predictive analytics itself can be employed to analyze, contextualize, and visualize the data. The analytical approaches can be employed to recognize inherent patterns, correlations and anomalies, which can be discovered as a result of integrating vast amounts of data from different data sources (e.g., sensor-data fusion). Also, computer-based

[1] Usage of "canary devices" is one of the PHM techniques: an early-warning device derived from the use of a canary bird to detect the presence of poisonous gases in a mineshaft. For example, certain bacteria and microbes could serve as canary devices to detect an impending health issue.

self-diagnosis and self-directed treatment programs could be a solution to known issues, such as underreporting on health challenges. It appears that sharing information with personal devices rather than with MED Ops Team (MEDical Operations Team) or CMO is a preferred option for astronauts.

The variety of predictive diagnostics techniques, which are based on predictive analytics, is usually divided into three categories (Langreth & Waldholz (1999)):

- **Predictive models** look for certain relationships and patterns that usually lead to certain behavior and predict system failures;
- **Descriptive models** aim at creating segmentations and find clusters of data elements with similar characteristics;
- **Decision models** use optimization techniques to predict results of decisions.

In predictive models the outcome of the dependent values could be predicted by determining the explanatory values. Where predictive models focus on a specific event or behavior, descriptive models identify as many different relationships as possible. Decision models, another branch of the predictive analytics, lean particularly heavily on operations research, including areas such as route planning, resource optimization, etc. This classification is very practical, since it provides an immediate understanding of the areas where predictive analytics add value.

The following are data types proposed for digitizing the data as Electronic Health Records (EHR) and using predictive diagnostics to support astronaut health maintenance on space exploration programs:

- Clinical data (up to 80% of health data is unstructured as documents, images, clinical or transcribed notes);
- Publications (clinical research and medical reference material);
- Clinical references (text-based practice guidelines and health product data, i.e., drug information);
- Genomic data (significant amounts of new gene sequencing data) (Langreth & Waldholz (1999));
- Streamed data (health monitoring with handheld and sensor-based wireless or smart devices).

There are many sources of data within the health care sector. However, it is unrealistic to assume that all data can be put to use for predictive diagnostics due to a range of operational and technical challenges (mainly interfacing and incompatibility issues) and privacy considerations.

5. Real-Time Monitoring for Astronaut Health Management

In order to assess the effects of environmental and operational factors on the health status, and to allow early detection of negative trends, real-time health monitoring is required. The ultimate goal of real-time monitoring as an essential component of a predictive capability is its potential for providing meaningful and up-to-date data for detecting trends in astronaut health status during a mission. In this context, “status” should be considered to include the capacity to perform mission-related tasks and the level of health/well-being. The challenge is to provide not only valid and reliable data, but also data sensitive to potentially subtle physiological and neuropsychological deficits caused by stressors. Typical stressors are listed below that can potentially lead to undesirable developments such as overgrowth of certain bacteria, decreasing immune response, anxiety, depression, tension, fatigue, daytime sleepiness, stress-related cardiac arrhythmias, memory impairments, etc. (2010 Interim Report “Life and Physical Sciences Research for a New Era of Space Exploration”):

- Exposure to solar and space radiation;
- Prolonged period of exposure to microgravity;
- Confinement in close, relatively austere quarters;
- Limited contact with family and friends;
- Isolation (small number of crew members);
- Chronically inadequate sleep;
- Work overload;
- Atmospheric composition (e.g., CO_2 concentration);
- Volatile organic compounds;
- Variation in light spectrum;
- Vibration;
- Noise;
- Monotony;
- Environment pollution.

A real-time monitoring approach (2010 Oracle white paper: “Predictive Analytics: Bringing the tools to the data”) is presented in Diagram 1 below.

Real-time monitoring as a predictive capability component is common for both PHM and astronaut health care based on predictive diagnostics. Unlike conventional medicine, which is based on taking “snapshots” (i.e., medical check-ups) to track health status, PHM with a predictive analytics capability takes advantage of analyzing additional information acquired during manned space exploration programs on a real-time basis.

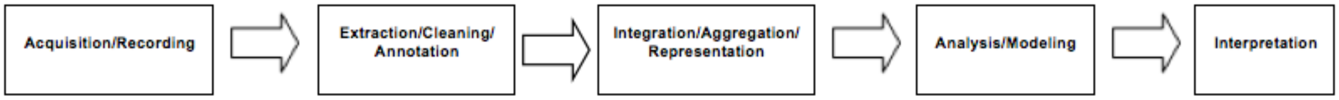

Acquisition/Recording
Extraction/Cleaning/ Annotation
Integration/Aggregation/ Representation
Analysis/Modeling
Interpretation

Diagram 1. Real-time monitoring (conceptual architecture)

6. Discussion & Conclusion

The primary benefit of the successful technology implementation is the ability to successfully achieve affordable human space missions to LEO and beyond (e.g., human settlement on the Moon and Mars). An implementation of the proposed technology with predictive diagnostics capability on the ISS, as a unique human-occupied test platform in space, will directly contribute to the knowledge base and advancements in managing health and human performance risks for space exploration. In addition to research, the ISS provides the capability to validate countermeasures and mitigation strategies. While countermeasures used on the ISS are largely effective at managing health and performance risks, the technology implementation could lead to a better understanding of the risks and to the development of novel countermeasures against these risks. The proposed technology with on-board predictive capability coupled with countermeasures against cardiovascular, musculoskeletal, and neurological or behavioral challenges associated with space flight is critical for human space exploration. Nutritional countermeasures are also essential, given the impact of diet and nutrition on human health both in space and on Earth. In addition, there are other potential factors being investigated, which might predispose individuals to certain changes in the visual system during space flight (e.g., "Longitudinal Study of Astronaut Health" (LSAH) and 2012 NASA Evidence Report "Risk of Spaceflight-Induced Intracranial Hypertension and Vision Alterations"), which could cause problems on future long-duration exploration missions and for which no countermeasures are currently known. However, progress on all these issues must be made before long-term exploration missions can be successful.

Since crew health and performance are primary, critical concerns, the space community and the ISS program should actively take advantage of ISS-based research to extend human space mission durations while ensuring crew health and performance (Popov, 2012). The health risks are significant enough to drive decisions related to planning of exploration missions beyond LEO.

In order to develop a mature PHM-based technology with a predictive capability the following recommendations on further research, detailed in the 2010 Interim Report "Life and Physical Sciences Research for a New Era of Space Exploration", need to be implemented:

- Determination of the mission-specific effects and other relevant stressors, alone and in combination, on the general psychological and physical well-being of an astronaut. Emphasis should be on determining the extent to which such stressors constitute a risk to mission success;
- Development of interventions to prevent, minimize, or reverse deleterious effects during extended missions.

To assess the effects of environmental factors on crew health and to enable early detection of negative trends a real-time monitoring is required. The monitoring challenge is to provide not only valid and reliable data, but also data sensitive to potentially subtle physiological and neuropsychological deficits caused by the stressors.

To build a sustainable human space exploration endeavor that lasts decades, the international space community should maintain a focus on delivering value to the public (2013 "Global Exploration Roadmap" report). The proposed technology concept with predictive diagnostics capability and a pilot implementation of the technology aboard the International Space Station includes evaluation and augmented research/testing of the technology, which will regularly and efficiently provide advancements during the development phases. The pilot implementation could serve as a contribution to the exploration-class mission readiness since it would demonstrate autonomous crew operation capability coupled with a reduced supply chain on health care delivery. Investments in the technology development, with bringing the technology to TRL 6, can lead to improvements in the quality of life here on Earth and create benefits of national and global interest.

History has repeatedly shown that finding ways to meet the challenges of safe and sustainable human space flight results in solutions that are applicable far beyond space flight (2013 "Global Exploration Roadmap" report). It is important to ensure consistent realization and broader dissemination of the benefits generated by the technologies validated on the ISS in order to meet requirements and challenges of exploration-class space missions.

References

Alexander, D. J., Gibson, C. R., Hamilton, D. R., Lee, S. M. C., Mader, T. H., Otto, C., Oubre, C. M., Pass, A. F., Platts, S. H., Scott, J. M., Smith, S. M., Stenger, M. B., Westby, C. M., Zanello, S.B. (2012). Risk of Spaceflight-Induced Intracranial Hypertension and Vision Alterations. *NASA Evidence Report:* http://humanresearchroadmap.nasa.gov/Evidence/reports/VIIP.pdf

Fink, W., Clark, J. B., Reisman, G. E., Tarbell, M. A. (2013). Comprehensive Visual Field Test & Diagnosis System in Support of Astronaut Health and Performance. *IEEE Aerospace Conference Proceedings*, paper #2675, Big Sky, Montana

Grossman, C., Powers, B., and McGinnis, J. M., Rapporteurs and Editors (2011). Digital Infrastructure for the Learning Health System: The Foundation for Continuous Improvement in Health and Health Care – Workshop Series Summary. Roundtable on Value & Science-Driven Health Care; Institute of Medicine, *The National Academies Press*, Washington, D.C.

Health Information Technology (Health IT): Policymaking, Regulation, & Strategy: http://www.healthit.gov/policy-researchers-implementers/meaningful-use. Office of the National Coordinator for Health Information Technology; *U.S. Department of Health and Human Services*

Langreth, R., & Waldholz, M. (1999). New era of personalized medicine: Targeting drugs for each unique genetic profile. *Oncologist* 4(5):426-427

Life and Physical Sciences Research for a New Era of Space Exploration: An Interim Report. National Research Council, *The National Academies Press*, Washington, D.C., 2010

Longitudinal Study of Astronaut Health (LSAH) and the respective LSAH Newsletters referenced therein; http://lsda.jsc.nasa.gov/scripts/mission/miss.cfm?pay_index=317&mis_index=218&program_index=13

Lucas, P. J. F., & Abu-Hanna, A. (1999). Prognostic methods in medicine. *Artificial Intelligence in Medicine*, 15:105-119

Pecht, M. (2008). Prognostics and Health Management of Electronics; *Wiley*

Popov, A. (2012). System Health Management and Space Medicine Predictive Diagnostics. Common Concepts and Approaches. *IEEE Aerospace Conference Proceedings*, Big Sky, Montana

Predictive Analytics: Bringing the tools to the data (2010). An *Oracle white paper*, September 2010

The Global Exploration Roadmap (2013): https://www.globalspaceexploration.org/c/document_library/get_file?uuid=6bdce6a3-1400-4b47-b6ba-3556755273c3&groupId=10812. *International Space Exploration Coordination Group (ISECG)*

Uckun, S., Goebel, K., Lucas, P. J. F. (2008). Standardizing Research Methods for Prognostics. *International Conference on Prognostics and Health Management*, October 2008

Biographies

Alexandre Popov received a M.Sc. in Computerized Control Systems Engineering from Moscow State University of Aerospace Technologies (former Moscow Aviation Institute) in 1983 and M.Sc. in Applied Mathematics from Moscow State University in 1988. As a test-engineer with Tushinsky Machine Building Enterprise (Russia) he had conducted verification tests of navigation and landing systems on the BURAN space program. At a later date he - as a data architect and business analyst – had led software development for mission planning on MIR and International Space Station (ISS) programs at the Energia Rocket Space Corporation (Russia) [1988-1998]. From 2000 to 2003 he served as an advisory member of Engineering with Lockheed Martin Canada working on the ISS program (ISSP) at the Canadian Space Agency (CSA). He joined the Canadian Space Agency as a mission planner in 2003 and has contributed to the ISSP process and data integration effort. From 2011 to 2012 he led CSA efforts on developing requirements for and prototyping of a space medicine decision support system for exploration class missions with predictive diagnostics capability. He is currently working as an Operations Engineer on the ISS program at CSA.

Wolfgang Fink is currently an Associate Professor and the inaugural Edward & Maria Keonjian Endowed Chair of Microelectronics with joint appointments in the Departments of Electrical and Computer Engineering, Biomedical Engineering, Systems and Industrial Engineering, Aerospace and Mechanical Engineering, and Ophthalmology and Vision Science at the University of Arizona in Tucson. He is a Visiting Associate in Physics at the California Institute of Technology, and holds concurrent appointments as Visiting Research Associate Professor of Ophthalmology and Neurological Surgery at the University of Southern California. Dr. Fink is the founder and director of the Visual and Autonomous Exploration Systems Research Laboratory at Caltech (http://autonomy.caltech.edu) and at the University of Arizona (http://autonomy.arizona.edu). He was a Senior Researcher at NASA's Jet Propulsion Laboratory from 2000 till 2009. He obtained a B.S. and M.S. degree in Physics and Physical Chemistry from the University of Göttingen, Germany, and a Ph.D. in Theoretical Physics from the University of Tübingen, Germany in 1997. Dr. Fink's interest in human-machine interfaces, autonomous & reasoning systems, and evolutionary optimization has focused his research programs on artificial vision, autonomous robotic space exploration, biomedical sensor/system development, cognitive/reasoning systems, and computer-optimized design. Dr. Fink is a Fellow of the American Institute for Medical and Biological Engineering (AIMBE). His work is documented in numerous publications and patents. Dr. Fink holds a Commercial Pilots License for Rotorcraft.

Andrew Hess is a 1969 graduate of the University of Virginia (BS Aerospace Engineering) and the U. S. Navy Test Pilot School. Andy attended George Washington University working towards a Masters in Technology Management and has completed many Navy and DOD sponsored professional and acquisition management courses. Andy is world renowned for his work in fixed and rotary wing health monitoring and is recognized as the father of Naval Aviation propulsion diagnostics. Working for the Naval Air System Command and beginning with the A-7E Engine Monitoring System program of the

Fault Monitoring Techniques for Nuclear Components

Gee-Yong Park[1] and Jung Taek Kim[2]

[1,2]*Korea Atomic Energy Research Institute,Yuseong-gu, Daejeon, Dukjin-dong 150, South Korea*

gypark@kaeri.re.kr
jtkim@kaeri.re.kr

ABSTRACT

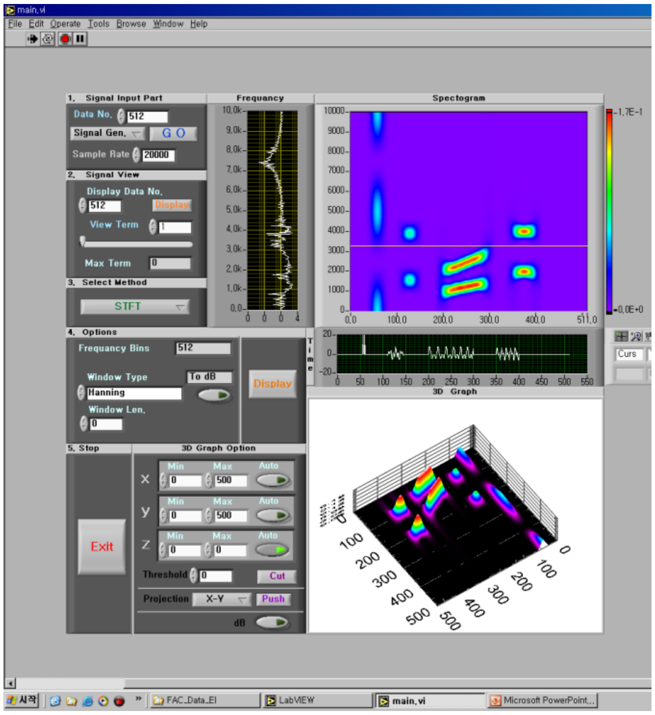
In this paper, we describe our previous studies for the development of an analysis algorithm and the application of a fault monitoring technique. Various signal processing methods have been implemented in the so-called monitoring tools to monitor and analyze abnormal conditions of components in nuclear power plants (NPPs). One of the analysis methods were devised by us for the efficient analysis of transient signals from NPP process components. This method, the adaptive cone-kernel distribution, is presented in this paper along with the description of the monitoring tool. Then, some application results using the monitoring tool are presented. As another application, the fault monitoring technique applied to the agitator driving system of a thermal chemical reduction reactor is also presented though this technique is not integrated in the monitoring tool yet.

1. INTRODUCTION

The fault monitoring technique consists of hardware and software elements to investigate successfully the status of a target component, equipment, or system. For the hardware part, a sensing type and an appropriate sensor for measuring relevant signals are first determined. Along with the sensors employed, the data acquisition system should be established. For the software part, it is usual in our studies that a signal processing method is selected to analyze the signal that were acquired by and stored in the data acquisition system, then some useful feature representing information of the target system status is extracted from the result of signal processing analysis, and finally a certain diagnostic method such as expert system or neural network is applied to the features, resulting in the identification of current status of the system to be monitored.

We implemented various signal analysis methods and improved some methods for application to the transient signals from a system. These various signal analysis methods covers from classical analysis methods such as the frequency or spectral analysis to the time-frequency analysis methods for analysis of transient signals such metal impact signal by a loose part in NPP.

Figure 1 shows the main page of our monitoring tool. This tool has classical spectral analysis methods and also various time-frequency analyses such as STFT (Short-Time Fourier Transform), WVD (Wigner-Vill Distribution), CWD(Choi-Williams Distribution), BJD (Born-Jordan Distribution), CKD (Cone-Kernel Distribution), and ACKD (Adaptive Cone-Kernel Distribution). The monitoring tool was implemented by LabVIEW program language. The analysis methods described above were implemented into "dll" libraries that were then integrated into the LabVIEW-based monitoring tool



Data No. 512
Signal Gen.
Sample Rate 20000
Display Data No. 512
View Term 1
Max Term 0
STFT
Frequancy Bins 512
Window Type Hanning
To dB
Window Len. 0
X Min 0 Max 500 Auto
Y Min 0 Max 500 Auto
Z Min 0 Max 0 Auto
Threshold 0
Projection X-Y
dB

Figure 1. Monitoring tool - main page

G. Y. Park et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The upper left part of Figure 1 indicates menus for reading input data and displaying this data and the lower left part is for menus for options and refining analysis results. The upper right screen is for displaying 2-dimensional time-frequency results along with frequency and time data. The lower graph is displayed only when 3 dimensional data need to be displayed. The more explanation of each component can be seen in Figure 2.

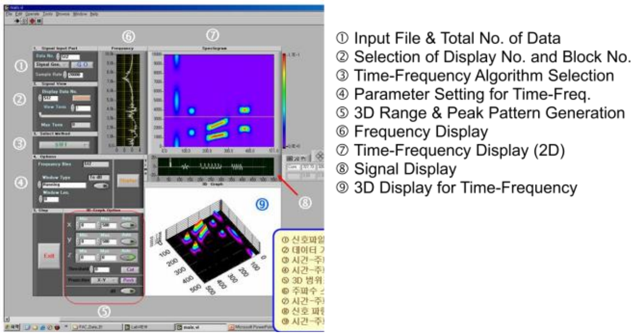

⑥
⑦
①
②
③
④
⑤
⑧
⑨
Spectrogram
Frequency
Exit
① Input File & Total No. of Data
② Selection of Display No. and Block No.
③ Time-Frequency Algorithm Selection
④ Parameter Setting for Time-Freq.
⑤ 3D Range & Peak Pattern Generation
⑥ Frequency Display
⑦ Time-Frequency Display (2D)
⑧ Signal Display
⑨ 3D Display for Time-Frequency

Figure 2. Menu functions of the monitoring tool

Among them, the adaptive cone-kernel distribution (ACKD) was devised by our team and this is described in detail in Section 2 with brief description of applications results. Section 3 describes other application result of fault monitoring techniques that had been carried out in our projects but are not implemented in this monitoring tool.

2. DEVELOPMENT OF ADAPTIVE CONE KERNEL DISTRIBUTION

In this section, an adaptive CKD, which was devised for improving analysis performance and also reducing calculation load, is described.

The general form of a time-frequency distribution (TFD) is represented by (Claasen & Meckleubrauker, 1980)

$$C_x(t,\omega;\phi)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{j(\xi t-\tau\omega-\xi u)}\phi(\xi,\tau)x(u+\tfrac{\tau}{2})x^*(u-\tfrac{\tau}{2})du d\tau d\xi \;,(1)$$

where x is an analytic signal, x* is the complex conjugate of x, τ represents the time lag, and ϕ is the kernel function. The form of TFD by Cohen (Cohen, 1995) is slightly different from Eq.(1) in that the parameter ξ has a negative sign. If the kernel function is such that $\phi(\xi,\tau) = 1$, then Eq.(1) is the Wigner-Vill distribution (WVD). The WVD is known to have very desirable properties in the time-frequency domain. For a signal composed of multi-component signals, however, it has a critical drawback in that ghost signals (i.e., called "cross-terms") present among the true signal components. These results make the time-frequency representation very difficult to interpret. To overcome this, time-frequency representation of a signal is usually performed base on the general form of Eq.(1) where the kernel function ϕ is designed from the previously proposed candidates (Hlawatsch & Boudreaux-Bartels, 1992) for a better representation of a case sensitive signal.

Various kernels have been proposed for satisfying the desirable properties (time and frequency maginals, finite support in time and frequency, and so forth) of TFD and at the same time reducing the undesirable effect (i.e., cross-terms). The exponential kernel, which is known as the Choi-Williams kernel (Choi & Williams, 1989), is the one that satisfies almost all of the desirable TFD properties and can suppress well the effects of the cross-terms. It is represented by

$$\phi(\xi,\tau)=e^{-\xi^2\tau^2/\sigma} \tag{2}$$

In Eq.(2) σ is a tuning parameter. Born-Jordan kernel has the form of

$$\phi(\xi,\tau)=\sin c(\alpha\xi\tau) \tag{3}$$

It fulfills almost all of the TFD properties. All the kernel functions are summarized well in the paper of Cohen (1995) and Hlawatsch and Boudreaux-Bartels (1992).

2.1. CKD and ACKD

Up to this time, there is no kernel that satisfies all the desirable properties and also shows the best cross-term reduction capability. The cone-kernel distribution proposed by Y. Zhao, et al (1990) is the one with the best capability of suppressing the cross-term effects; instead, of sacrificing many of desirable properties (Loughlin et al., 1993). The design of CKD is originated from the idea that a kernel should satisfy the time support and also enhancing the frequency resolution by paying a penalty to the neighbors of signal frequencies by the use of a so-called lateral inhibition (Zhao, et al., 1990). A different form of the general class of TFD of Eq.(1) can be presented such as

$$C_x(t,f;\phi)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi(t-u,\tau)x(u+\tfrac{\tau}{2})x^*(u-\tfrac{\tau}{2})e^{-j2\pi f\tau}du d\tau \tag{4}$$

The kernel function in Eq.(4) is the inverse Fourier transform of the kernel function in Eq.(1) with respect to ξ. The cone kernel in the t-τ domain in (4) is represented as

$$\phi(t,\tau)=\begin{cases} g(\tau), & |\tau|\geq a|t| \\ 0, & \text{O.W.}\end{cases} \tag{5}$$

In Eq.(5), the cone boundary parameter, a, adjusts the slopes of the cone with the constraint that $2 \leq a < \infty$ and usually set to 2 according to the finite support property (Claasen & Meckleubrauker, 1980). The function $g(\tau)$ is a sort of window in the Fourier transform for preventing a frequency leakage and it is usually represented by the Gaussian function.

The discrete form of the above equation (Czerwinski & Jones, 1995) is represented by

$$C_x(n,f;CKD) = \sum_{k=-T}^{T} g(k) \sum_{p=-|k|}^{|k|} x(n+p+k)x^*(n+p-k)e^{-j2\pi fk} \tag{6}$$

In Eq.(6), the continuous variable f will be represented by the discrete Fourier data at the implementation phase. According to the cone length T in Eq.(6), the resolutions of time and frequency domains are traded off.

For a signal with multi-components, it is necessary for the cone length to vary adaptively according to the signal type. The adaptive cone-kernel distribution (ACKD) was proposed by Czerwinski & Jones (1995) where the performance measure was the highest time-frequency signal energy normalized by the square of the cone length. This method usually shows a reasonably optimal value of the cone length according to the signal type, but it requires massive computations and a careful selection of the time-axis range for the signal data.

In this paper, a more computationally efficient adaptive method is proposed. In this method, at a particular time step n, the frequency values are calculated for each incremental step of a variation of the cone length. The performance measure is the normalized Shannon's entropy that is applied to the frequency data obtained at each incremental step in the cone length for a particular time. The normalized Shannon's entropy is expressed as

$$E(T,n) = \sum_{m=0}^{M-1} f_N(m,n,T) \ln \frac{1}{f_N(m,n,T)} \tag{7}$$

In Eq.(7), n is the time index and m is the discrete frequency index, M is the total frequency data. The f_N is the normalized energy and represented by

$$f_N(m,n,T) = \frac{\left|C_x(n,m;CKD\ \text{for}\ T)\right|^2}{\sum_{m=0}^{M-1}\left|C_x(n,m;CKD\ \text{for}\ T)\right|^2} \tag{8}$$

The optimal cone length T at a particular (fixed) time index is determined when the entropy has a local minimum value over the variations of the cone length T. If it is decreasing smoothly then the optimal T is determined when the entropy value is below the threshold value such as

$$E_{th} = E_{min} + \lambda(E_{max}\text{-}E_{min}) \tag{9}$$

In Eq.(9), E_{th} is the threshold entropy, E_{min} and E_{max} represent the minimum and maximum values of the entropy, respectively, and λ is the given threshold parameter. In determining the optimal cone length, the entropy trend shows a small fluctuating behavior, which may induce an inaccurate determination of the local minimum point. In order to remedy this problem, the entropy plot for the variation of T at a particular sample point is smoothed by the curve smoothing technique (Moon, 1998).

2.2. Performance Evaluation for Arbitrarily Synthesized Signal

Figure 3 shows an arbitrary, multi-component signal that contains two impulses at the data sample index i=56 and 60, respectively, one burst signal at i=108~148, two chirp signals at i=200~300, and two mixed sine waves at i=350~400. The sampling rate is given by 50 kHz. The detailed information of the multi-components (S1, S2, S3, S4, and S5) is presented in Table 1.

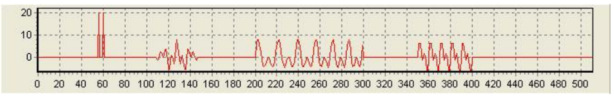


Figure 3: An arbitrarily generated signal

Table1. Description of signal components

Signals	Expression & Description
#1 (S1)	S1 = 20 at i=56, and S1 = 0, otherwise
#2 (S2)	S2 = 20 at i=60, and S2 = 0, otherwise
#3 (S3)	S3 = 4M{cos(2π×20×i/256)+ cos(2π×50×i/256)}, and $M = \begin{cases} 1-\frac{1}{400}(i-128)^2 & ,108 \le i \le 148 \\ 0 & ,\text{Otherwise} \end{cases}$

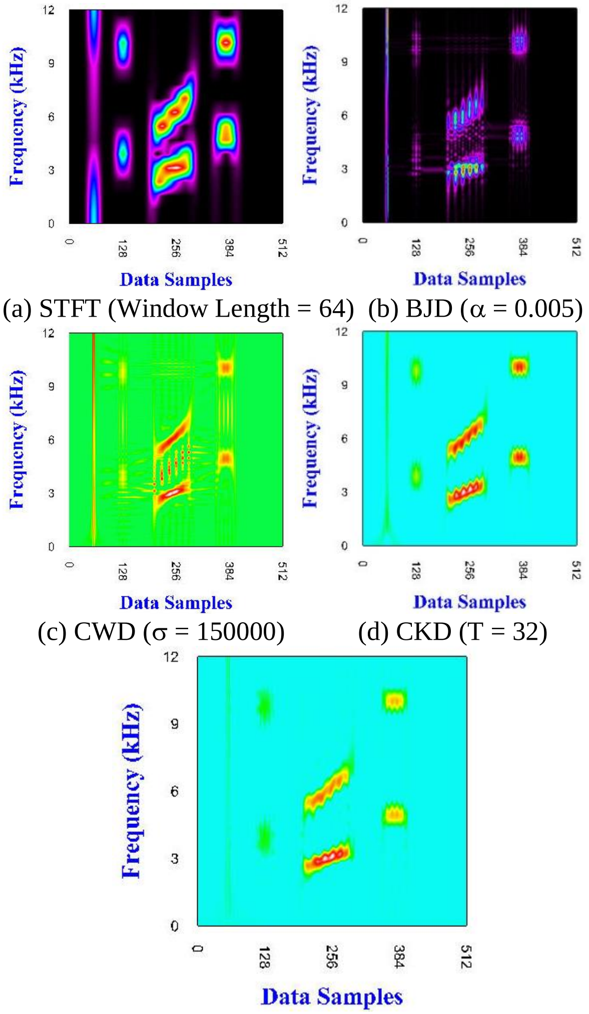
#4 (S4)	S4 = 4{cos(2π×(i/8)×(i/512))+
	cos(2π×(i/16)×(i/512))},
	for 200≤i≤300, and S4=0, otherwise

#5 (S5)	S5 = 4{cos(2π×i/5)+cos(2π×i/10)},
	for 200≤i≤300, and S5 = 0, otherwise

Figure 4 displays the results of the STFT, the Born-Jordan distribution (BJD), the Choi-Williams distribution (CWD), the Cone-Kernel distribution (CKD) and the adaptive CKD (ACKD). Figure 4(a) represents the STFT with the window length of 64, which shows clearly the superior depression of the cross-terms but very poor time and frequency resolutions. The BJD in Figure 4(b) is obtained from the parameter α = 0.005 in Eq.(3). The CWD with σ=150000 in Eq.(2) is displayed in Figure 4(c), where the cross-terms are still disturbing the time-frequency representation. Figure 4(d) displays the CKD with T=32 in Eq.(6). All the figures depict the real part. Though the CKD with an appropriate cone length (T=32) shows a better the time-frequency representation than the other time-frequency representations in Figures 4(a),(b), and (c), it is necessary the cone length be varied according to the signal type.

Figure 4(e) shows the result of the ACKD proposed in this paper. The maximum search range of the cone length is given by 64 and the threshold parameter is set by λ = 0.05. Figure 4(e) shows a good time-frequency representation and moreover, the two vertical strips for corresponding impulse signals can be discriminated. Figure 5(a) and Figure 5(b) magnify the results of ACKD of Figure 4(e) and CKD of

Figure 4(d), respectively, for the data sample range of 0~100. As can be seen in Figures 5, the discrimination of the two impulses are much clearer in ACKD. The optimal values of the cone length calculated in this adaptive method are shown in Figure 6 along the data sample index.

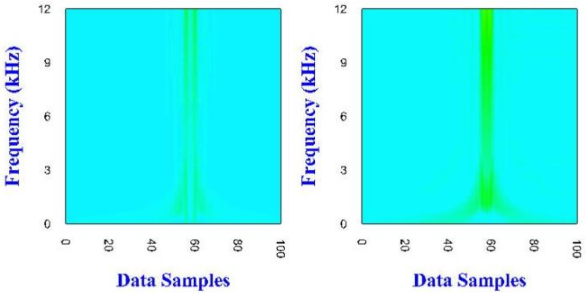


(a) STFT (Window Length = 64) (b) BJD (α = 0.005)

(c) CWD (σ = 150000) (d) CKD (T = 32)

(e) ACKD

Figure 4: Time-Frequency representations for an arbitrarily generated signal



(a) ACKD (0~100) (b) CKD (0~100)

Figure 5: Comparison of magnified ACKD and CKD

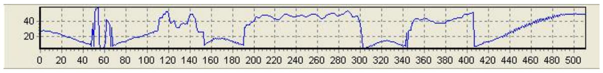
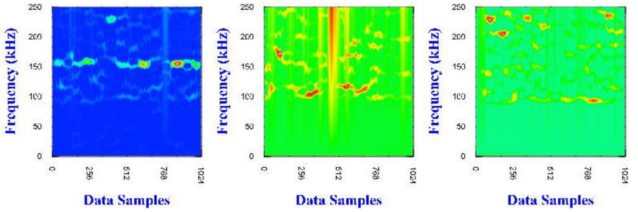


Figure 6: Optimal cone length along signal sequence

2.3. Application of Monitoring Tool

The monitoring tool as in Figure 1 was used for monitoring of the integration of a check valve and also for identifying the status of pipe corrosion, which had been performed by the joint research between the KAERI and the SNL (Sandia National Laboratory) as an I-NERI project.

Some example of application of ACKD to acoustic emission (AE) sensor signals for the check valve monitoring is depicted in Figure 7.



(a) Normal Close (b) Disk wearing (c) Foreign object

Figure 7: Results of ACKD for AE signals from a check valve disk

As can be seen in Figure 7, the AE signals from a healthy check valve typically have a signal component dominant at 150 kHz and, if an abnormal situation happens, there can be seen that signal components dominant at other frequencies begin to appear. When the leak occurs due to disk wearing, the dominant frequencies are extensively spreading over 100 kHz as can be seen in Figure 7(b). For the disk stuck by a foreign object, it can be seen that the signal components at higher frequencies are generated.

For the identification of pipe corrosion, it was identified that the signals from accelerometers installed in the area of the pipe elbow did not represent distinguishable transient characteristics in such a passive method (in other words, in the case of two transmitting- and receiving- accelerometers, a certain discriminating feature was supposed to be identified.) For the experiment for the pipe corrosion, the STFT was applied to signals from accelerometers and the linear scale of the distribution of STFT was transformed into the log scale. The ridge pattern is extracted from this result. Figure 8 shows the ridge pattern for the pipe in the normal state.

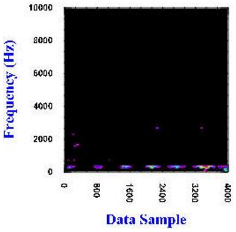
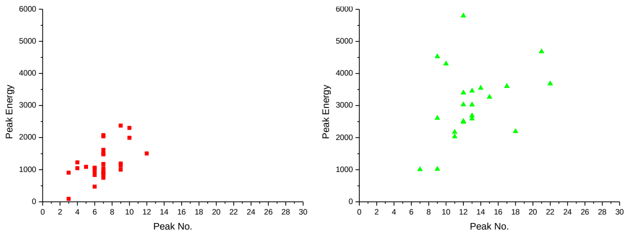


Figure 8: Ridge pattern for the pipe with normal state

The ridge pattern in Figure 8 was abstracted by the use of peak number and peak energy. Figure 9 shows some application results from the two types of experimental pipes; the pipe with normal state and 2 mm mechanical thinning.



(a) Normal state (b) 2 mm thinning

Figure 9: Peak number vs. peak energy for pipes with different local width

As can be seen in Figure 9, the abstracted ridge pattern is moving from the lower left portion to the upper right portion. This shift trend could be identified consistently for the pipe with 1 mm thinning, and was validated for the real data from the pipe corrosion.

3. Application of Fault Monitoring Technique

This section presents additional application example from our previous study (Park et al., 2003) that are not integrated into the monitoring tool in Section 2. The target system to which a fault monitoring technique was applied is the agitator driving system. The agitator driving system equipped on the top of the thermal reduction reactor for a high-temperature chemical reaction is composed of the magnetic driver and the agitator whose rotating axis is connected to the motor installed on the side of the thermal reduction reactor via the flexible joint.

Figure 10 shows the configuration of the thermal reduction reactor and the agitator driving system. The power transmission through the flexible joint produces vibrations and, for series of the operations, this induces the looseness of the surrounding bolts. In the real operations, there existed that the operator sometimes forgot to tighten some of bolts.

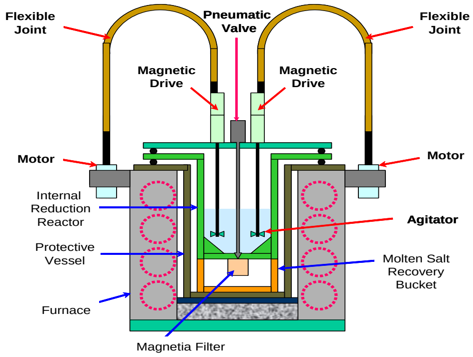


Figure 10. Configuration of the full-scale thermal reduction reactor and the agitator driving system

During the thermal reduction process, the fume is generated from the chemical reaction and propagates through the internal space of the agitator driving system. The bearings in the magnetic driver are affected by the fume, which results in the corrosion/wear of the bearings and the blocking of the clearance between the rotating axis and the outer ring.

The fault in the agitator driving system increases the burden of regulating the agitator rotating speed to the pre-set point (200 rpm) for the optimal chemical reaction. In order to identify the fault occurrence and its cause, the fault monitoring technique for the agitator driving system is developed. This technique is implemented on the vibration signals measured by two accelerometers on the outer shroud of the magnetic drive as shown in Figure 11.

Through the experiments, the vibration signals for a speed of 200 rpm with various faults were measured. The sampling rate was set to 25.6 kHz for all cases. The data for 5 types of faults are analyzed. The five faults presented in this paper are the clearance blocking, the bearing defects, the lower bolts looseness, the upper bolts looseness, and the upper-right bolt looseness.

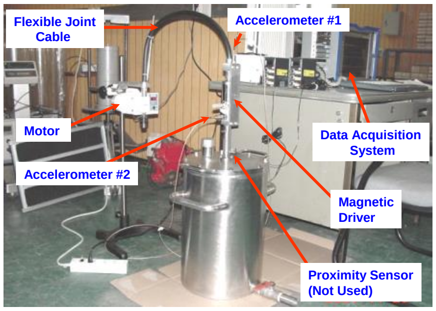


Figure 11. Experimental facility for the agitator driving system

The rotation speed is fluctuated with the maximum deviation of ±20 rpm around 200 rpm. In order to classify the faults, the spectral analysis was first applied to the signals for 5 faults and the normal rotation. From the spectral analysis, some faults such as the clearance blocking and the bearing defects show slightly different trend other than the normal rotation but it is not easy to identify the distinguishing peaks for discriminating a fault.

In order to identify accurately a fault in the agitator driving system, the wavelet decomposition (Burrus, et al., 1998) was applied to the vibration signal. Figure 12 shows the wavelet decomposition for a normal vibration signal that has 65,536 data samples. As can be seen in Figure 12, the vibration signals are decomposed into the corresponding components that have the octave band frequency contents. The feature extraction was performed by the 2nd order moment calculation. The 2nd order moment calculation was simple and showed good distinguishable features for corresponding faults as can be seen in Figure 13. For establishing the diagnosis process, the neural network classifier, which is called Fuzzy ARTMAP (Carpenter, et al., 1992), was constructed.

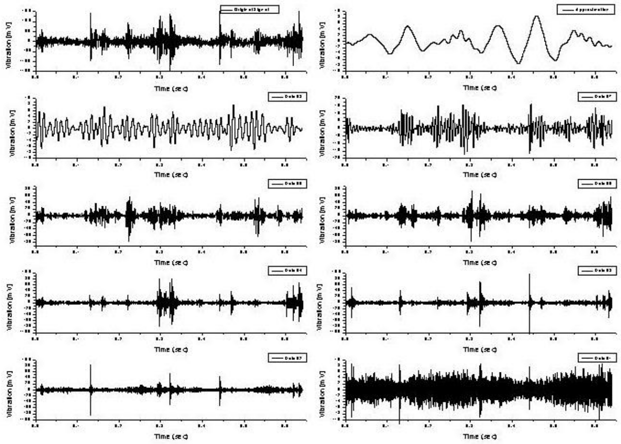


Figure 12. Wavelet decomposition for normal signal

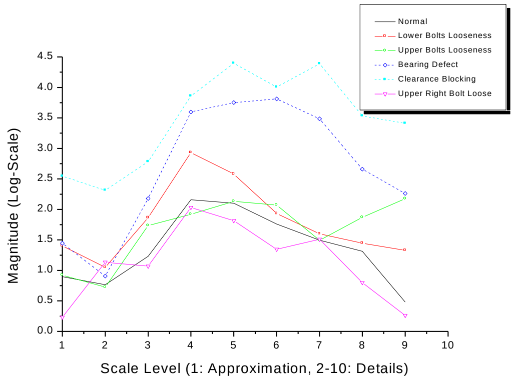


Figure 13. Features extracted from the results of wavelet decomposition for 6 types of cases

The fuzzy ARTMAP is a class of neural networks that perform incremental supervised learning of recognition categories and multidimensional maps in response to input vectors presented in arbitrary order. Figure 14 shows the architecture of the fuzzy ARTMAP where two fuzzy ART (Carpenter, et al., 1991) modules (ART$_a$ and ART$_b$) and a map field F$_{ab}$ are involved. The input vectors **A** of dimension M$_a$ and **B** of dimension M$_b$ are the feature vectors respectively corresponding to the symptom and the cause. The components of each input vector are analog or binary values within the range of [0,1]. The each component of the input vector represents a feature item that is grouped to establish the representative feature. Thus the analog value of each component of the input vector means the degree of belongingness to the corresponding feature item, and this is similar to the fuzzy membership value. The detailed description on the operational mechanisms of the fuzzy ARTMAP is presented in Carpenter, et al (1992).

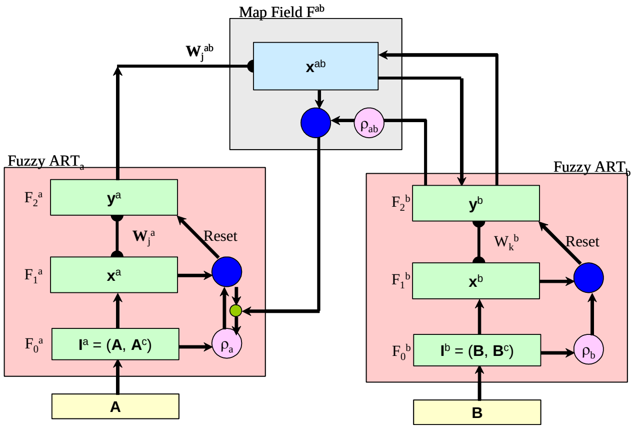
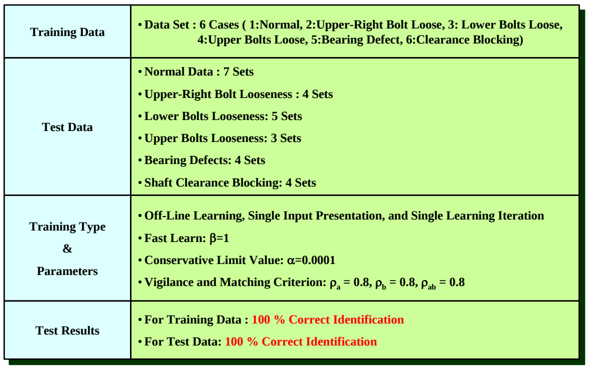


Figure 14. Configuration of fuzzy ARTMAP

During training the fuzzy ARTMAP, one set for each fault was selected from the data set and the remainder data set were used to test the fuzzy ARTMAP performance. In the training phase, training data for each fault is presented just once to the fuzzy ARTMAP and single learning iteration is performed. The input vector **A** for the fuzzy ART$_a$ is the 2nd momentum feature values and has a dimension of M$_a$ = 9. All the inputs for the fuzzy ART$_a$ are normalized. The input vector **B** for the fuzzy ART$_b$ is the binary values with a single '1' that represent a specific fault and has a dimension of M$_b$ = 6. Figure 15 summarizes the test data and the parameters for the fuzzy ARTMAP. From tests, the fuzzy ARTMAP showed the perfect fault identification though some test data are distorted from the training data.



Training Data	• Data Set : 6 Cases (1:Normal, 2:Upper-Right Bolt Loose, 3: Lower Bolts Loose, 4:Upper Bolts Loose, 5:Bearing Defect, 6:Clearance Blocking)
Test Data	• Normal Data : 7 Sets • Upper-Right Bolt Looseness : 4 Sets • Lower Bolts Looseness: 5 Sets • Upper Bolts Looseness: 3 Sets • Bearing Defects: 4 Sets • Shaft Clearance Blocking: 4 Sets
Training Type & Parameters	• Off-Line Learning, Single Input Presentation, and Single Learning Iteration • Fast Learn: β=1 • Conservative Limit Value: α=0.0001 • Vigilance and Matching Criterion: ρ_a = 0.8, ρ_b = 0.8, ρ_{ab} = 0.8
Test Results	• For Training Data : 100 % Correct Identification • For Test Data: 100 % Correct Identification

Figure 15. Parameter settings of the fuzzy ARTMAP and the diagnosis results for faults

4. CONCLUSIONS

In this paper, the monitoring tool established at our department and the fault monitoring techniques using this tool and other means are briefly described. In the nuclear field, classical analysis methods such as a spectral analysis or an auto-regressive model are applicable to most of signals stored in the data acquisition system. For a more delicate analysis for transient signals, an analysis method based on the time-frequency basis is useful. In this paper, an adaptive cone-kernel distribution whose window size is varied according to its adaptive mechanism is presented. This method is so efficient for computing time that it can be used on line. The monitoring tool described in this paper contains various signal-analysis methods. In our works, this tool was applied to the monitoring of the check valve and the identifying the status of pipe corrosion. By the use of the monitoring tool we developed, a new method or technique can be easily implemented and incorporated into this tool.

As one additional application, the fault monitoring technique of the agitator driving system was described. In this monitoring technique, the wavelet decomposition is used as a signal processing analysis and 2nd order momentum is used to extract the signal features from the decomposed signals. For investigating or diagnosing the fault status, the fuzzy ARTMAP is employed for discriminating robustly the fault patterns. These signal processing algorithm and fault monitoring technique are also going to be implemented in the monitoring tool.

REFERENCES

Burrus, C. S., Gopinath, R. A., & Guo, H. (1998), *Introduction to Wavelets and Wavelet Transforms*, Prentice Hall.

Claasen, T. A. C. M., & Meckleubrauker, W. F. G. (1980), The Wigner Distribution – A Tool for Time-Frequency Signal Analysis; Part III-Relations with Other Time-Frequency Signal Transforms. *Philips J. of Research*, vol.35, no.6, pp.372~389.

Cohen, L. (1995), *Time-Frequency Analysis*, Prentice-Hall.

Choi, H. I., & Williams, W. J. (1989), Improved Time-Frequency Representation of Multicomponent Signals using Exponential Kernels. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vo.37, no.6, pp.862~871.

Czerwinski, R. N., & Jones, D. L. (1995), Adaptive Cone-Kernel Time-Frequency Analysis. *IEEE Transactions on Signal Processing*, vol.43, no.7, pp.1715~1719.

Hlawatsch, F., & Boudreaux-Bartels, G. F. (1992), Linear and Quadratic Time-Frequency Signal Representations. *IEEE SP Magazine*, pp.21~67, April.

Loughlin, P. J., Pitton, J. W., & Atlas, L. E. (1993), Bilinear Time-Frequency Representations: New Insights and Properties. *IEEE Transactions on Signal Processing*, vol.41, no.2, pp.750~767.

Moon, B. S. (1998), A Curve Smoothing Method by Using Fuzzy Sets. *Fuzzy Sets and Systems*, vol.96, pp.353~358.

Park, G. Y., et al. (2003), Fault Diagnosis for Agitator Driving System in a High Temperature Thermal Reactor. *Nuclear Engineering and Technology*, vol.35, no.5, pp.454~470.

Zhao, Y., Atlas, L. E., & Marks II, R. J. (1990), The Use of Cone-Shaped Kernels for Generalized Time-Frequency Representations of Nonstationary Signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol.38, no.7, pp.1084~1091.

BIOGRAPHIES

Gee-Yong Park Dr. Gee Yong Park achieved Ph.D. degree at the Department of Nuclear Engineering in Korea Advanced Institute of Science and Technology (KAIST) in 1996. Since then, he has been involved in Department of I&C and Human Factors of Korea Atomic Energy Research Institute (KAERI). His major research area includes the controller design and the monitoring and diagnosis of nuclear power plants, software reliability, and cyber security.

Jung Taek Kim Mr. Jung Taek Kim is in course of Ph.D. degree at the Department of Electronic Engineering in Chungnam National University (CNU) since 2002. He is working in Division of I&C and Human Factors of Korea Atomic Energy Research Institute (KAERI) during 26 years. His major research area includes a design of the alarm processing system and the monitoring and diagnosis system of nuclear power plants.

Poster Papers

Signal Processing Techniques to Improve an Acoustic Emissions Sensor

Eric Bechhoefer[1], Yongzhi Qu [2], Junda Zhu [2] and David He [2]

[1]*Green Power Monitoring Systems, LLC, Vermont, 05753, USA*
ebechhoefer@gmail.com[1]

[2]*Dept of Mechanical and Industrial Engineering, The University of Chicago, Illinois, 60607 USA*
yongzhiqu@hotmail.com [2]
jz@renewablenrgsystems.com[2]
davidhe@uic.edu[2]

ABSTRACT

Acoustic Emissions (AE) are stress waves produced by the sudden internal stress redistribution of material caused by changes in the internal structure of the material. Possible causes of these changes are crack initiation and growth, crack opening/closure, or pitting in monolithic materials (gear/ bearing material). Where as vibration can measure the effect of damage, AE is a direct measure of damage. Unfortunately, AE methodologies suffer from the need of high sample rates (4 to 10 Msps) and relatively immature algorithms for condition indictors (CI). This paper hypothesizes that the AE signature is the result of some forcing function (e.g. periodic motion in the case of rotating machinery). By using analog signal processing to demodulating the AE signature, one can reconstruct the information carried (e.g. modulation) by the AE signature and provide improved diagnostics. As most on-line condition monitoring systems are embedded system, analog signal processing techniques where used which reduce the effective sample rate needed to operate on the AE signal to those similarly found in traditional vibration processing systems. Further, by implementing another signal processing technique, time synchronous averaging, the AE signal is further enhanced. This allowed, for the first time, an AE signal to be used to identify a specific component within gearbox. This processing is tested on a split torque gearbox and results are presented.

1. INTRODUCTION

The promise of condition based maintenance (CBM) systems is to produce maintenance saving by reducing unscheduled maintenance events. As confidence in CBM improves and systems mature, maintenance paradigms can be moved to a true, “On Condition” practice. Unfortunately, for many industries, CBM is an immature technology and has not proven itself in operational circumstances. The low penetration (3% of installed turbines) of condition monitoring systems (CMS) into the wind turbine industry is symptomatic of the lack of confidence in the ability of CMS to deliver their’ promised performance. The industry needs better sensing and analysis capabilities in order to capture these markets.

One aspect of condition monitoring on wind turbines is the extraordinarily low frequencies of the environment. The main shaft rate on utility scales wind turbines range from 0.11 to .25 Hz. With typical planetary gearbox frequencies of 1:5, gear mesh frequencies in the range of 10 to 25 Hz are not uncommon. Because acceleration is the second derivative of displacement, gear mesh frequencies are on the order 0.005 to 0.02 G’s, making gear fault detection difficult with tradition vibration condition monitoring systems.

Acoustic Emissions (AE) are the stress waved produced by the sudden internal stress redistribution of material caused by the changes in the internal structure of the material. Possible causes of these changes are crack initiation and growth, crack opening and closure, or pitting in various monolithic materials (gear, bearing material) or composite materials (concrete, fiberglass). Most sources of AE are damage related. Thus the ability to detect AE can be used to give diagnostics indications of component health.

AE is a direct measure of damages instead of the indicator of the result of damage (such as vibration). AE can potentially be a more sensitive of fault, especially in the low frequency seen on the planetary gearboxes of wind turbines.

Bechhoefer, E. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

However, AE systems tend to be more expensive and difficult to implement over vibration based non-destructive test or CMS.

From a development perspective, AE has a number of perceived disadvantages:

- AE signals are relatively high frequency, 1 to 4 MHz, thus the sample rates are high (4 to 10 MSPS),
- Processing of the data, needed for feature extraction, is made more difficult because of the high sample rates and large volume of data needed to be processed (consider 10 MSPS for 40 seconds, which would capture just 6 revolutions on a 2.5 MW wind turbine, is 400 million samples!),
- Typical AE analysis is limited in its "action ability", meaning that its can detect or count AE events, but does not tie the event to a component in the gearbox.

The ideal CMS would:

- Allow for fault detectability afforded the direct measure of damage in the AE signal, which is independent of rotation rate,
- The maturity of vibration processing techniques to provide actionable information by identifying the component which is damages,
- Reduce the computation burden of AE by sampling at the lower rate.
- Improve the detection of low frequency components which are now difficult to do with vibration based CMS.

We developed an analog Hilbert envelope circuit to demodulate the AE signal, which greatly reduced the sample rate typically needed for AE. This reduces the cost of a AE CMS system by allowing the use of lower end, audio analog to digital converts and low end microcontroller for processing of that data. Additionally, we applied time synchronous averaging of the demodulated signal to improve the signal to noise ratio of the signal which normally would be undetectable, but also allowed the identification of the damaged component (AE analysis typically only identifies that there is damage, but not what is damage). Finally, this new technique was compared against tradition vibration analysis, using similar algorithms, and was found to be significantly better and gear fault detection.

2. AE: Acoustic Emissions

AE as phenomena, has been observed in many disparate fields of study. The earliest use of AE analysis was in geology and seismology. Here the analysis of elastic waves produced by an earthquake was used to find the location and depth of the event. Similarly, AE was proposed as a method to predict rockburst in mines. Tinsmiths have noted the "tin cry" associated with twinning deformation, and the clicks noted during heat treatment of steels is well documented (this is related to martensitic transformations of metals, which has been show to be a strong emitter of AE).

The general acceptance that AE is associated with dislocation and plastic deformation/crack propagation in metal was first proposed by (Liptai, 1969). The essential principles of AE where explored in (Liptai, 1970), by considering a grain of polycrytstalline material (steel, for example), where the grain boundary has a diameter of $d=5 \times 10^{-3}$in. During a strain event, the upper half of the grain slips over the lower half by a distance of $d=1 \times 10^{-3}$in. Given a shear modulus of $G=4 \times 10^{6}$ psi, then the stress driving the deformation is eq 1, and the energy change occurring with a deformation is eq. 2.

$$\sigma_s = {}^{sG}\!/\!_{d} \tag{1}$$

$$\omega = {}^{s^2GA}\!/\!_{2d} \equiv 10^{-12}\, {}^{in}\!/\!_{lbs} \tag{2}$$

where A is the sheared area. This allows one to estimate the frequency of an event as:

$$\omega = \sqrt{{}^{2GA}\!/\!_{dm}} \approx 5 \times 10^{6}\, {}^{rad}\!/\!_{\sec} \approx 0.8 MHz \quad (3).$$

where m is the half mass of the grain (assumed to be steal). While estimates vary with density, grain size, and material, this estimate serves to bound the AE frequency from 500 KHz to perhaps 40 MHz.

2.1. AE: State of the Art

Most AE products quantify five basic condition indicators (Figure **1**): Amplitude, Duration, Rise Time, Counts and the mean area under the rectified signal envelope (MARSE). Other condition indicators (CI), such as average frequency (counts/duration), are function of the basic AE CIs and have been found to be useful in non-destructive test literature (Miller, 2005). Cumulative counts and cumulative absolute energy have also been shown to correlate to the fatigue crack growth process (Barsoum 2009).

More recent studies have focused on the use of wavelet to de-noise and enhance the AE signal. For example (Abouel-seoud 2012) used a continuous wavelet transform improve the signal to noise ratio for diagnostics on a wind turbine planetary gearbox. In (Gu, 2011), a signal processing method for AE signal by envelope analysis with discrete wavelet transforms followed by spectral analysis allowed visualization of the gearbox fault frequencies.

It most be noted that none of these analysis can directly tie the AE CI with a specific component within a gearbox. Instead, these techniques can give indicators that there is a fault present, and rely upon more tradition techniques such a Borescope to identify the damage component.

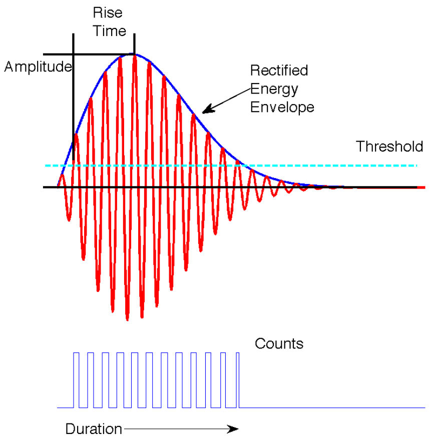


Figure 1 AE Condition Indicators

These methodologies, while successful in diagnostics, fundamentally do not address the hardware/software need to sample at lower data rates (needed for lower cost systems). Additionally, wavelets required off-line processing/ optimization techniques to select coefficients/levels to achieve successful diagnostics.

For a commercial product, the reapplication cost (e.g. off line analysis, configuration for the given application, etc.) and hardware costs are a large driving factor. This is the motivation to develop a system with the performance of AE fault detection, but without sacrificing the cost advantage of tradition vibration based systems.

3. Analog Processing to Improve the AE Sensor

The AE signal is generated by an impulse or forcing function, which causes a dislocation in the material. For rotating components such as gears, that force and the resulting damage is periodic. With this view, we hypothesized that the AE signal is the carrier signal on which the forcing function is modulated. The forcing function information content relates to the damage it is exciting (e.g. the AE signal). For nominal gears, there should be no AE signal, while a damaged gear should generate a period AE response.

In the fault case, the information of interest is not the AE signal, but the modulated force/load that is causing the AE burst. This type of information process is similar to the information in an amplitude modulated (AM) radio frequency signal, where the information is recovered by demodulating the radio signal.

In an AM radio, the carrier is demodulated using an analog signal conditioning circuit. This allows the system to be designed at audio frequencies (10s of KHz vs. MHz of the carrier signal). In the proposed analysis, the AE signal is demodulated with an analog circuit, and the result acquisition system is designed to performance at tradition vibration processing frequencies (100 KHz vs. MHz). The signal processing can then be performed on low cost embedded microcontrollers instead of higher end computers.

A demodulator shifts the carrier frequency to baseband, eq(4), followed by low pass filtering and enveloping.

$$\cos(a)\times\cos(b)=1/2[\cos(a-b)+\cos(a+b)] \qquad (4)$$

The envelope is the absolute value of the Hilbert transform. In frequency domain, the Hilbert transform is defined in the Fourier domain as: $2X(f)$, for $f>0$, and $X(f)=0$, for $f<0$, which easily computed in software. As stated, one of the objectives is to perform this signal processing in an analog circuit, such as in (Figure **2**). The raw, time domain signal from the AE sensor is defined as x(t). x(t) is quadrature demodulated by convolving the signal x(t) with a frequency near the carrier frequency (cos(ft)) and then low pass filtered to remove the image. The carrier frequency is generated by a voltage-controlled oscillator (VCO) or through low pass filtering a pulse width modulated (PWM) signal. This allows one to configure the demodulation process for different materials (which may have different AE carrier frequencies) or for different AE sensors, which may have different frequency responses.

After low pass filtering to remove the image (e.g. cos(a+b)) of the baseband signal, the quadrature is create by phase shifting the baseband signal by π/2 radian. The quadrature signals are then squared, summed and the square root is taking. This circuit can be built at low cost using operational amplifiers (op amps) per (Horowitz, 1995) or by using a monolithic multiplier/divider such as the AD532. This transforms the AE signal to its demodulated envelop.

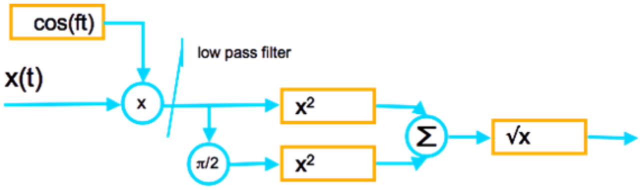


Figure 2 Analog Signal Process for the AE Sensor

The advantage of using an analog/hardware solution instead of the using a digital approach is a greatly simplified acquisition and processing system. Instead of designing a system to sample at potentially 10 MSPS (which includes increased memory, a high performance processor, high speed ADC, increased capacity of the power supply, increased heat dissipation), more modest 100 KSPS system can be designed. Note that the limit of the system is no longer the sample rate of the ADC, but the bandwidth of the analog devices, typically on the order to 2 GHz.

While other researchers have "enveloped" the AE signal by low pass filtering the rectified AE signal (Miller, 2005), this does capture the modulation rate of the forcing function. This is because a rectified/low pass enveloping technique does not heterodyne the AE signal to base band, as does the presented technique.

For rotating machinery, e.g. a gearbox, where the load is periodic the, the envelope of the AE sensor contain the information related to any gear faults within the gearbox. This has the advantage of giving actionable information as to the faulted component, as the AE signal is generated as a result of the periodic load of a specific component. As such, the modulation rate is the same as the damaged component rate. This in turn is easily identified through spectral analysis.

3.1. Feature Extraction from the AE Envelope

Vibration signatures for machinery faults tend to be small relative to other vibration signatures. For example, in the typical gearbox, the energy associated with gear mesh and shaft vibrations will be orders of magnitude larger than a fault feature. This is also the the case in performing analysis on the AE envelope. Spectral analysis or root mean squares (RMS) of AE signal are not powerful enough CIs to detect an early fault, let alone provide information useful for prognostics. Additionally, since all rotating equipment has limits on the feedback controls driving it, there is some variation in speed. When taking the spectrum, this variation in shaft speed violates requirement of stationarity.

To improve the performance of the gear analysis and to control for variation in shaft rates, the analysis will based on operations of the time synchronous average (Bechhoefer, 2009). Time synchronous averaging (TSA) is a signal processing technique that extracts periodic waveforms from noisy data. The TSA is well suited for gearbox analysis, where it allows the AE signature of the gear under analysis to be separated from other gears and noise sources in the gearbox that are not synchronous with that gear. Additionally, variations in shaft speed can be corrected, which would otherwise result in spreading of spectral energy into an adjacent gear mesh bins. In order to do this, a signal is phased-locked with the angular position of a shaft under analysis.

This phase information can be provided through a *n* per revolution tachometer signal (such as a Hall sensor or optical encoder, where the time at which the tachometer signal crosses from low to high is called the zero crossing).

The model for vibration in a shaft in a gear box was given in (McFadden, 1985) as:

$$x(t) = \Sigma_{i=1:K} X_i(1+ a_i(t))cos(2\pi i f_m(t)+ \Phi_i)+b(t) \qquad (5)$$

where:

X_i is the amplitude of the *k*th mesh harmonic

$f_m(t)$ is the average mesh frequency

$a_k(t)$ is the amplitude modulation function of the *k*th mesh harmonic.

$\phi_i(t)$ is the phase modulation function of the *k*th mesh harmonic.

Φ_i is the initial phase of harmonic *k*, and

$b(t)$ is additive background noise.

The mesh frequency is a function of the shaft rotational speed: $f_m = Nf$, where N is the number of teeth on the gear and f is the shaft speed, with no reduction in the analysis performance. This is a general model, and it is hypothesized in this paper that the vibration signal can be replace by the AE envelope signal.

This TSA model assumes that f is constant. As noted, due to the finite bandwidth of the feedback control, there is some wander in the shaft speed due to changes in load or feedback delay. This change in speed will result in smearing of amplitude energy in the frequency domain. The smearing effect, and non synchronous noise, is reduced by resampling the time domain signal into the angular domain: $m_x(\theta) = E[x(\theta)] = m_x(\theta+\Theta)$. The variable Θ is the period of the cycle to which the gearbox operation is periodic, and E[] is the expectation (e.g. ensemble mean). This makes the assumption that $m_x(\theta)$ is stationary and ergodic. If this assumption is true, than non-synchronous noise is reduce by 1/sqrt(*rev*), where *rev* is the number of cycles measured for the TSA.

3.2. Condition Indicators based on the TSA

The TSA is an example of angular resampling (McFadden, 1985), where the number of data points in one shaft revolution (r_n) are interpolated into *m* number of data points, such that:

- For all shaft revolutions *n*, *m* is larger than *r*,
- And $m = 2^{\text{ceiling } (\log2\ (r))}$ (assumes a radix 2 Fast Fourier Transform).

Linear, bandwidth limited linear interpolation, and spline techniques have been used. In this study, linear interpolation was used as it is considerable faster than spline or bandwidth limited filtering, with no apparent reduction in analysis performance of the TSA.

The TSA itself can be used for CIs. Typically, a CI is a statistics of a waveform (in the case the TSA). Common statistics are RMS, Peak to Peak, Crest Factor, and Kurtosis.

3.2.1. Gear Fault Condition Indicators

There are at least six failure modes for gears (IS10825, 2007): surface disturbances, scuffing, deformations, surface fatigue, fissures/cracks and tooth breakage. Each type of failure mode, potentially, can generate a different fault signature. Additionally, relative to the energy associated with gear mesh frequencies and other noise sources, the fault signatures are typically small. A number of researchers have proposed analysis techniques to identify these different faults (McFadden, 1985, Zakrajsek, 1993). These analyses are based on the operation of the TSA. In this study the fault is a broken tooth, and the following analysis where conducted (note the gear mesh frequency is found by: take the FFT of the TSA, take the absolute value of the number teeth + 1 bin):

- The common statistics of the TSA.
- Figure of Merit 0: the TSA peak-to-peak divided by the sum of the 1st and 2nd gear mesh frequencies;
- Side Band Modulation: the ration of the sum of the gear mesh side bands (+/-1 bin) divided by the gear mesh frequency.
- Residual Analysis: where shaft order 1, 2, and 3 frequencies, and the gear mesh harmonics, of the TSA are removed. Faults such as a soft/broken tooth generate a 1 per rev impacts in the TSA. In the frequency domain of the TSA, these impacts are expressed as multiple harmonic of the 1 per rev. The shaft order 1, 2 and 3 frequencies and gear mesh harmonics in the frequency domain, and then the inverse FFT is performed. This allows the impact signature to become prominent in the time domain. CIs are statistics of this waveform (RMS, Peak 2 Peak, Crest Factor, Kurtosis).
- Energy Operator: which is a type of residual of the autocorrelation function. For a nominal gear, the predominant vibration is gear mesh. Surface disturbances, scuffing, etc, generate small higher frequency values which are not removed by autocorrelation. Formally, the EO is: $\mathbf{TSA}_{2:n-1}$ x $\mathbf{TSA}_{2:n-1}$ x – $\mathbf{TSA}_{1:n-2}$ x $\mathbf{TSA}_{3:n}$. The bold indicates a vector of TSA values. The CIs of the EO are the standard statistics of the EO vector
- Narrowband Analysis: operates on the TSA by filtering out all frequencies except that of the gear mesh and with a given bandwidth. It is calculated by zeroing bins in of the Fourier transform of the TSA, except the gear mesh. The bandwidth is typically 10% of the number of teeth on the gear under analysis. For example, a 23 tooth gear analysis would retain bins 21, 22, 23, 24, and 25, and there conjugates in frequency domain. Then the inverse FFT is taken, and statistics of waveform are taken. Narrowband analysis can capture sideband modulation of the gear mesh frequency due to misalignment, or a cracked/broken tooth.
- Amplitude Modulation (AM) analysis is the absolute value of the Hilbert transform of the Narrowband signal. For a gear with minimum transmission error, the AM analysis feature should be a constant value. Faults will greatly increase the kurtosis of the signal
- Frequency Modulation (FM) analysis is the derivative of the angle of the Hilbert transform of the Narrowband signal. It's is a powerful tool capable of detecting changes of phase due to uneven tooth loading, characteristic of a number of fault types.

For a more complete description of these analyses, see (McFadden, 1985., Zakrajsek, 1993).

The analysis for the experiment used 17 CIs. In general, there is no consensus on which CIa are best, as different CIs seem to work better than other CIs depending on the fault type.

4. Experimental Test

The test was conducted on a split torque gearbox (STG). While not a planetary gearbox, the TSG similarly splits the torque path from a drive pinion to a driven gear. A full description of the article is available in (Li, 2012). The test consisted of the comparison of the nominal gears, with a idler shaft pinions that was missing 100% of a tooth (e.g. the "Bad Gear"). The idler shaft rate was rate was 0.556 x the input shaft, on which a 1/rev tachometer takeoff was mounted (Figure **3**).

A Physical Acoustic sensor, model: WD was used. This sensor is a wideband differential sensor with high sensitivity and bandwidth (100-900 KHz). The sensor was mounted on the output side of the gearbox, and after pre-amplification, was demodulate using a Analog Devices quadrature demodulator. The heterodyne frequency was 500 KHz. This frequency was optimized via testing using a Hsu-Neilsen source. The output of the demodulator and the tachometer 1/rev signal was sampled at 100 KHz using a 18 bit National Instruments data acquisition system. This represents an AE bandpass signal of 400 to 600 KHz, and an envelope bandwidth of 50 KHz. The gearbox was run at 60 Hz input shaft speed, where data was collected for 8 seconds per trail.

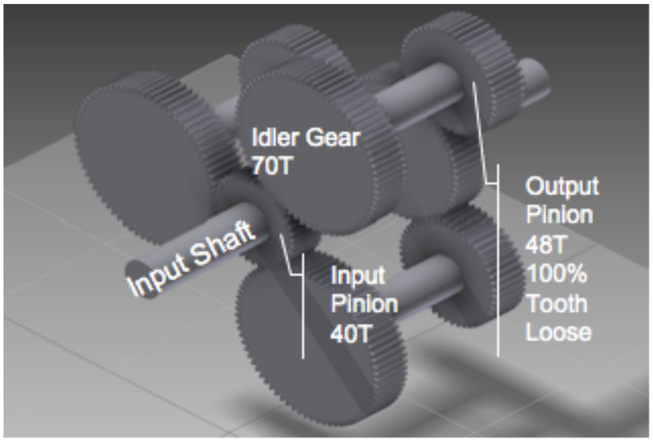


Figure 3 Split Torque Gearbox, Exploded View

4.1. Initial Results

The tachometer, shaft ratio and AE envelope signal was processed using the linear interpolation TSA algorithm presented in (Bechhoefer 2009). For the 8-second acquisition, the TSA had approximately 250 revolutions. The TSA length was 2ceil(log2(100000/(60*40/70)) = 4096 points. Some experimentation was performed where inline decimation was conducted to reduce the effective sample rate to 50000, with no loss in signal fidelity (Bechhoefer, 2012). Figure **4** displays the nominal vs. bad gear. The Bad Gear tooth fault is clearly visible when compared to the Nominal Gear.

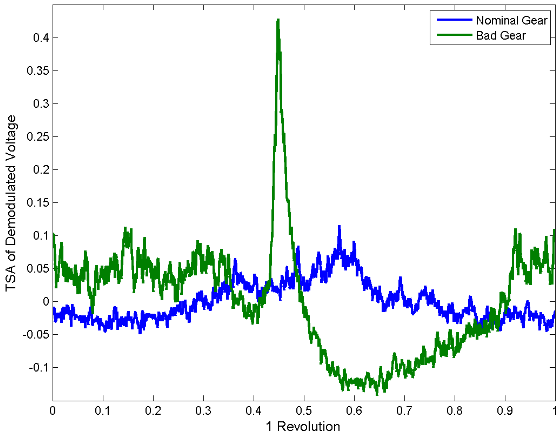


Figure 4 TSA of the AE Envelope: Nominal vs. Bad Gear

By removing the 1/Rev and gear mesh tone, the residual signal improves the fault visually (Figure **5**).

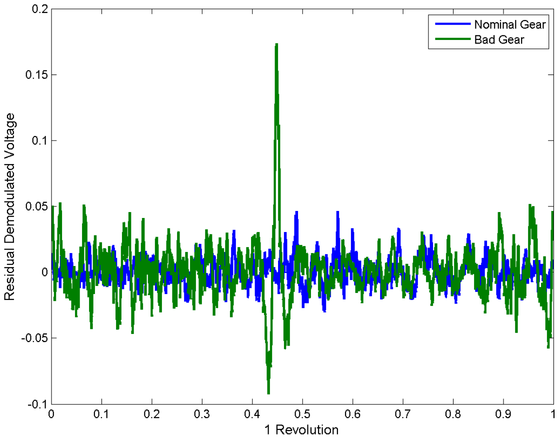


Figure 5 Residual of the AE Envelope

The one per revolution seems to be the result of an eccentric output gear, as the 1/Rev was present in both the nominal and damage gear TSA waveform. The Bad Gear fault is also clearly evident in the Energy Operator (Figure **6**). The waveform of the FM analysis shows a large phase reversal at the fault, which large Frequency modulation RMS/Peak to Peak (Table 1). The TSA, Residual, Energy Operator and Frequency modulation condition indicators are highly significant (Table 1).

It should be noted that without TSA, the raw spectrum of the demodulated AE signal indicated that the “Nominal Gear” was more damaged than the “Bad Gear”, as evident by a large 50 Hz frequency associated with the shaft rate. This was likely caused by the eccentric pinion – the fault was not discernable without the use of the TSA.

4.2. Quantifying Result Performance

To quantify the performance, the measure of seperatbility was calculated using the pooled sample standard deviation. The sample size was 5 acquisitions per trail, where the populations for the null set came from the nominal gear (no damage) and the alternative set came from the damage gear population.

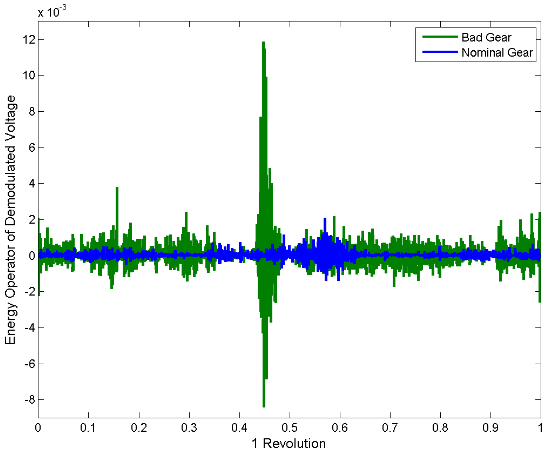


Figure 6 Energy Operator of the AE Envelope

The test statistics is then:

$$T = E[Y_1] - E[Y_2] \Big/ S_p \sqrt{2/n} \qquad (6)$$

where,

$$S_p = \sqrt{(n-1)S_1^2 + (n-1)S_2^2 / 2n - 2} \qquad (7)$$

A test statistic T greater than 3.58 is considered significant and would indicate that the CI could detect the fault (Wackerly, 1996). Note that AE is for the T statistics using the AE Envelope, while Vib is the T statistics from an earlier study using vibration data alone (Table 1).

Table 1. AE Envelop CI Algorithm Results

Condition Indicator	AE	Vib
TSA RMS	21.6	3
TSA Peak-to-Peak	9.2	4
FM0	3.25	
Sideband Modulation	3.14	4.3
Residual RMS	24.45	2.8
Residual Kurtosis	6.54	0.065
Residual P2P	15.4	2.75
Residual Crest Factor	6.91	1.24
Freq. Mod. RMS	6.65	0.22
Freq. Mod. P2P	4.14	0.616
Energy Operator RMS	33.26	3.5
Energy Operator P2P	7.74	2.4
Narrowband RMS	5.87	1.1
Narrowband P2P	6.6	0.1
Narrowband CF	5.24	7.1
Amp. Mod. RMS	6.04	1.1
Amp. Mod. P2P	6.9	3.1

These results are very encouraging. In general, the AE envelope T statistics is far more significant than the vibration based T statistic.

5. Conclusion

The AE envelope analysis show promises to be a powerful tool for gear fault diagnostics. By heterodyning the raw AE signal, it is possible to reduce the hardware resources and cost normally associated with AE processing. In this experiment, the acquisition-sampling rate of 100 KSPS was used on an AE sensor with a signal bandwidth of 600 KHz, using an analogy Hilbert transform circuit.

The AE envelope signal was then processed using time synchronous averaging (TSA). The TSA is commonly used with vibration-based diagnostics: this is the first time its use has been published using AE data. The TSA of the AE envelope was used to control for variation in shaft speed, and to reduce non-synchronous noise. The use of the TSA allowed the gear fault to be identified.

Condition Indicators, based on the TSA, were calculated for both the AE sensor and the for vibration sensor (accelerometer). The CIs for the AE enveloped signal were 3x more statistically significant than for the vibration sensor. This indicated that the combination of demodulated AE sensor data and the use of the TSA was superior for gear fault detection than traditional vibration/accelerometer sensors.

Currently, we are working on the deployment of a prototype AE sensor for application in wind turbines.

References

Liptai, R., Dunegan, H., and Tatro, C., (1969). Acoustic Emissions Generated During Phase Transformation in Metals and Alloys, Int. J. Nondestruct. Test. 1, 213

Liptai, R., Harris, D., Engle, R., and Tatro, C., (1970). Acoustic Emissions Techniques in Materials Research, Proceedings of the Symposium on Advanced Experimental Techniques in Mechanics of Materials.

Miller, R.K., Hill, E.v.K., and Moore, P.O., (2005). *Nondestructive Testing Handbook, 3rd Ed., Vol. 6. Acoustic Emission Testing*. Columbus, OH: American Society for Nondestructive Testing, p. 32

Barsoum, F. F., Suleman, J., Korcak, A., and Hill, E. V., (2009). Acoustic Emission Monitoring and Fatigue Life Prediction in Axially Loaded Notched Steel Specimens, J. Acoustic Emission, 27.

Abouel-seoud S. A., Lemosry, M., (2012). Enhancement of Signal Denoising and Fault Detection in Wind Turbine Planetary Gearbox Using Wavelet Transform, International Journal of Science and Advanced Technology, Volume 2 No 5 May.

Gu, D. S., and Choi, B. K., (2011). "Machinery Faults Detection Using Acoustic Emission Signal, Acoustic Waves - From Microdevices to Helioseismology, ISBN 978-953-307-572-3.

Horowitz, P., Winfield, H., (1998). The Art of Electronics, Cambridge University Press.

AD532 data sheet, "Internally Trimmed Integrated Circuit Multiplier", www.analog.com

McFadden, P. (1987). A revised model for the extraction of periodic waveforms by time domain averaging. Mechanical Systems and Signal Processing 1 (1), 83-95

Bechhoefer, E., Kingsley, M. (2009). A Review of Time Synchronous Average Algorithms. Annual Conference of the Prognostics and Health Management Society

McFadden, P., Smith, J., (1985), A Signal Processing Technique for detecting local defects in a gear from a signal average of the vibration. Proc Instn Mech Engrs.

ISO 10825. (2007) Gears -- Wear and damage to gear teeth -- Terminology

Zakrajsek, J. Townsend, D., Decker, H. (1993). An Analysis of Gear Fault Detection Method as Applied to Pitting Fatigue Failure Damage. NASA Technical Memorandum 105950.

Li, R., Seckiner, S. U., He, D., Bechhoefer, E., Menon, P., (2012) Gear Fault Location Detection for Split Torque Gearbox Using AE Sensor, IEEE Transactions on Systems, Man, and Cybernetics – Part C:, Applications and Reviews" IEEE 1094-6977.

Bechhoefer, E., Wadham-Gagnon, M., Boucher, B., (2012). Initial Condition Monitoring Experience on a Wind Turbine, PHM Society Annual Forum, Minneapolis, MN.

Wackerly, D., Mendenhall, W., Scheaffer, R.,(1996), Mathematical Statistics with Applications, Buxbury Press, Belmont.

Bechhoefer, E., Li, R., He, D., (2009). Quantification of Condition Indicator Performance on a Split Torque Gearbox, American Helicopter Society 65th Annual Forum, Grapevine, Texas, May 27-29

BIOGRAPHIES

Eric Bechhoefer received his B.S. in Biology from the University of Michigan, his M.S. in Operations Research from the Naval Postgraduate School, and a Ph.D. in General Engineering from Kennedy Western University. His is a former Naval Aviator who has worked extensively on condition based maintenance, rotor track and balance, vibration analysis of rotating machinery and fault detection in electronic systems. Dr. Bechhoefer is a board member of the Prognostics Health Management Society, and a member of the IEEE Reliability Society.

Yongzhi Qu received his B.S. in Measurement and Control and M.S. in Measurement and Testing from Wuhan University of Technology, China. He is a PhD student in the Department of Mechanical and Industrial Engineering at The University of Illinois-Chicago. His research interests include: rotational machinery health monitoring and fault diagnosis, especially with acoustic emission sensors, embedded system design and resources allocation and scheduling optimization.

Junda Zhu received his B.S. degree in Mechanical Engineering from Northeastern University, Shenyang, China, and M.S. degree in Mechanical Engineering from The University of Illinois at Chicago in 2009. He is a Ph.D. candidate at the Department of Mechanical and Industrial Engineering. His current research interests include lubrication oil condition monitoring and degradation simulation and analysis, rotational machinery health monitoring, diagnosis and prognosis with vibration or acoustic emission based signal processing techniques, physics/data driven based machine failure modeling, CAD and FEA.

David He received his B.S. degree in metallurgical engineering from Shanghai University of Technology, China, MBA degree from The University of Northern Iowa, and Ph.D. degree in industrial engineering from The University of Iowa in 1994. Dr. He is a Professor and Director of the Intelligent Systems Modeling & Development Laboratory in the Department of Mechanical and Industrial Engineering at The University of Illinois-Chicago. Dr. He's research areas include: machinery health monitoring, diagnosis and prognosis, complex systems failure analysis, quality and reliability engineering, and manufacturing systems design, modeling, scheduling and planning.

Decision Layer by Fusion of Diagnostic Algorithms

Jérôme Lacaille[1] and Tsirizo Rabenoro[1]

[1]*Snecma, 77550 Moissy-Cramayel, France*

jerome.lacaille@snecma.fr
tsirizo.rabenoro@snecma.fr

Abstract

For manufacturers systems monitoring or production equipment optimization solutions are founded on specific algorithms that produce low level local information about risk of degradation or production loss. In either case local results are combined in synthetic reports aimed to help decision taking at higher level. This work is about the description of an automatic fusion mechanism able to build expert output with direct understanding of the system behavior and help to infer causes of efficiency loss. An example application was built and tested in a semiconductor fab. The algorithms diagnosed yield degradation in different subsystems or work-area and were digested in a weekly report that highlighted the main production problems. We deployed the same methodology for condition based maintenance of aircraft engines on a test platform. The first part of this document sketches out some notations, the second part describes the semiconductor application and the conclusion is dedicated to the transfer in the aeronautic domain for the decision level of an engine fleet health monitoring system.

1. Methodology

Our decision implementation is based on a Bayesian framework (Yu, Cleary, Osborn, & Rajiv, 2007) but is also known as a mixture of experts (Yuksel, Wilson, & Gader, 2012). A decision process is based on available information with confidence levels. Health monitoring information is mainly the result of computations of diagnosis algorithms. This result, for an algorithm a, is the probability that a score S_a reaches a given level θ_a. We call this probability a risk, R_a in Eq. (1):

$$R_a = P(S_a > \theta_a) \tag{1}$$

We often complete this risk value with precision information σ_a computed as an estimation of the output variance or a square error obtained by a supervised cross-validation process.

Each algorithm produces values from a variety of experiments. An experiment is materialized by a set of observations. For example one observes successive flights of the same aircraft and registers all parameters during the last month; this leads to a dataset $X_{aircraft}([t - 1\,\text{month}, t])$. This dataset may be used by an algorithm that produces a risk $R_a(t)$. If we shift the observation window (one month) by one new day (about 10 flights), we obtain another lot of observations and the same algorithm will compute a new risk $R_a(t + 1\,\text{day})$. Others results may also be obtained by modification of the algorithm's parameters. For example a change in the input sources, a change of a scale parameter, etc. We finally get a whole population of available results at a given time.

1.1. Fusion operator

Many results are linked together and from the analyst point of view it may be interesting to transform raw risk data into something more accurate that better corresponds to the underlying problem.

A fusion operator, Eq. (2), is a function that transforms a collection of risk results with precision into a new risk with a higher meaning for the analyst.

$$R_{a'} = \Phi_{a'}(R_{a_1}, R_{a_2}, \dots, R_{a_n}) \tag{2}$$

This seems to be another way to increase the risk population. But this process may be represented graphically and leads to better interpretation. We progressively replace low level risk computations by synthesis about subsystems or components.

If we use the case of successive diagnostics, then a fusion operator may also be some diagnostic confirmation that exploits the list of preceding computations to ensure the presence of degradation (e.g. the probability that one observes 8 detections over the last 10 flights).

Jérôme Lacaille et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1.2. Implication coefficients

Looking at the algorithm's parameters, its input sources and the observed dataset, we may associate each risk result to system component fault f with a weighting vector. The implication value $\xi_a^f = P(F = f|S_a > \theta_a)$ may be interpreted as an a priori probability that fault f may occur when algorithm a result let us expect a risk of degradation.

The first implication vectors are defined by experts for initial algorithms which already are based on specific fault identification. It is more complex after some fusion operation. Moreover, as there may always be some dependencies between faults, even across different components, those implication vectors will be updated by a learning procedure using capitalization of experience.

1.3. Synthesis or projections

The decision process should use all available risk data. But the plurality of risk results makes it impossible. So we need to reduce the amount of information in a synthesis for each subsystem we want to monitor.

Suppose we have a component ζ that may be damaged according to a list of possible faults $f \in \zeta$. Let $\pi_f^\zeta = P(\zeta|f)$ the probability that fault f leads to a damage of component ζ. This value is often available from design and certification documents like the FMEA (Failure modes and Effect Analysis). A nice projection representing the risk of component ζ to be damaged may be given by Eq. (3):

$$R_\zeta = \sum_{f \in \zeta} \frac{\pi_f^\zeta}{\pi^\zeta} \sum_{a/\xi_a^f \neq 0} \frac{\xi_a^f}{\xi^f} R_a \qquad (3)$$

This proposition is just a simple example using a logical weighted sum without taking care of faults and algorithms dependencies. Normalization constants π^ζ and ξ^f are used to ensure that the weighted sum has a value between 0 and 1 and keeps a meaning as kind of risk. Different fusion methods may also be used (like for example taking the highest detection). Eq. (4) below can be used to update the implication coefficients by a supervised learning procedure.

Independency of scores may be reached by designing algorithms related to subsystems. For example, in a fabrication process, if different but equivalent tools are used, local algorithms using only local measurements are independents. Using independent sets of faults and corresponding algorithms may lead to an exact formulation of a probability that a component is defective. Methodologies exist to group results into independent variables (Alhoniemi et al., 2007). Our actual goal is to converge to an approximate formulation with a Bayesian neural network. Other approaches may use dependency matrixes (Singh & Holland, 2010) or decision trees (Ricordeau & Lacaille, 2010) but at the end we always get a parametric model we may update with new experiments.

Fusion algorithm with risk result R_ζ is associated to the component ζ. If needed, we may compute implication values ξ_ζ^f from Bayes rule: $\xi_\zeta^f = \pi_f^\zeta P(f)/P(\zeta)$, the latter two probabilities are known and updated by maintenance people: $P(f)$ is the occurrence rate of fault f and $P(\zeta)$ is the repair or change rate of component ζ. The computation of this new implication coefficient serves the only purpose of being able to chain the analysis drawing a hierarchic graph of decisions from subsystems' components to higher subsystems and the global system that is monitored.

1.4. Confidence and efficiency

The main goal to achieve in decision making is gaining expert confidence. An optimization process manipulating the risk population is driven by this confidence. Gaining confidence is assured by a measure of the efficiency of the synthesis results.

Each expert may be viewed as a "human" or test algorithm e. So each expert produces one risk R_e usually linked to specific component's faults. Implication of the expert ξ_e is straightforward and given by the expert himself. However, an expert is subject to mistakes and so his precision σ_e is unknown and should be estimated. Usually, expert risk is a binary result: either faulty $R_e = 1$ or healthy $R_e = 0$.

In datamining processes, the expert result is known as supervision information. This supervision is used to build an efficiency measure like the mean-square error in Eq. (4).

$$\eta_a^2 = \sum_e \frac{(R_a - R_e)^2}{\sigma_a \sigma_e} \langle \xi_a . \xi_e \rangle \qquad (4)$$

This value gives for any algorithm a measure of the adequacy between expert knowledge and the algorithm computation. It may be calculated for any component projection R_ζ and measure a confidence level for a decision about the health of this component.

2. SEMICONDUCTOR FAB YIELD LOSS ANALYSIS

An experiment at the ST-Microelectronics 8" fab in Rousset (France) showed that a great deal of information measured during the manufacturing process had a direct impact on the output yield. An influence analysis (measures of entropy and mutual information) measuring the stochastic dependences between the yield and automatically selected combinations of measurements proved the need to develop a prototype based on defectivity measurements.

A pilot project for collecting defectivity measurements and information on the routing of the wafers in the fab ran in real time at the Rousset fab (Lacaille & Dubus, 2005; Lacaille, 2008). It has highlighted many causes of wafer deterioration. Engineers and defectivity operators where able to automatically generate synthesis reports to supervise equipment health and production quality (Lacaille, 2005).

2.1. Microfabrication process

In a semiconductor fab, wafers are produced with a high rate (more than 1000 wafers a week). A single wafer is processed during almost three months and different products are manufactured at the same time by the same equipments. Wafers are grouped together in lots of 25, each lot is carried in a pod and all wafers in a pod are of the same kind. Sometime a lot is inspected, specific metrology equipments observe 4 to 6 wafers in this lot and we get defectivity measurements. Almost 20% of the lots are inspected for defectivity observations. This procedure allows only a very small proportion of the production to be observed. Moreover as defectivity operations are not productive steps, operators bypass them.

The defectivity metrology measures defects and produces observations like a number of defects of a minimal size, a defect density and a proportion of dies affected by the defects on the wafer area. Those defects are responsible of almost 80% of the yield loss in an industrial fab for a mature product.

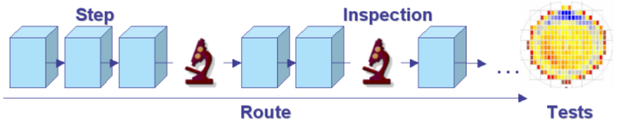


Figure 1 - Inspection steps are dispatched over the route of the wafers in the fab. The yield is computed at the end of the production by test steps.

In the 8" fab of ST in Rousset, defectivity operation steps are put on the wafer/product route but can be moved randomly to improve the metrology sampling on specific parts of the production (Figure 1). The route of the lot is determined by the product recipe but for each production step, each layer, the specialized operator can choose randomly between similar but certified equipments in his work-area.

To analyze the overall fab production in ST Rousset 8" it was necessary to model the random trajectories of wafer pods within the park of equipments. The stochastic nature of these trajectories is the result of the availability of similar and interchangeable equipments per work-area.

On those trajectories (simplified by successive arrows on Figure 2), defectivity steps appear like some measurement points (red dots) on a very complex curve. And route information, WIP (wafer in process) data, can provide observations on the same curve. For each step and each wafer we observe route information: date, equipment, recipe of the equipment, delay from the last step, slot position in the pod and sometime metrology information like defect density or defective dies. Route information is systematic while metrology is random.

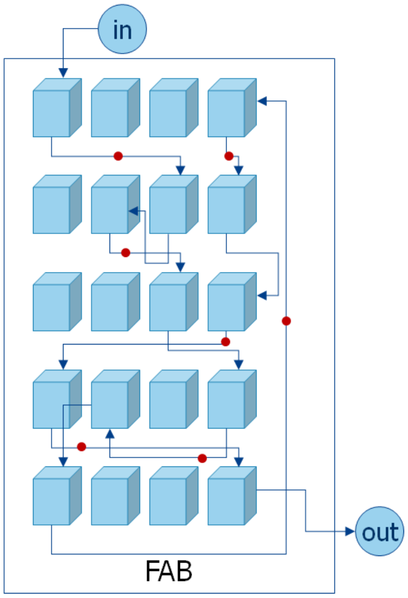


Figure 2 – Trajectories of wafers in the fab are complex curves (arrows) on which specific defectivity measurement points (red dots) are set.

The yield is measured at the end of the production in a specific back-end area by an electrical batch of tests. The yield information, results of those tests, almost corresponds to a proportion of surviving dies on the wafer.

2.2. Diagnosis methodology

The solution exploits a population of micro-algorithms: the agents. Those algorithms try to estimate a low-yield risk using some measurements taken on the wafer trajectories. This risk output R_a for each agent a is the probability of a low yield output.

Each agent uses only few points of measurements, thus ensuring the robustness of its calculation. In fact, the small analyzes carried out independently are not sufficient to explain the behavior of the fab, but a global information with a high degree of accuracy emerges from the entire population of local predictors.

Each agent uses some defectivity measurements taken during the route of the pods and/or process information taken from the tools (temperatures, pressures, etc.) and/or logistic data like equipment-id, receipt, inter-operation delays… The largest agent uses no more than 6 entries. The very large number of available measurements implies the existence of a lot of possible combinations of small number of measurements. The software solution solves this problem by using a population of agents distributed on a network of computers. This population evolves every day to adapt to the fab production.

This methodology is an implementation of machine ensemble theory (Figueiras-Vidal & Rokach, 2012) which goal is to model complex functions using an ensemble of small data-driven components. One of the reference work in this domain is (Jacobs, Jordan, Nowlan, & Hinton, 1991).

2.2.1. Neural networks agents

The agents are self-adapting algorithmic components. Each one of them contains a batch of neural-networks regularly calibrated in real time. Each agent uses about fifty neural-networks (Figure 3). Each network inputs are the same inputs given to the agent; it is a small selection of WIP measurement points in the fab: defectivity, process values, and logistic data which are measured for each wafer. The neural-network models a relation between its inputs data and the wafer yield, the quality measure obtained by electric tests at the end of the fabrication process. Prediction error on calibration set is a quality indicator of the estimator. The wafer set used for calibration is a subset of the past processed wafers so the difference between two neural-networks comes from the “random” selection of the subset of wafers (see below). The final response of the agent is given by an estimation Y_a of the yield from a vote of a selection of the 20 best neural-networks. Then each agent a produces a low yield score $R_a = P(Y_a < \theta)$ and associated quality indicator σ_a as a final estimation of Y_a precision. The threshold θ is fixed according to the current fab expectation but may be adapted specifically for each product.

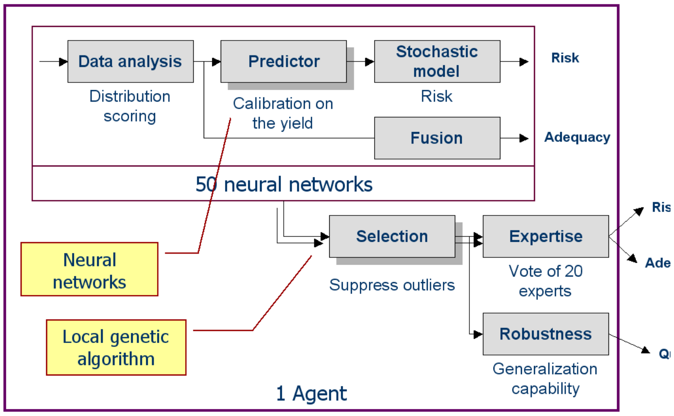


Figure 3 – Description of the inside of the agent. 50 neural networks, essentially multi-layer perceptrons with 3 layers, predict the yield from given parameters. Each neural network is learnt on a different training set ensuring accessibility to local behavior models. 20 among the best predictors are kept and used simultaneously in a voting process.

In this specific implementation the experts’ skill is limited to the electric tests giving the real yield Y_0 and risk $R_e = P(Y_0 < \theta)$ for each processed wafer ($\sigma_e = 1$ or may be adapted by product). It is possible to compute an efficiency measure η_a from Eq. (4). This measure is used by the population optimization algorithm (Figure 4) to select the best agents.

The input observations sent to each neural network for calibration may be product-specific. Some agents work with all kind of products but the majority is specialized. A classification algorithm automatically selects the set of old products adapted for an agent based on the prediction efficiency on past measurements. Then new products are compared with preceding technologies using an unsupervised clusterisation algorithm to initiate implication values between agents and new products.

2.2.2. Agent population optimization

To make the agent population relevant at each time when the fab production evolves, we implement a genetic algorithm (Figure 4): the agents communicate together and exchange information on the relevance of each one of their inputs. The agent inputs are exchanged when agents meet together. The whole population is really moving over the fab and can be seen as a sort of swarm. The population of agent evolves gradually and it emerges an increasingly relevant response while leaving the system able to integrate each day some new characteristics of the fab production.

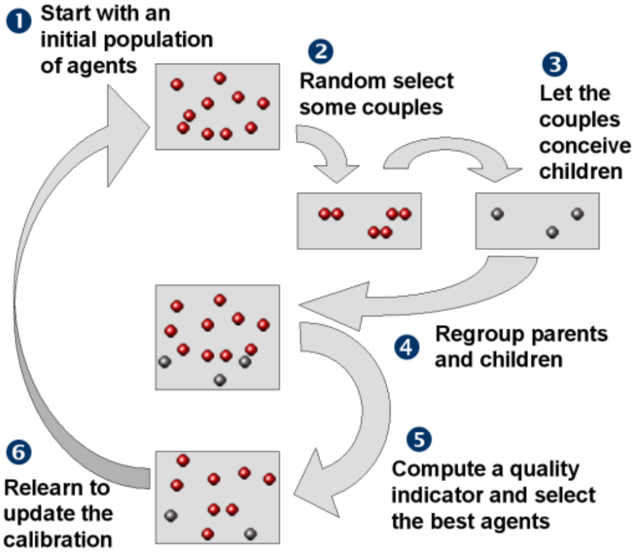

1 Start with an initial population of agents
2 Random select some couples
3 Let the couples conceive children
4 Regroup parents and children
5 Compute a quality indicator and select the best agents
6 Relearn to update the calibration

Figure 4 – A genetic algorithm optimizes the agent population.

The genetic algorithm optimizes the quality of the agent population but it has to respect hard constraints to ensure a homogeneous repartition of the agents over the whole fab and all products (see Figure 5).

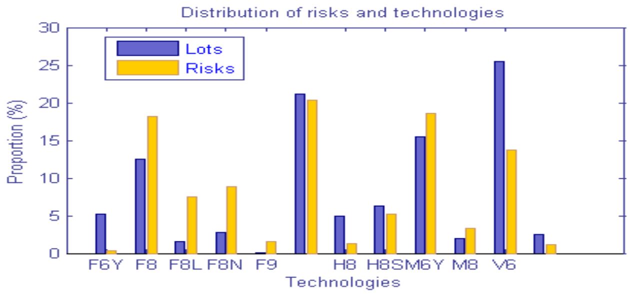


Figure 5 – A constraint forces the agents' population to be dispatched with a similar repartition to the production.

2.3. Decision help by fusion of diagnostics

A homogeneous population of agents emitting each one the same type of information replaces the complexity of the trajectories of the pods of wafers. The outputs of each algorithm are a risk of "low yield" R_a and a measurement of reliability η_a (Figure 6). Thus one can produce dashboards measuring the health of the equipments of work-areas, anticipating the quality of production and locating the equipments responsible for crisis and deterioration of the quality (Figure 8, Figure 9 and Figure 10).

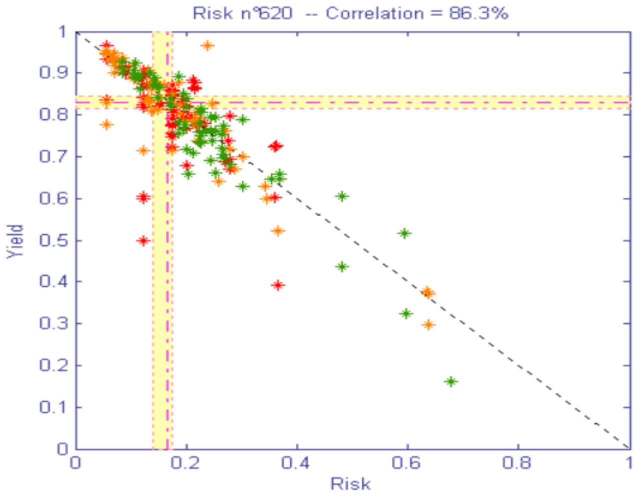


Figure 6 – Output of an agent. The risk (x-axis) is drawn with the yield (y-axis) for tested wafers. The green stars are computation of more than 66% of reliability, the orange have an reliability between 33% and 66% and the red ones are less than 33% of reliability.

Figure 7 schematizes the production of high level outputs from the original risk computations. The following figures are examples of such fusion algorithms.

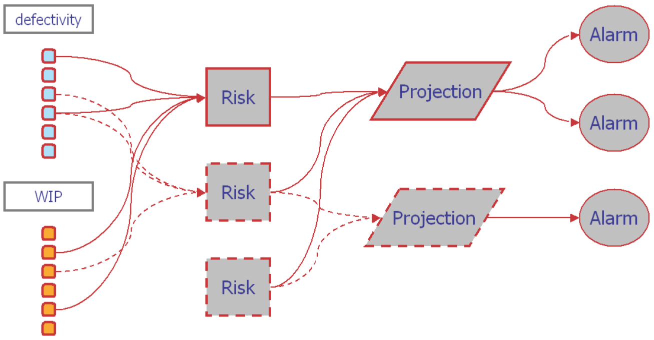


Figure 7 – Organization of the statistic computations. Risks are only intermediate results used by high-level fusion projectons.

Figure 8 is a fusion over an etch work-area. Here the fusion sums risks of low yield estimated for each wafer going through a given etch tool. For each tool, the summation is weighted according to Eq. (3) (with just one fault corresponding to the degradation of a die causing a decrease of the yield) and using implications that are high if the agent use information acquired on the given equipment, lower if not but if the wafer was processed by this equipment (depends on the distance on the wafer's route).

Figure 9 shows the evolution over time of the low yield estimation for a given tool using the same preceding computation. This graph helps to understand how to build alerts when a local risk crosses a maximum threshold or how to detect tendencies and eventually anticipate a crisis.

Figure 10 presents a synthesis for the worse equipments within all work-areas of the fab with recall of the past evolution.

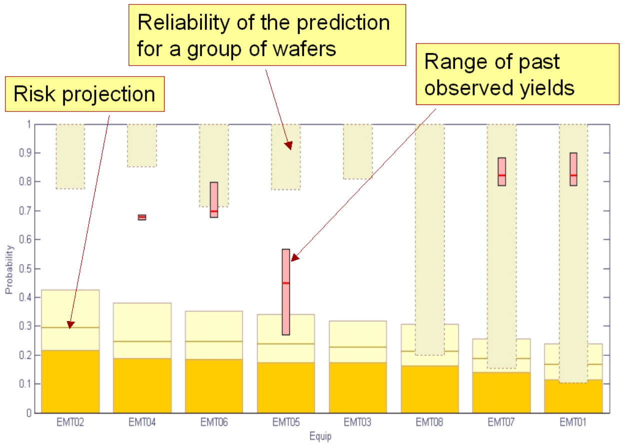


Figure 8 – High-level and detailed risk analysis of the equipements in a work-area. One have imediately a snapshot of the efficiency of each tool.

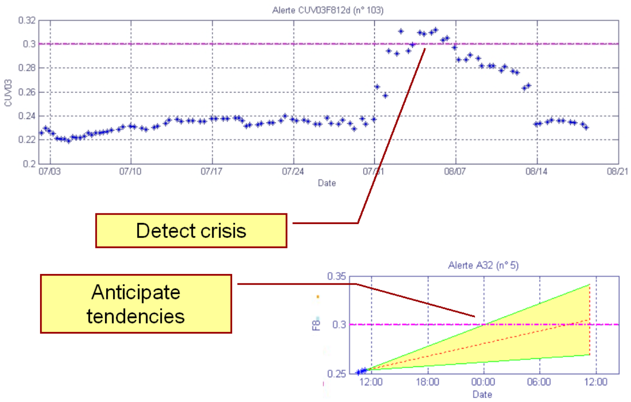


Figure 9 – Real-time survey of equipment. Alerts are automatically sent to operators.

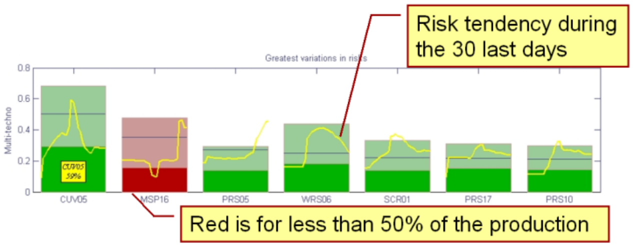


Figure 10 – Global analysis across all work-areas. All equipments of the fab are sorted by decreasing risks or decreasing risk-variations. The graph can also be filtered by work-areas or technologies.

2.4. Automatic inline scheduling

This whole system is entirely distributed over a cluster of computers (Figure 11). The agents are independent and their computations can be done in parallel. Only a central database for data collection (DC) is needed but it can be accessed simultaneously by different requests.

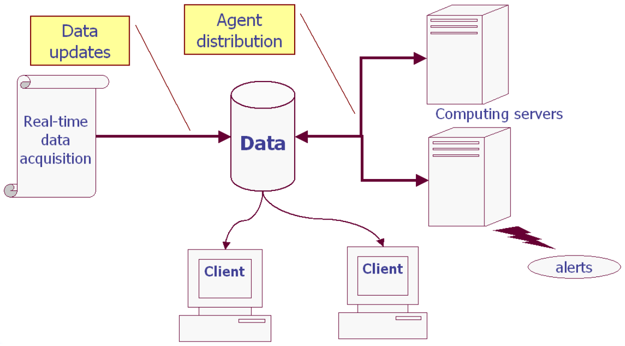


Figure 11 – Distributed architecture of the system.

The computation uses two internal cycles (Figure 12).

- The on-line scheduling cycle is based on the acquisition of new measurements. The new data are automatically pushed to the system by an ftp channel and the loader program automatically detects the arrival of new information.
- The genetic algorithm and the maturity of the agent population drive the internal scheduling cycle.

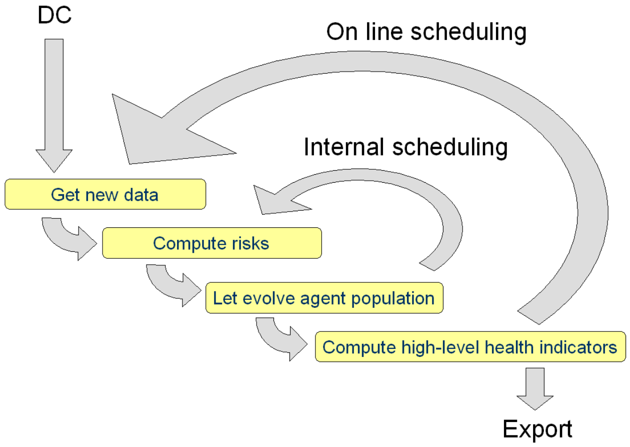


Figure 12 – System scheduling. The system is working non-stop ensuring a maturity of the agent population when the production evolves.

3. Engine Condition Maintenance

Advanced health monitoring is becoming a standard for new engine applications, in order to enable in-service event reduction and engine maintenance optimization. The goal is to reduce operational events such as IFSD (In Flight Shut Down), ATO (Aborted Take-Off), D&C (Delay & Cancellation) and to substitute them with maintenance operations that are planned long enough in advance in order to minimize their operational impacts for the airlines. IFSDs and ATOs are very seldom but still stressing for the pilots and they often produce secondary damages that might increase reparation costs. D&Cs are usually not critical but occur more often; their consequences can be traffic disorganization, customer dissatisfaction. They are partially linked to procedures and controls to perform troubleshooting (Lacaille, 2012; Ricordeau & Lacaille, 2010).

The performance of engine health monitoring functionalities is driven by the capability to model the engine behavior and to identify engines on healthy or unhealthy conditions. Methodologies based on fusions of diagnostics applied to aerospace monitoring may be find in (Tang et al., 2009) for optimization of remaining useful life (RUL) estimations; in (Klein, Rudyk, & Masad, 2011) for the identification of bearing faults using vibration or acoustic signatures extracted from multiple microphones or accelerometers; and in (Romessis, Kyriazis, & Mathioudakis, 2007) for performance analysis of the turbofan engine.

3.1. Health monitoring algorithms

Algorithms written for health monitoring purpose in Snecma are decomposed in two parts: the first one is made of embeddable code aimed to produce health indicators from engine raw measurements. Those computed indicators are sent to the ground with context information coming from the aircraft computer.

The ground application hosts the scoring process (Figure 13). The scoring process transforms health indicators into risk of abnormality R_a . This algorithm first uses a normalization phase to suppress the context dependency (flight specification like altitude, speed, weather conditions …) The second phase is either a model of normality which produces a likelihood to measure a range from normal behavior or an identification process using physical knowledge of the monitored system to target specific faults. At the end we have generic novelty detectors implying a subsystem in general or specific fault diagnostics (Lacaille & Nya Djiki, 2010; Lacaille, 2009, 2010).

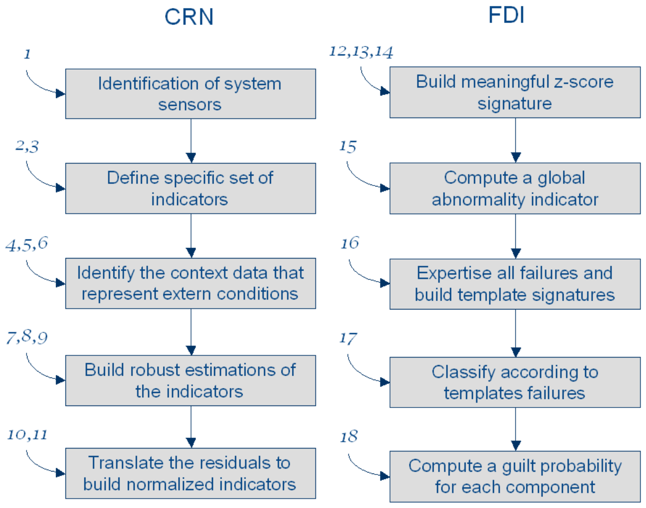

CRN
FDI
1
Identification of system sensors
2,3
Define specific set of indicators
4,5,6
Identify the context data that represent extern conditions
7,8,9
Build robust estimations of the indicators
10,11
Translate the residuals to build normalized indicators
12,13,14
Build meaningful z-score signature
15
Compute a global abnormality indicator
16
Expertise all failures and build template signatures
17
Classify according to templates failures
18
Compute a guilt probability for each component

Figure 13 – Two phases of the ground diagnostic process. CRN (Context Removal and Normalization) suppress flight dependencies and FDI (Fault Detection and Identification) detects unusual behavior and try to identify the specific faults of a subsystem.

Precision σ_a of each algorithm is given by an estimation of the diagnostic error computed on a supervised dataset (Figure 14).

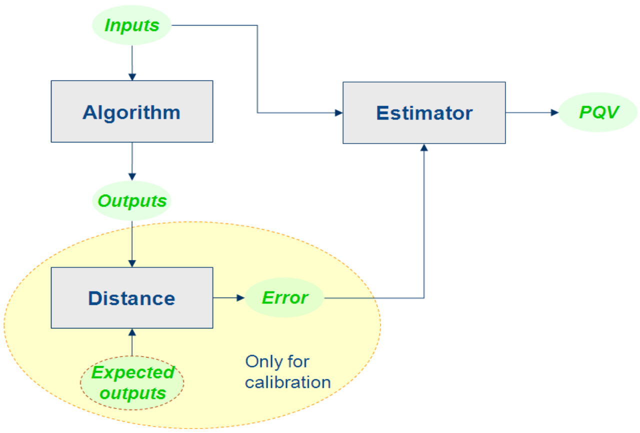


Figure 14 – Precision quality value (PQV) of the algorithms computed by an estimator learned on a supervision dataset.

The engine start capability (ESC) algorithm is a good application example (Flandrois, Lacaille, Masse, & Ausloos, 2009). Different subsystems and components are involved (Figure 15): the auxiliary power unit (ACU) that provides external pressure to the starter, the admission valve, the fuel pump, the fuel metering unit (FMU), the ignition system, etc.

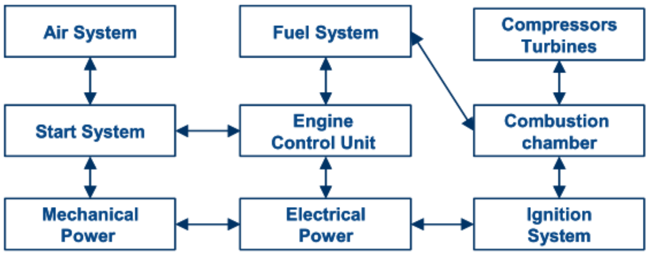


Figure 15 – Subsystems parts of the start system of a turbofan engine.

The health indicators are extracted from the raw temporal measurements with a “specific instant detection algorithm”. They are mostly time delays (Figure 16).

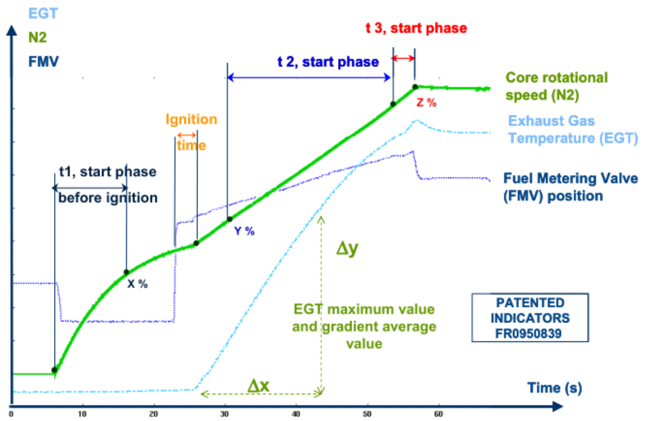


Figure 16 – Health indicators for the ESC algorithm.

Normality scores are computed for each indicator or coherent group of indicators if a multivariate observation is needed. At the end we build 8 scores listed in the following Table 1:

Table 1 – List of scores used for degradation risks computation of the ESC application.

Ref	Indicators
I1	APU Air Pressure
I2	Ignition time
I3	Exhaust Gas Temperature (EGT) gradient average value
I4	Exhaust Gas Temperature (EGT) maximum value
I5	t1 : start phase before ignition
I6	t2 : start phase between X % of N2 nominal value to cut out (SAV off)
I7	t3 : start phase to cut out to Y% of N2 nominal value
I8	N2 maximum acceleration (x,y)

3.2. Decision methodology

Decision step (on the ground) takes inputs from the set of risks generated by each specific algorithm. The specificity of algorithms comes from their design: there exist algorithms to detect each pump fault, gear wear, damaged bearings, oil leakage, filter clogging, intermittent sensors, etc. The faults relative to each subsystem produce comparable effects so the risk measurements are highly dependent.

The decision layer is actually built following the scheme presented in section 1 and derived specifically for a semiconductor fab in section 2. But in the case of aeronautic data we have a very small number of observed degradations, it is why the algorithms are mostly based on normality scores and their efficiency measured on a small number of observations. The implications come from expert knowledge but are progressively adapted using troubleshooting tools that records new data.

For the engine start capability (ESC) algorithm, the implication coefficients are roughly initialized by experts (Table 2). Decision on each impacted component is computed by fusion knowing the relation between faults and components.

Efficiency of the results is measured using maintenance reports on which identification of the problem is given after repair. Maintenance results are not always known by the manufacturer but as engine's design is completely mastered by the company; experts systematically give prognostics about the engines conditions. This supervision helps improving the implication values by a learning procedure.

Figure 17 shows an example of fusion of ESC detection algorithms. This fusion gives information about all components used in this process: APU, starter, valves, etc. This image presents the results of an artificial scenario where several defects were simulated sequentially.

The final decision is obtained only after some confirmation process that uses successive risk computations to solve incompatibilities and increases likelihood by a temporal confirmation.

Table 2 – Implication coefficients between faults and indicators. Values are roughly proposed by experts.

Failures	I'1	I'2	I'3	I'4	I'5	I'6	I'7	I'8
F1	0	0	0	0	0	NaN	NaN	0
F2	0	+1	+2	0	0	+1	+1	0
F3	+1	0	-1	0	0	0	-1	0
F4	0	0	0	-2	0	-1	-2	0
F5	0	+2	0	0	0	0	0	0
F6	0	0	0	0	NaN	NaN	NaN	0
F7	0	0	0	0	0	NaN	NaN	0
F8	-2	0	0	0	0	0	NaN	0
F9	0	-1	0	0	0	0	+2	0

+2	Very higher degradation as usual
+1	Higher degradation than usual
NaN	Not a Number
-1	Lower degradation than usual
-2	Very lower degradation than usual

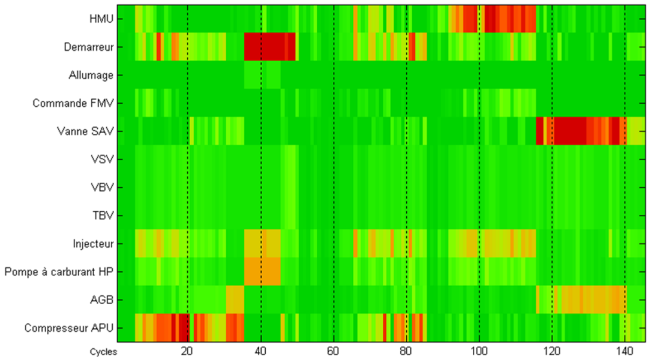


Figure 17 – Synthesis of the decision process for the different components that impacts the engine start system. Here the computed risk corresponds to the probability of a component failure leading to a D&C.

3.3. Conclusion

There is still a lot of freedom in the parameters used for normalization and identification algorithms as well as in size and shape of the confirmation by fusion of past computations. The methodology presented in the first section, applied in the second on a manufacturing process is compatible with engine health monitoring and is currently tested on Snecma's health monitoring research platform.

NOMENCLATURE

APU	Auxiliary Power unit
ATO	Aborted Take-Off
CRN	Context Removal and Normalization
DC	Data Collection
D&C	Delay and Cancellation
ESC	Engine Start Capability
FDI	Fault Detection and Identification
FMEA	Failure Mode and Effect Analysis
IFSD	In Flight Shut Down
PQV	Predictive Quality Value
WIP	Wafer In Process

REFERENCES

Alhoniemi, E., Honkela, A., Lagus, K., Seppä, J., Wagner, P., & Valpola, H. (2007). Compact modeling of data using independent variable group analysis. *Transactions on Neural Networks*.

Figueiras-Vidal, A. R., & Rokach, L. (2012). An Exploration of Research Directions in Machine Ensemble Theory and Applications. In *ESANN* (pp. 25–27). Bruges.

Flandrois, X., Lacaille, J., Masse, J.-R., & Ausloos, A. (2009). Expertise Transfer and Automatic Failure Classification for the Engine Start Capability System. In *AIAA InfoTech*.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive Mixtures of Local Experts. *Neural Computation*, *3*(1), 79–87. doi:10.1162/neco.1991.3.1.79

Klein, R., Rudyk, E., & Masad, E. (2011). Decision and Fusion for Diagnostics of Mechanical Components. In *PHM* (pp. 1–9). Montreal (Canada): PHMSociety.

Lacaille, J. (2005). Mathematical Solution to Identify the Causes of Yield Deterioration. In *International Sematech Manufacturing Initiative (ISMI)*. Austin, TX: Sematech.

Lacaille, J. (2008). Global Predictive Monitoring System for a Manufacturing Facility. PDF Solutions. Patent US 2008082197 A1.

Lacaille, J. (2009). Standardized failure signature for a turbofan engine. In *IEEE Aerospace conference* (p. 11/0505). Big Sky (MT): IEEE Aerospace society. doi:10.1109/AERO.2009.4839670

Lacaille, J. (2010). Identification of Defects in an Aircraft Engine. Patent WO 2010076469 A1.

Lacaille, J. (2012). Validation Environment of Engine Health Monitoring Algorithms. In Snecma (Ed.), *IEEE Aerospace conference*. Big Sky (MT).

Lacaille, J., & Dubus, H. (2005). Defectivity Analysis by a Swarm of Intelligent Distributed Agents. In *AEC-APC*. Indian Wells (CA), Sematech.

Lacaille, J., & Nya Djiki, R. (2010). Detection of Anomalies in an Aircraft Engine. Patent WO 2010061136 A1.

Ricordeau, J., & Lacaille, J. (2010). Application of Random Forests to Engine health Monitoring. In *ICAS*.

Romessis, C., Kyriazis, A., & Mathioudakis, K. (2007). Fusion of Gas Turbines Diagnostic Inference - The Dempster-Schafer Approach. In *ASME & GT* (Vol. 1). Montreal.

Singh, S., & Holland, S. W. (2010). Trends in the Development of System-Level Fault Dependency Matrices. In *IEEE Aerospace conference*. IEEE Aerospace society.

Tang, L., Kacprzynski, G. J., Goebel, K., Vachtsevanos, G., Odel, B. A. M., & Echniques, S. E. T. (2009). Methodologies for Uncertainty Management in Prognostics. In *IEEE Aerospace conference*. IEEE Aerospace society.

Yu, L., Cleary, D., Osborn, M., & Rajiv, V. (2007). Information Fusion Strategy for Aircraft Engine Health Management. In *ASME & GT* (pp. 1–8). Montreal.

Yuksel, S. E., Wilson, J. N., & Gader, P. D. (2012). Twenty Years of Mixture of Experts. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(8), 1177–1193. doi:10.1109/TNNLS.2012.2200299

BIOGRAPHY

Jérôme Lacaille is a Safran emeritus expert which mission for Snecma is to help in the development of mathematic algorithms used for the engine health monitoring. Jérôme has a PhD in Mathematics on "Neural Computation" and a HDR (habilitation à diriger des recherches) for "Algorithms Industrialization" from the Ecole Normale Supérieure (France). Jérôme has held several positions including scientific consultant and professor. He has also co-founded the Miriad Technologies Company, entered the semiconductor business taking in charge the direction of the Innovation Department for Si Automation (Montpellier - France) and PDF Solutions (San Jose - CA). He developed specific mathematic algorithms that where integrated in industrial processes. Over the course of his work, Jérôme has published several papers on integrating data analysis into industry infrastructure, including neural methodologies and stochastic modeling.

Tsirizo Rabenoro is currently a PhD student in statistics at Pantheon-Sorbonne University. He is also working at Snecma SAFRAN, as part of a research team involved in the operational monitoring. He is currently studying integration of mathematical methods into board and ground engine monitoring systems.

A Fusion Framework with Nonlinear Degradation Improvement for Remaining Useful Life Estimation of Lithium-ion Batteries

Datong Liu, Limeng Guo, Jingyue Pang, Yu Peng

Department of Automatic Test and Control, Harbin Institute of Technology, Harbin, 150080, China

liudatong@hit.edu.cn
glm_hit1991@163.com
pjy19880909@163.com
pengyu@hit.edu.cn

ABSTRACT

Fusion prognostic framework for lithium-ion battery remaining useful life (RUL) estimation has become a hot spot. Especially, the cycle life prediction has been conducted widely, for which many prognostic methods have been proposed correspondingly. However, many fusion frameworks which can achieve high precision are accompanied with high computing complexity and high time consumption which makes these methods low real-time performance. Either, some widely used prediction models with low complexity are weak to handle the nonlinear degradation features. To solve these problems, a fusion framework is proposed combining the model-based extended kalman filter (EKF) and the data-driven improved nonlinear scale degradation parameter based autoregressive (NSDP-AR) models. The proposed approach takes advantage of the state tracking ability of EKF algorithm to define the specific state transition model for the battery sample. Meanwhile, NSDP-AR model which contains the degradation features of each period is to promote the universality of the ND-AR (Nonlinear Degradation Autoregressive) model. NSDP-AR model is used to obtain the long term trend prediction results which are adopted as the observation data. Finally, a combination is made to realize the RUL prediction under the kalman filter (KF) system, which is an improvement to meet the practical applications. Experimental results with the battery test data from NASA PCoE and CALCE show that the fusion prognostic framework can predict the lithium-ion battery RUL with high efficiency and accuracy.

1. INTRODUCTION

Lithium-ion (Li-ion) batteries have become the preferred energy solution for various electrical-driven systems such as consuming electronics, electric vehicles, and even the aerospace field due to the high energy density, high galvanic potential, lightness of the weight and long lifetime compared to traditional energy storage batteries (He, Williard, Osterman & Michael Pecht, 2011). However, no matter how excellent the performance of the lithium-ion battery is, it degrades over time for aging, environmental impacts, and dynamic loading (Zhang & Lee, 2011). In order to satisfy the increasing demand for operation reliability, study on the effective methodologies for battery performance evaluation becomes considerably necessary and important. In particular, the remaining useful life (RUL) estimation of the Li-ion battery is the essential part in the field of electronic prognostics and health management (PHM). RUL also can be named remaining service life or residue life, which refers to the available service time (always using how many charge and discharge cycles the battery can experience to describe this variable) left before the degradation level of the system is unacceptable (Zhang & Lee, 2011). Successful RUL prediction is highly desirable for ensuring reliable system operation.

Recently, extensive research activities have been conducted on the RUL estimation of Li-ion batteries. Generally speaking, prognostics methods can be classified into data-driven and model-based approaches. Artificial Neural Networks (ANN) (Liu, Saxena, Goebel, Saha & Wang, 2010) and Relevance Vector Machine (RVM) (Zhang & Lee, 2011) are typical representatives for data-driven approaches which establish the prediction model using the characters selected from the data without considering the physical system features. Model-based approaches focus on the state space model studies which proceed from the system characteristics like Extended Kalman Filter (EKF) (He, Williard, Osterman & Pecht, 2011) and Particle Filter (PF) which can obtain both the RUL and uncertainty representation (Saha & Goebel, 2009).

Datong Liu et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Current researches show that there are several limitations in the above two kinds. Although the empirical degradation model is proposed on the basis of a large amount of experiments and rigorous analyses, it still cannot take into account all the complicated online operation conditions. That is to say, the performance of model-based method which is determined by the accuracy of the model cannot adapt to various actual working conditions. Meanwhile, estimation of the parameter in a specific model is an uneasy work and satisfied results are rarely obtained. Even though there is little such problem in data-driven approaches, they still suffer from the drawback for ignoring the distinctions among different types of systems in which the data sets belong to.

In order to address the aforementioned problems, data-model-fusion prognostic frameworks are proposed and attract more and more attention. Till now, various studies have been conducted in this field. A fusion prognostic framework with model-based PF algorithm and data-driven neural networks method has been proposed by Liu etc for dynamic system state forecasting and achieved a successful result for battery RUL estimation (Liu, Wang, Ma, Yang & Yang, 2012). Fusion frameworks combining data-driven and model-based prediction approaches can overcome the aforementioned shortcomings. With adoption of the data features obtained from the data-driven approach, dependence of the model-based method on the empirical model is alleviated. Meanwhile, system characteristics are added into the data-driven framework. In short, both the system and data features are contained in the fusion framework whose effect is expected to be better than any single method.

For online real-time applications, the accuracy as well as the real-time performance of the prognostic algorithm is crucial. However, majority of the fusion framework focuses on the accuracy without considering the efficiency and the calculation complexity. In order to satisfy the rigorous requirements in practical forecasting, a data-model-fusion framework with a strong emphasis on real-time prediction capability is worthy of more studies.

EKF algorithm used for the nonlinear systems is an extension of the Kalman Filter (KF) which is a recursive solution under the least-squares principle. EKF is one kind of stochastic filtering method based on the state space model which stands for the system features. With linearization of the system equations using Taylor expansion and cycles of state estimation and updating, EKF can provide an efficient computational solution even when the precise nature of the modeled system is unknown (Welch & Bishop, 1995). EKF has a good state tracking ability in the capacity prediction and the RUL can be obtained when the failure threshold is provided (He, Williard, Osterman & Pecht, 2011). Meanwhile, massive applications of the performance estimation for the lithium-ion batteries show that EKF is a promising algorithm with low computational complexity, strong real-time estimation ability and satisfied prognostic effect (Zhang & Lee, 2011).

Although such advantages have been shown in practical applications, EKF still has the limitation as a model-based approach. As discussed before, adoption of a data-driven method can effectively address this problem. Various data-driven methods such as machine learning and artificial intelligence have a satisfied prediction effect but also with a complicated calculation process and a lot of time consumption which make these algorithms have less practical value in real-time battery performance estimation especially when implemented in hardware. Due to such a practical concern, a much simpler data-driven method that has been applied in many prediction fields, namely the AR (Auto Regressive) model, draws our attention. AR model is suitable for the real-time estimation with small data sets (Wei, 1994). This combination is expected to possess better efficiency for real-time prognostic applications.

However, degradation of the Li-ion battery has obvious nonlinear characteristics. As a consequence, linearization using Taylor expansion leads to inaccurate approximation of state transition and observation equations resulting in non-optimal battery performance estimation. Meanwhile, AR model is a linear model which establishes a linear equation between the current state and several previous states. Although researchers have conducted some studies on the AR model improvement (Liu, Luo, Peng, Peng & Pecht, 2012), there still exists certain theoretical and application limitations in the modified AR model.

To address these problems, a data-model-fusion prognostic framework is proposed by combining the EKF algorithm with a modified nonlinear scale degradation parameter based autoregressive (NSDP-AR) model for the RUL estimation. EKF system is established on the basis of the state space model which contains state transition and observation functions. Using the strong state tracking ability of the EKF algorithm, parameters in the empirical degradation model (i.e. the state transition function) (Saha & Goebel, 2009) are obtained so that a specific model for the battery sample is established. To improve the absolute degradation parameter based ND-AR model for better long term prediction, a more reasonable scale parameter based nonlinear degradation factor is proposed which contains the degradation changing information for certain type of batteries. Correspondingly, NSDP-AR model is established by combining the proposed factor and the AR model. For a reasonable extension of the obtained NSDP-AR model in practical applications, Grey Correlation Analysis (GCA) method is used to determine the weight of the parameter groups from different battery samples which are used to establish the modified model using the true degradation information. On the basis of those parameters and corresponding weights, specific NSDP-AR model is obtained for the battery's RUL estimation. On the basis of the above work, fusion-data-model prognostic framework is established. Experimental results using the NASA PCoE and CALCE battery data sets show that the framework can predict the Li-ion battery RUL efficiently and accurately which indicates a strong practical application of the proposed framework. While, there is one thing that need to be pointed out that there are many reasons that will cause the degradation of the battery. Here, we only consider the major part of these factors, which are the battery aging caused by charge and discharge cycle. We only take the cycle aging caused degradation into consideration here.

This paper is organized as follows. In Section 2, the related prediction models including KF/EKF algorithm and current ND-AR model as well as the correlation analysis method GCA are introduced. The proposed NSDP-AR model and the corresponding fusion-data-model prognostic framework for RUL estimation of the Li-ion battery are introduced in Section 3. The effectiveness of the proposed prognostics framework is demonstrated via battery RUL prediction experiments using Li-ion battery data sets from NASA PCoE and CALCE in Section 4. Finally, the conclusion and future work are given in Section 5 and 6, respectively.

2. RELATED WORK

2.1. KF/EKF Algorithm

Many researchers have used KF/EKF algorithm to estimate the unknown parameters in the battery empirical degradation model and obtain the RUL of the battery (He, Williard, Osterman & Pecht, 2011). These researches used the state tracking ability of KF/EKF algorithm and the state space model of the battery system. EKF algorithm is the expansion of the KF algorithm in order to meet the requirements of the nonlinear applications. So, here we give more information of KF algorithm, EKF theory can be obtained by analogy of the KF algorithm.

KF is one kind of model-based algorithms that provides an efficient recursive solution of the least-squares method for the discrete-data linear systems. The state space model of the system can be described in Eq. (1).

$$\begin{cases} x_k = F_k x_{k-1} + B_k u_k + w_k \\ z_k = H_k x_k + v_k \end{cases} \tag{1}$$

Here, F_k stands for the state transition matrix, B_k is the control matrix, x_k is the k^{th} state of the system, u_k is the control input of the system, while the w_k is the system process noise which obeys the Gaussian distribution that its mean value is zero and the variance is W. The first equation named state transition function describes the relationship between the k^{th} (i.e. x_k) and the k-1$^{\text{th}}$ (i.e. x_{k-1}) state. The second equation named observation function turns the implied system state into measurable outputs. Here, the H_k is the observation matrix, z_k is the measurements and the v_k is similar with w_k but its variance is R where W and R are both real.

The prediction process using the KF algorithm is divided into two main steps: time update and measurement update which can be described by two groups of mathematical equations (Welch & Bishop, 1995). The calculation flow of KF algorithm is as Figure 1 shows.

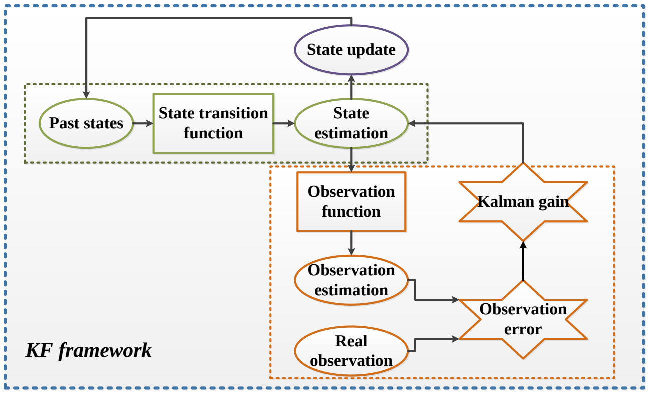


Figure 1. KF algorithm framework

Time Update:

$$\begin{cases} x_{k|k-1} = F_k x_{k-1|k-1} + B_k u_k \\ P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \end{cases} \tag{2}$$

Measurement Update:

$$\begin{cases} y_k = z_k - H_k x_{k|k-1} \\ S_k = H_k P_{k|k-1} H_k^T + R_k \\ K_k = P_{k|k-1} H_k^T S_k^{-1} \\ x_{k|k} = x_{k|k-1} + K_k y_k \\ P_{k|k} = (I - K_k H_k) P_{k|k-1} \end{cases} \tag{3}$$

Here, the meaning of F_k , B_k and u_k are same with the introduction in Eq. (1). $x_{k|k-1}$ and $P_{k|k-1}$ are the predicted mean and covariance of the state, respectively, on the time step k before seeing the measurement. $x_{k|k}$ and $P_{k|k}$ are the estimated mean and covariance of the state, respectively, on time step k after seeing the measurement. y_k is the innovation or the measurement residual on time step k. S_k is the measurement prediction covariance on the time step k. K_k is the filter gain, which tells how much the predictions should be corrected on time step k. Time update step depends on the k-1$^{\text{th}}$ state measurement update result $x_{k-1|k-1}$ and $P_{k-1|k-1}$ as well as the k^{th} control input of the system. At the same time, on the basis of the measurement residual, the filter gain can be obtained to update the previous estimation result in Eq. (2). This step means using the real measurement information to update the previous result. After each time and measurement update as shown in Eq. (2) and Eq. (3), the process is repeated to obtain the state estimate and corresponding prediction covariance. More detailed description of these equations and theory can be referred in related literature (Welch & Bishop, 1995).

However, KF algorithm is used for the systems whose features can be described into a linear equation as shown in Eq. (1). In most of the practical applications like Li-ion battery, systems always have some nonlinear characteristics such that the system model has to be described as follows, the meaning of the variables are same with Eq. (1):

$$\begin{cases} x_k = f(x_{k-1}, u_k, w_k) \\ z_k = h(x_k, v_k) \end{cases} \tag{4}$$

In order to extend KF algorithm into nonlinear conditions, EKF which uses the Taylor expansion to linearize the system functions around the current estimate is employed. After the linearization step, the same approaches as KF algorithm are used to estimate the system states. EKF is a subprime optimal estimate method which can satisfy the requirements of certain systems which have a low or median nonlinear feature.

2.2. The ND-AR Model

To address the nonlinear degradation prediction problem, an improved AR model named ND-AR model which describes the degradation features using an accelerated factor K_T as follows:

$$K_T = \frac{1}{1 + a \cdot (k+b)} \tag{5}$$

Here, a and b stand for the unknown parameters in the factor which contain the nonlinear degradation information of the capacity, and k is the prediction step which can be described the discharging cycle number the battery has experienced in another way. Here, the ND-AR model only considers the battery aging caused by charge and discharge cycle which is the major reason of the degradation. This means that ND-AR model chooses an approximated way to describe the nonlinear feature of the capacity through only taking the cycle aging into consideration but ignoring other factors. The factor extracts a correlation relationship between the prediction step k and the accelerated feature of the capacity degradation. This improvement provides a valuable reference that adopts a mathematical factor related to the degradation parameters to describe the degradation characteristics in the data-driven prediction model. The specific expression of the ND-AR model is as follows:

$$x_t = K_T \times [\phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + a_t] \tag{6}$$

Here, K_T is the accelerated factor and the remaining part of the Eq. (6) is the basic AR model. In the AR model, ϕ_k is the autoregressive coefficient and a_t is the noise which obeys Gaussian distribution. The adoption of the accelerated factor makes the AR model nonlinear. As a result, the accelerated and nonlinear degradation trend of the capacity of Li-ion battery can be retrieved (Liu, Luo, Peng & Pecht, 2012).

2.3. GCA Algorithm

GCA is an efficient and convenient correlation analysis approach which has been generally used in data variation trend similarity analysis. After choosing the standard data array Y={$Y(k)$ | k = 1,2, …,n} and the comparison data set X_i={$X_i(k)$ | k = 1,2,…,n} (i = 1,2,…,m), correlation coefficient can be calculated according to the following equation:

$$\varsigma_i(k) = \frac{\min_i \min_k |Y(k) - X_i(k)| + \rho \max_i \max_k |Y(k) - X_i(k)|}{|Y(k) - X_i(k)| + \rho \max_i \max_k |Y(k) - X_i(k)|} \tag{7}$$

where ρ is the discrimination whose value ranges from zero to infinity. According to some investigation, the effect of the resolution is best when $\rho \leq 0.5463$ (Shen, Xue & Zhang, 2003). In order to facilitate the overall comparison, a concentration which turns the correlation coefficient of each point into one final result will be made through calculating the average of the above coefficients. The correlation degree between Y and X_i is defined in Eq. (8):

$$r_i = \frac{1}{n}\sum_{k=1}^{n} \varsigma_i(k), k = 1, 2, ..., n \tag{8}$$

Here, ς_i is the correlation degree between X_i and Y and n is the number of the comparison data sets.

3. Fusion Prognostic Framework for Lithium-ion Battery RUL Estimation

3.1. Estimation of the Empirical Degradation Model

In order to establish the specific state space model for the Li-ion battery, we need to confirm the state transition function which indicates the system state transition features. Researchers such as Saha in NASA PCoE have conducted studies on establishing an empirical state model for lithium-ion batteries and proposed a degradation model to describe the capacity degradation properties (Saha & Goebel, 2009). The model can be described as follows:

$$C_{k+1} = \eta_C C_k + \beta_1 e^{(-\beta_2/\Delta t_k)} \tag{9}$$

The model makes a relevancy between the k^{th} charge cycle capacity C_k and the k+1th discharge cycle capacity C_{k+1}. Moreover, Δt_k is the rest time between cycles k and k+1, η_C is the charge and discharge efficiency named coulomb efficiency which describes the difference between the capacity filled in the battery and the capacity that the battery could provide during usage, β_1 and β_2 are the parameters that need to be determined. The first part of the model $\eta_C C_k$ describes the degradation trend of the battery capacity and the other part $\beta_1 e^{(-\beta_2/\Delta t_k)}$ represents the regenerative capacity during the rest time Δt_k. However, due to the complex operation conditions, the coulomb efficiency η_C doesn't remain constant for each cycle. So, we make η_C as another parameter that needs to be estimated.

With outstanding state tracking capability, the EKF algorithms can be used to identify the parameters in the empirical degradation model. Thus, β_1, β_2 and η_C are the states need to be estimated. Given the empirical model and the capacity degradation data sets, we can conduct the state and measurement updates as described in Eq. (10).

$$\begin{cases} \eta_{C,k} = \eta_{C,k-1} + w_{\eta_c} \\ \beta_{1,k} = \beta_{1,k-1} + w_{\beta_1} \\ \beta_{2,k} = \beta_{2,k-1} + w_{\beta_2} \\ C_{k+1} = \eta_{C,k} C_k + \beta_{1,k} e^{(-\beta_{2,k}/\Delta t_k)} + v_k \end{cases} \tag{10}$$

Here, w_{η_C}, w_{β_1}, w_{β_2} and v_k stand for the noise in the state space model which used for the parameters estimation.

3.2. NSDP-AR Model for Measurements Obtaining

According to the related researches, the ND-AR model has some limitations because the nonlinear degradation is described using an acceleration factor related to the prediction step. This is not reasonable as the parameters of the factor will change a lot with different degraded speed and trajectory, different prediction starting points, sample length of the data, sample interval and density. This makes the ND-AR model not general for other applications.

According to the degradation curve, we can draw a conclusion that the degradation trend has a close relationship with the number of the charging and discharging cycles. The degradation degree increases with the growth of the cycle number. However, the life-cycle length and prediction starting point is different according to individual application. So, we must define or extract a new parameter which can stand for the degradation period with better generalization. Current percentage of the life-cycle length (CPoL) is a scale quantity related to the charging and discharging cycles. Here, CPoL is similar with the definition of the state of health (SOH) of the battery. But the standard definition equations of SOH often use capacity or the power of the battery, so there are certain differences between the two variables. CPoL is the ratio of the current cycles the battery has experienced *k* and whole cycles during the entire lifetime of the battery *L*. The specific expression is defined as Eq. (11).

$$CPoL = \frac{k}{L} \tag{11}$$

Eq. (11) shows a better applicability for the individual battery. The CPoL can represent the current internal reaction phase which determines the capacity degradation feature.

However, *L* is a parameter just needed to be predicted, moreover, the accurate CPoL cannot be obtained in practical applications. Therefore, we need to find an approximated method to estimate it. With GCA analysis, current percentage of the predicted-life-cycle length (CPoP) shown in Eq. (12) indicates high correlation relationship with the CPoL by replacing the *L* with the predicted-life-cycle length *L'* obtained using AR model. So, we apply the CPoP as the nonlinear degradation factor, regarding CPoP as an approximated CPoL to implement the proposed approach.

$$CPoP = \frac{k}{L'} \tag{12}$$

The nonlinear degradation factor in this paper has two specific forms. The first one is the same as the related ND-AR model researches have used which can be described as Eq. (5). The other one is obtained from the capacity degradation feature perspective which can be described in an exponential related form (He, Williard, Osterman & Pecht, 2011) (Miao, Xie, Cui, Liang & Pecht, 2012). Two specific forms can be described as Eqs. (13) and (14).

$$K_T = \frac{1}{[1 + a \cdot (kp + b)]} \tag{13}$$

$$K_T = a \cdot e^{b \cdot kp} + c \cdot e^{d \cdot kp} \tag{14}$$

Here *kp* is the parameter of the CPoL and *a*, *b*, *c* and *d* are the parameters need to be estimated. With the true degradation data set of certain samples and the corresponding AR model prediction results, we apply EKF algorithm for estimating the parameters in Eqs. (13) and (14). The detailed modeling steps are similar as the ND-AR model (Liu, Luo, Peng, Peng & Pecht, 2012).

In addition, we must determine the parameters of the factor in applications when the true degradation information is unknown. We can conceive that the correlation degree of the history degradation knowledge contributes to the extension of the modified model. Therefore, the weight is determined by the correlation degree and the weighted average results can be described as Eq. (15)

$$m = \sum_{i=1}^{n} m_i \cdot \frac{r_i}{\sum_{j=1}^{n} r_j} \tag{15}$$

where *n* is the number of batteries we apply for modeling, r_i is the correlation degree between the *i*th battery sample whose parameter is m_i.

With this improved data-driven method, we can achieve the long-term degradation trend prediction of the capacity.

3.3. Fusion-data-model Prognostic Framework for Lithium-ion Battery RUL Estimation

By applying empirical degradation model as the state transition function with model-based EKF algorithm, and the prediction result with data-driven NSDP-AR model as the measurements update, we can get a novel fusion-data-model prognostic framework shown as Eq. (16).

$$\begin{cases} C_{k+1} = \eta_C \cdot C_k + \beta_1 \cdot e^{(-\beta_2/\Delta t_k)} + w_k \\ y_k = C_k + v_k \end{cases} \tag{16}$$

The whole prognostics framework is shown in Figure 2.

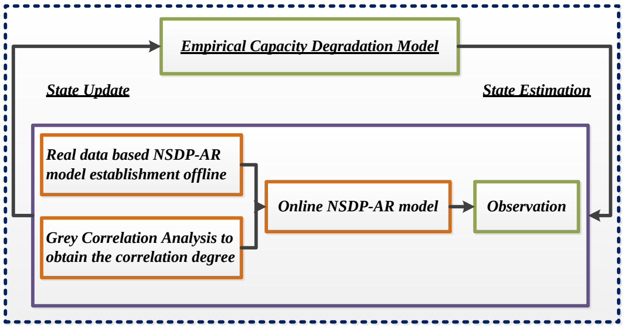

Empirical Capacity Degradation Model
State Update
State Estimation
Real data based NSDP-AR model establishment offline
Grey Correlation Analysis to obtain the correlation degree
Online NSDP-AR model
Observation

Figure 2. Fusion-data-model Framework for Lithium-ion Battery RUL Estimation

The detailed flow of fusion-data-model is as follows.

Definition: prediction starting point *T*, cycle number *k*, the length of the data for modeling *Length*, real capacity value *Capacity*, predicted capacity value *C*(*k*), failure threshold *U*, long-term trend of the capacity obtained by NSDP-AR model *NSDP-ARpredict*(*k*), and the predicted life-cycle length given by AR model *L'*.

Step 1. Choose individual batteries as the historical samples to modeling the specific NSDP-AR models.

Step 2. Pre-process the data like smoothing, outliers removing and then set parameters such as *T*, *U*, noise covariance matrix and other feature matrix. Capacity data set is divided into the historical part *Capacity*(1:T) and the testing part according to the parameter *T*.

Step 3. Predict with AR model.

Step 3-1. Determine the order of the AR model *p* according to the AIC (Akaike information criterion) principle

$$AIC(p) = N\ln\sigma_p^2 + 2p \tag{17}$$

Here *N* is the number of the data samples, *p* is the model order, σ_p^2 is the prediction variance of *p* order model.

Step 3-2. Use a fusion approach with both Burg method and Yule-Wallker method to calculate the regression coefficient $\phi_i (i = 1, 2, ..., p)$ of the *p* order AR model.

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + a_t \tag{18}$$

Choose the dynamic weight P_{1i} and P_{2i} for the respective results from each method ϕ_{1i} and $\phi_{2i} (i = 1, 2, ..., p)$, then calculate the fusion value as Eq. (19).

$$\phi_i = P_{1i}\phi_{1i} + P_{2i}\phi_{2i} (i = 1, 2, ..., p) \tag{19}$$

Step 3-3. Estimate RUL using AR(*p*) model by performing a multi-step iterative computation to obtain the long-term trend of the capacity *ARpredict*, and the corresponding life-cycle prediction result *L'*.

Step 4. Model NSDP-AR.

Step 4-1. Calculate the real value sequence of the nonlinear degradation factor $K_{T,real}$ using Eq. (20).

$$K_{T,real} = \frac{Capacity}{ARpredict} \tag{20}$$

Step 4-2. Compute the CPoL with Eq. (12) and CPoP with Eq. (11), and ensure the feasibility of the approximated method using GCA algorithm.

Step 4-3. Identify the parameters of the proposed factor with EKF algorithm.

Step 4-4. Model the NSDP-AR by combining the factor and basic AR model for historical individual battery sample.

Step 4-5. Repeat the Step 4-4 for each battery sample.

Step 4-6. Utilize GCA algorithm and weighted average method with Eq. (15) to obtain the NSDP-AR model.

Step 5. Establish the fusion prognostic model with Eq. (16).

Step 6. Predict RUL using the model obtained in Step 5.

There is a key problem need to be explained, that is in this research we only know the data obtained from the system, when we conduct the experiments, the state estimation result will be put into next state estimation step. That is, the prediction is a multi-step iterative prediction, we use the last data point of the collective data set and then all the steps are completed with the help of the previous estimation results.

4. EXPERIMENTS AND ANALYSIS

4.1. Battery Data Set

We utilize two battery data sets from NASA PCoE and CALCE for evaluate the performance of the proposed framework, respectively. The first Li-ion battery data set including battery #5, #6 and #18 are from NASA PCoE. These batteries were tested under certain condition (with the temperature +25℃). The 2 Ah batteries were charged with the charging current 1.5A until the batteries voltage reaches 4.2V, then discharged with the discharging current 2A until the batteries voltage reached 2.5V (Saha, Goebel, Poll & Christophersen, 2009) (Saha & Goebel, 2007). When the battery capacity reaches about 70% of rated capacity, the Li-ion battery is regarded to reaching its end of life (EOL). In the experiments, the threshold is set as 1.38Ah.

The second Li-ion battery data sets are from the CALCE of the University of Maryland containing battery #8, #21 and #33. The cycling of the batteries was implemented with the Arbin BT2000 battery testing system under room temperature. The 1.1 Ah rated capacity of batteries were adopted in the test with the discharging current (0.45A that the discharging speeds is 0.5C) (He, Williard, Osterman & Pecht, 2011) (He, Williard, Osterman & Pecht, 2011) (He, Williard, Osterman & Pecht, 2011). The threshold is set as 0.88 Ah.

4.2. Methods for Comparison

In order to verify the effectiveness of the NSDP-AR model and the fusion prognostic framework, another two algorithms are applied for comparison:

Method 1. Model-based EKF framework for RUL prediction.

Method 2. Fusion-data-model prognostic framework with EKF algorithm and ND-AR model.

These three methods will be utilized to predict the lithium-ion battery RUL.

4.3. Evaluation Criteria

The performance of the algorithm will be evaluated by RUL and capacity prediction errors.

RUL prediction error: We use the RUL prediction result $RUL_{prediction}$ to minus the true RUL value RUL_{real} to describe the RUL prediction error E_{RUL}.

$$E_{RUL} = RUL_{prediction} - RUL_{real} \tag{21}$$

Mean Absolute Error (MAE) for capacity prediction: The MAE is described in Eq. (22) to evaluate the prediction error.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|x(i) - \overline{x}(i)\right| \tag{22}$$

Here, $x(i)$ is the true capacity value and $\overline{x}(i)$ is the prediction value for each cycle, n is the length of data.

Root Mean Square Error (RMSE) for capacity prediction: The RMSE can be described as Eq. (23):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[x(i) - \overline{x}(i)\right]^2} \tag{23}$$

The symbols above have the same meaning as MAE.

4.4. Parameters Identification of the Empirical Degradation Model

With high performance of state tracking for EKF algorithm, we can determine the fit parameters of the empirical degradation model. We have conducted a series of experiments with several data sets to estimate the reasonable model for each battery. In order to describe the tracking result briefly, we choose NASA Battery #18 and CALCE Battery #33 to show the result.

The parameters in the experiments for NASA Battery data are set as follows: the length of the history data set L_1 = 50; the failure threshold U = 1.38Ah; initial state vector [a_0; b_0; c_0] = [1; 10; 10] where a_0 stands for the coulombic efficiency , b_0 and c_0 stand for the unknown parameters β_1 and β_2; the process noise variance matrix Q = [0.0001, 0,0; 0, 0.0001, 0; 0, 0, 0.0001]; observation noise variance R = 0.0001. The weighted average numerical parameters results are: a = 0.9958; b = 10.0040; c = 9.9602.

The parameters in the experiments for CALCE Battery data are set as follows: the length of the history data set L_1 = 280; the failure threshold U = 0.88 Ah; initial state vector [a_0; b_0; c_0] = [1; 10; 10] where a_0, b_0 and c_0 have the same meaning as NASA experiments; the process noise variance Q = [0.0001, 0, 0; 0, 0.0001, 0; 0, 0, 0.0001]; observation noise variance R = 0.0001. The weighted average numerical parameters results are: a = 0.9991; b = 10.0015; c = 9.9847.

4.5. Parameters identification of NSDP-AR Model

According to the modeling flow of the NSDP-AR model, we can obtain the specific nonlinear degradation factor for each battery data set as well as the weighted average result on the basis of the correlation degree of the degradation trend. Both of the factor forms proposed before have been adopted into the experiments respectively for a comparison between different factor forms.

In the experiments for NASA battery data set, Battery #5 and #6 are selected to establish the NSDP-AR models. The obtained weighted parameters will be adopted into the RUL estimation for Battery #18. Similarly, in the experiments for CALCE data set, Battery #8 and #21 are used for modeling and Battery #33 is used for verification. We set the prediction starting point T in the medium-term of each battery sample. Correlation degree between the modeling batteries and the estimated battery are calculated with GCA method where the distinguish coefficient is set as 0.5463. The results are shown in Table 1.

Table 1. Weighted parameters for prediction application

Index	GCA	Form	Parameters			
			a	*b*	*c*	*d*
NASA #5	0.6117	1	-0.0303	0.3938	-	-
NASA #6	0.7612	2	0.9896	0.0028	0.0005	3.9656
CALCE #8	0.6985	1	-0.1326	-0.5346	-	-
CALCE #21	0.6776	2	1.0357	-0.0268	0.0011	3.3169

4.6. RUL Estimation with the Fusion-data-model Prognostic Framework

With the parameters obtained in Section 4.4 and 4.5, we conducted RUL estimation experiments using different methods. With the fusion prognostics, the RUL prediction results of NASA Battery #18 are shown in Figures 3 and 4. The result in Figure 3 is based on the nonlinear degradation factor form as Eq. (13), while Figure 4 shows the result based on the factor as Eq. (14).

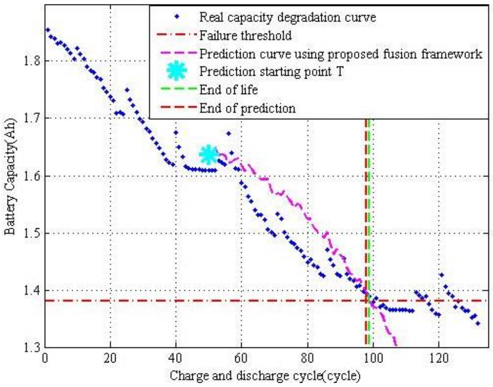


Figure 3. RUL prediction based on proposed fusion framework for NASA Battery #18 (Form 1)

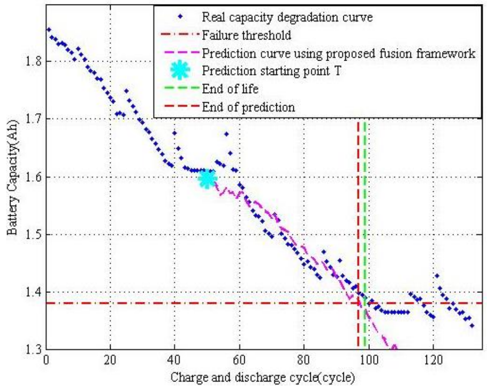


Figure 4. RUL prediction based on proposed fusion framework for NASA Battery #18 (Form 2)

Similarly, Figures 5 and 6 show the experimental results for CALCE battery data sets. Here, two different factor forms are also adopted in the fusion prognostic framework, which are similar as the experiments above.

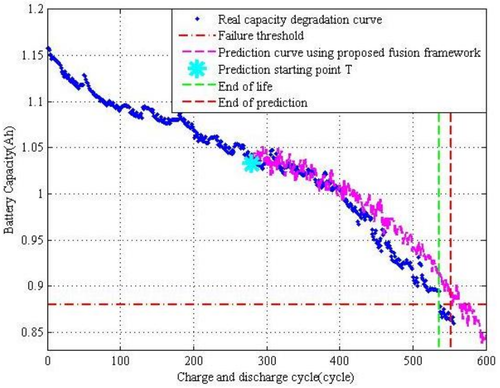


Figure 5. RUL prediction based on proposed fusion framework for CALCE Battery #33 (Form 1)

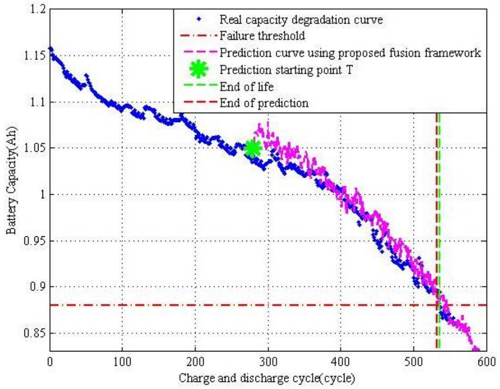


Figure 6. RUL prediction based on proposed fusion framework for CALCE Battery #33 (Form 2)

From the RUL prediction results for two types of battery data sets, we can find that the degradation curves as well as the estimated EOL points are very close to the actual values. It is indicated that the proposed method is efficient and accurate.

4.7. Comparison and Analysis

To compare and evaluate different RUL prediction methods, we also conducted more experiments. The quantitative results are shown in Table 2 and Table 3 for NASA and CALCE batteries, respectively.

By analyzing the experimental results above, it can be seen that the proposed method can achieve better performance than the other methods with the MAE and RMSE shown in Tables 2 and 3. The ND-AR model based on prediction step *k* may sometimes make a deterioration of the prediction results, because this method does not take into account the diversity of degradation speed. In contrast, the NSDP-AR model can overcome this problem and lead to a better RUL estimated result. It is showed that the NSDP-AR model based fusion framework achieve an improvement on prediction accuracy.

Table 2. Comparison of RUL estimation with different prognostics for NASA Battery #18

Methods	E_{RUL}	MAE	RMSE
EKF	9	0.0804	0.1112
EKF & ND-AR (1)	1	0.0862	0.1335
EKF & ND-AR (2)	5	0.0458	0.0636
EKF & NSDP-AR (1)	1	0.0938	0.1338
EKF & NSDP-AR (2)	2	0.0726	0.1083

Table 3. Comparison of RUL estimation with different prognostics for CALCE Battery #33

Methods	E_{RUL}	MAE	RMSE
EKF	0	0.0206	0.0239
EKF & ND-AR (1)	48	0.0287	0.0357
EKF & ND-AR (2)	×	116.5041	257.2003
EKF & NSDP-AR (1)	16	0.0173	0.0213
EKF & NSDP-AR (2)	4	0.0133	0.0162

Note that, the RUL estimated error for CALCE Battery #33 obtained with EKF algorithm equals to 0 in Table 3. However, the estimated degradation trend is obviously diverse from the real curve, and the satisfied RUL prediction result is a coincidence for certain battery samples.

At the same time, the real-time performance of EKF algorithm is superior to the other statistical filtering such as Particle Filter. The comparison between the operating speed of PF and EKF is shown in Table 4. We can find that the execution time of fusion EKF algorithm is shorter than the fusion PF method but the accuracy is close with each other.

Table 4. Comparison of RUL estimation with different prognostics for CALCE Battery #33

Methods	E_{RUL}(cycle)	Time(s)
Fusion EKF	7	2.27
Fusion PF	9	11.45

Moreover, the NSDP-AR model realizes satisfied prediction for the nonlinear degradation trend, and more reasonable than ND-AR model, which stands that the proposed fusion framework shows a good application prospect.

5. Conclusions

This paper explores an improved fusion-data-model prognostic framework with EKF algorithm and NSDP-AR model for battery RUL estimation. The main contributions of this research can be concluded as follows. (1) A data-model fusion prognostic framework with low computation complexity and better real-time capability is proposed. (2) Improvements are obtained to modify the ND-AR model including a scale parameter based nonlinear degradation analysis, and an approximate method to obtain the CPoL data sets, and an extension method using GCA method, as a result, an improved NSDP-AR model is achieved. (3) A combination of model-based EKF algorithm and improved data-driven NSDP-AR model which weakens the dependence on empirical degradation model and improves the nonlinear predicting accuracy.

6. Future Work

In future, more efforts should be focused on obtaining the RUL estimation result when the capacity cannot be observed on-line, which is the actual condition in practical applications. Up to now, we have conducted some studies with a novel parameter called time intervals to equal discharging voltage difference (TIEDVD) to realize indirect RUL prognostics. Meanwhile, in this paper we only conduct the study on estimating the RUL when the battery is fully discharged and discharged, which is not true in practical applications. On the other hand, the empirical degradation model used in this framework only takes the capacity degradation principle and the regeneration phenomenon of the capacity during test time into consideration. More factors such as depth of discharge (DOD) and internal temperature of the lithium-ion battery should be considered while modeling. Especially, the indirect health indicator (HI) with on-line monitoring parameters and the fusion HI combines with more correlation analysis methods should be applied to realize more flexible and applicable RUL estimation.

Acknowledgement

This work is partly supported by National Natural Science Foundation of China under Grant No. 61301205, Twelfth Government Advanced Research Fund under Grant No. 51317040302, Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20112302120027, Fundamental Research Funds for the Central Universities under Grant No. HIT.NSRIF.2014017, and China Scholarship Council. The author would like to thank Dr. Zhimin Xi at University of Michigan – Dearborn for his revision and suggestion to this manuscript.

References

He, W., Williard, N., Osterman, M., & Pecht, M. (2011). Prognostics of Lithium-ion Batteries using Extended Kalman Filtering. *IMAPS Advanced Technology Workshop on High Reliability Microelectronics for*

Military Applications. Linthicum Heights, MD, USA, 1- 4.

He, W., Williard, N., Osterman, M., & Pecht, M. (2011). Remaining useful performance analysis of batteries. *2011 IEEE Conference on Prognostics and Health Management* (*PHM*) *Conference*, Denver, CO, USA, June 20-23.

He, W., Williard, N., Osterman, M., & Pecht, M. (2011). Prognostics of lithium-ion batteries based on Dempster-Shafer theory and the Bayesian Monte Carlo method. *Journal of Power Sources*, vol. 196, pp. 10314-10321.

Liu, D., Luo, Y., Peng, Y., Peng, X., & Pecht, M. (2012). Lithium-ion Battery Remaining Useful Life Estimation Based on Nonlinear AR Model Combined with Degradation Feature. *Annual Conference of the Prognostics and Health Management Society 2012*, Minneapolis, Minnesota, USA, September, 24-27.

Liu, J., Saxena, A., Goebel, K., Saha, B., & Wang, W. (2010). An Adaptive Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-ion Batteries. *Annual Conference of the Prognostics and Health Management Society 2010*. Portland, Oregon, USA, October 10-16.

Liu, J., Wang, W., Ma, F., Yang, Y. B., & Yang, C. S. (2012). A data-model-fusion prognostic framework for dynamic system state forecasting. *Engineering Applications of Artificial Intelligence*, vol. 25, pp. 814-823.

Miao, Q., Xie, L., Cui, H., Liang, W., & Pecht, M. (2012). Remaining useful life prediction of lithium-ion battery with unscented particle filter technique. *Microelectronics Reliability*: vol. 53, pp. 805-810.

Saha, B., & Goebel, K. (2007). "Battery Data Set", NASA Ames Prognostics Data Repository, [http://ti.arc.nasa.gov/project/prognostic-data-repository], NASA Ames, Moffett Field, CA.

Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009), Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework. *IEEE Transactions on Instrumentation and Measurement*, vol. 58, pp. 291-297.

Saha, B., & Goebel, K. (2009). Modeling li-ion battery capacity depletion in a particle filtering framework. *Annual Conference of the Prognostics and Health Management Society 2009*, San Diego, CA, USA, September 27 – October 1.

Shen, M., Xue, X., & Zhang, X. (2003). Determination of Discrimination Coefficient in Grey Incidence Analysis. *Journal of Air Force Engineering University* (*Natural Science Edition*), vol. 4, pp. 68-70.

Wei, W. W. S. (1994). *Time series analysis*. Redwood City, California: Addison-Wesley.

Welch, G., & Bishop, G. (1995). *An introduction to the Kalman filter*.

Zhang, J., & Lee, J. (2011). A review on prognostics and health monitoring of Li-ion battery. *Journal of Power Sources*, vol. 196, pp. 6007-6014.

BIOGRAPHIES

Datong Liu received the B.Sc. and M.Sc. degrees in Department of Automatic Test and Control from Harbin Institute of Technology (HIT), Harbin, China in 2003 and 2005, respectively. During 2001 to 2003, he also minored in the Computer Science and Technology in HIT. He received the Ph.D. degree in major of measurement and instrumentation from HIT in 2010. He is now an assistant professor in Department of Automatic Test and Control, HIT. His research interests include automatic test and intelligent information processing, time series analysis, Data-driven PHM, Machine Learning, Data Mining, etc. He is currently an IEEE member, ACM member, PHM society member, China Computer Federation member. He has published more than 30 journal and conference papers, and holds 15 invention patents and more than 30 invention patents pending in China. He is now in charge of 8 projects related to PHM that supported by National Natural Science Foundation of China, Research Fund for the Doctoral Program of Higher Education of China, Twelfth government advanced research fund in China, Fundamental Research Funds for the Central Universities, etc.

Limeng Guo received her B.Sc. degree in Department of Automatic Test and Control from Harbin Institute of Technology (HIT), Harbin, China in 2013. Now she is a postgraduate in Major of Instrumentation Science and Technology of HIT. Her research interests include data-driven PHM, fusion prognostic approach, Battery Management System, and so on. She wins the best paper award of ICEMI 2013, and best graduated thesis award of HIT in 2013. She is now involved in 5 projects focusing on system health management of complex system, and also holds 10 invention patents pending in China.

Remaining Useful Life Prognostics for Lithium-ion Battery Based on Gaussian Processing Regression Combined with the Empirical Model

Shan Yin, Jingyue Pang, Datong Liu, Yu Peng

Department of Automatic Test and Control, Harbin Institute of Technology, Harbin, 150080, China

yy.yinshan@163.com
pjy19880909@163.com
liudatong@hit.edu.cn
pengyu@hit.edu.cn

ABSTRACT

Data-driven techniques based on Bayesian framework like Gaussian Process Regression (GPR) can not only predict the lithium-ion battery Remaining Useful Life (RUL), but also provide the uncertainty representation. However, it is always difficult to choose the covariance function of GPR and the confidence bound is usually large if the training data are not enough. In order to solve this problem, a combining method is proposed, it is a prognostic framework based on GPR model combined with Empirical Model (EMGPR) to realize the lithium-ion battery RUL prediction. EMGPR has the advantages of predicting the tendency and uncertainty management for RUL estimation. The modeling process of EMGPR consists of two steps. The self-deterministic part, which reflects the real physical process of battery degradation, is approximated by the empirical model. And the disturbance part, which is caused by random noise such as measurement and environment noise, is expressed by the GPR model. In application, two key factors of EMGPR are focused. Firstly, the prediction result is not accurate enough if the training data are not very reliable. In this case, more reliable training data should be selected optimized. Secondly, the characteristic of the disturbance is involved to determine the kernel function of GPR model. With this EMGPR framework, the RUL result is estimated with uncertainty representation, as well, the covariance function of GPR is easy to choose. Experiments with NASA PCoE and CALCE battery data show the satisfactory result can be obtained with the EMGPR approach.

1. INTRODUCTION

Lithium-ion batteries have been widely used in the domains of portable designs, notebook computers, electric vehicles and airplanes and spacecrafts because of their high energy density, low self-discharge rates, wide operating temperature ranges, and high charge-discharge rates. Lithium-ion batteries which act as energy storage components are critical to the safety of electric devices or systems (Yang, Ye, Guo & Ma, 2012). However, the lithium-ion battery will degrade over time on account of aging, environmental impacts and dynamic loading, etc. Failure of battery may lead to loss of operation, decreased output, and it might even bring danger to the operators. Hence, it is meaningful to detect the underlying degradation and take measures to prevent the potential failures and ultimately prevent the disastrous failures. Prognostics and health management (PHM), is to predict how soon a system or component will loss efficacy or reach the failure threshold (Zhang & Lee, 2011) (Widodo, Shim, Caesarendra & Yang, 2011). Effective precaution measures could be taken in advance if we predict the failures successfully. For health state monitoring, battery parameters included voltage, current, temperature and capacity are measured to estimate the state of charge (SOC), the State of Health (SOH), the end of life (EoL) and the remaining useful life (RUL) (here we only focus on the remaining cycle life) of lithium-ion battery (Saha, Goebel & Christophersen, 2009).

In order to estimate lithium-ion battery RUL well and make an optimized design of battery-systems, both model-based and data-driven techniques are applied. Gao *et al.* presents a dynamic model which is suitable for virtual-prototyping of portable battery-powered systems. The model takes nonlinear equilibrium potentials, rate, temperature-variation, thermal effects and transient power demand into consideration (Gao & Liu, 2002). Rong *et al.* introduces an analytical model to predict the remaining capacity of a lithium-ion battery, which is in view of the cycle-aging and temperature effects (Rong, 2006). Erdinc *et al.* proposes a dynamic model which cares about the significant temperature and capacity fading (Erdinc, Vural &

Shan Yin et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Uzunoglu, 2009). The model-based technologies are direct and intelligible. However, the performances and characters of batteries are highly influenced by their complex operating conditions and internal inaccessible. Hence, it is difficult to establish an exact model to describe whole information of lithium-ion batteries.

With the advancement of sensor and data storage technologies, the data-driven prognostics are emphasized. The data-driven approach Auto-Regressive Integrated Moving Average model (ARIMA) is utilized for RUL estimation. Without taking any physical process into consideration, it is possible to deal with the non-stationary monitoring data. But the ARIMA model is unsuitable for long-term prediction. Then the Extended Kalman Filter (EKF) is proposed to handle the nonlinear and non-stationary modeling. The EKF cannot accommodate the non-modeled process (Saha, Goebel & Christophersen, 2009) (Do, Forgez, Benkaha & Friedrich, 2009) (He, Williard, Osterman & Pecht, 2011). Lots of data-driven methods cannot manage the uncertainty of prognostics.

Hence, the uncertainty caused by the measurement errors, the environmental noise and the model noise is addressed in prognostics. Especially, for long-term state prediction, the uncertainty must be represented and managed effectively (He, Williard, Osterman & Pecht, 2011). Bayesian framework based data-driven approaches such as Particle Filter (PF) (Dalal, Ma & He, 2011) (Xing, Tsui & Pecht, 2012), Relevance Vector Machine (RVM) and Gaussian Process Regression (GPR) can provide the uncertainty reprepsentation of the RUL value (Goebel, Saha & Saxena, 2009) (Saha, Goebel, Poll & Christophersen, 2009) (Goebel, Saha, Saxena, Celaya & Christophersen, 2009) (Chen & Pecht, 2012).

Among these algorithms, the GPR model based on the Bayesian framework is flexible to be adopted in the non-linear regression of stochastic time series. And it can predict the RUL of lithium-ion batteries, offerring the confidence interval of predicted value and manage the uncertainty (Rasmussen & Williams, 2006) (Saha, Goebel & Christophersen, 2009). The GPR algorithm provides variance around its means predictions, and combines prior knowledge with observed data (Li & Zhang, 2010) (Cristianini & Taylor, 2000) (Scholkopf & Smola, 2002).

Although the GPR provides a theoretically framework for prognostics, there are some limitations in practical usage. First, choosing proper kernel function (covariance function) is critical. However, it is sometimes difficult to make an optimal selection due to lack of knowledge about the actual process. In addition, the GPR model can predict the mean function and variance function with hyper-parameters, if the training data is not available enough, the prediction confidence bound will become so large and the result is not reliable (Goebel, Saha & Saxena, 2009).

In this work, a novel empirical mode combined with GPR is proposed to predict the RUL of lithium-ion battery (it is as EMGPR). Firstly, by the combination, we can conquer the inconvenience of choosing the prediction covariance function. Secondly, the optimization of the selection of the training data is focused to reduce the prediction uncertainty.

In the EMGPR framework, training data are divided into two parts. One is self-deterministic and could be estimated by empirical model. This part reflects the real physical deterioration of lithium-ion battery. The other is the disturbance components which reflects the random noise including measure noise, model noise and environment noise could be expressed by GPR. The final prediction result of RUL is the sum of the two parts. Experiments have been done with data set of NASA and the University of Maryland to illustrate the effectiveness of EMGPR prognostics framework for lithium-ion battery.

This paper is organized as follows. The GPR method is depicted in Section 2. In Section 3, the lithium-ion prognostic method of EMGPR is introduced. Experiments of lithium-ion battery RUL estimation with EMGPR are discussed in details in Section 4. The conclusion and future work are described in Section 5 and 6.

2. Gaussian Processing Regression Model

The GPR model affects input variables to output crack growth by probabilistically inferring the nonlinear relationship between input and output (Mohanty, Das, Chattopadhyay & Peralta, 2009). It has been widely applied in machine learning (Rasmussen & Williams, 2006) (Snelson, 2007), data mining, image processing, pattern recognition and prognostics of both metallic and electronics systems. Particularly, GPR model is utilized for the prognostics of lithium-ion battery (Liu, Pang, Zhou, & Peng, 2010).

The basic idea of GPR modeling is to define the Gaussian Processing (GP) to describe a function distribution. The GP is a collection of random and finite stochastic variables which follows to Gaussian distribution. GP is fully described by its mean function $m(x)$ and the covariance function $k(x,x')$.

$$f(x) \sim GP(m(x),k(x,x')) \tag{1}$$

$$m(x) = E[f(x)] \tag{2}$$

$$k(x,x') = E[(f(x)-m(x)),(f(x')-m(x'))] \tag{3}$$

where the symbol E means the expectation.

For the regression, we model as,

$$y = f(x)+\varepsilon \tag{4}$$

Here x is the input vector, f is the function output and y is the observed values with noise. The noise ε is usually assumed to follow the Normal distribution.

$$\varepsilon \sim N(0, \sigma_n^2) \tag{5}$$

The prior distribution of y is,

$$y \sim N(m(x), K(X,X) + \sigma_n^2 I_n) \tag{6}$$

The prior joint distribution of y and the prediction value f_* is described as follows:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim N\left(m(x), \begin{bmatrix} K(X,X)+\sigma_n^2 I_n & K(X,x_*) \\ K(x_*,X) & k(x_*,x_*) \end{bmatrix} \right) \tag{7}$$

The parameter $K(X,X) = K_n = (k_{ij})$ is a symmetric positive definite covariance matrix. The element in the matrix k_{ij} means the correlation of x_i and x_j. The equation $K(X,x_*) = K(x_*,X)^T$ is the covariance matric of test data x_* and training data X. $k(x_*,x_*)$ is the covariance of x_* itself. The symbol I_n is a unity matrix.

We can compute the posterior distribution of prediction value f_*:

$$f_* \mid X, y, x_* \sim N(\overline{f}_*, \mathrm{cov}(f_*)) \tag{8}$$

$$\overline{f}_* = m + K(x_*,X)[K(X,X) + \sigma_n^2 I_n]^{-1}(y - m) \tag{9}$$

$$\mathrm{cov}(f_*) = k(x_*,x_*) - K(x_*,X) \times [K(X,X) + \sigma_n^2 I_n]^{-1} K(X,x_*) \tag{10}$$

Hence, the mean of the prediction output is,

$$\hat{\mu}_* = \overline{f}_* \tag{11}$$

And the variance of the prediction output is,

$$\hat{\sigma}_*^2 = \mathrm{cov}(f_*) \tag{12}$$

Different mean functions and covariance functions contain some unknown parameters, they are hyper-parameters. Based on marginal likelihood Bayesian theory, we can identify the optimal hyper-parameters with a numerical optimization routine such as conjugate gradients (Rasmussen & Williams, 2006) (Li & Zhang, 2010).

By analyzing Eq. (9) to Eq. (12), the main challenge is to determine the covariance function for the prognostics with GPR.

3. Prognostics Framework of EMGPR

The remaining useful capacity of lithium-ion battery is predicted in this paper to calculate the RUL. A fused framework of EMGPR is proposed to predict the RUL of lithium-ion battery. Here two important steps are involved. Firstly, the characteristic of battery is analyzed to set the kernel function of GPR. Secondly, experiments are implemented to optimize the preferable training data.

Theoretically, any time series can be represented as consisting of two parts, a self-deterministic part and a disturbance component (Saha, Goebel & Christophersen, 2009). The self-deterministic part depends on the real physical process, while the disturbance component mainly influenced by the random noise containing measurement noise, process noise, surrounding environment noise, etc. In the EMGPR framework, the self-deterministic part is described by the empirical model such as double-exponential model and Gaussian model, which influent the output by curve fitting algorithms. The disturbance component is expressed by GPR Model. The final prediction result is the fusion of the two parts. The prognostics flowchart is shown in Figure 1.

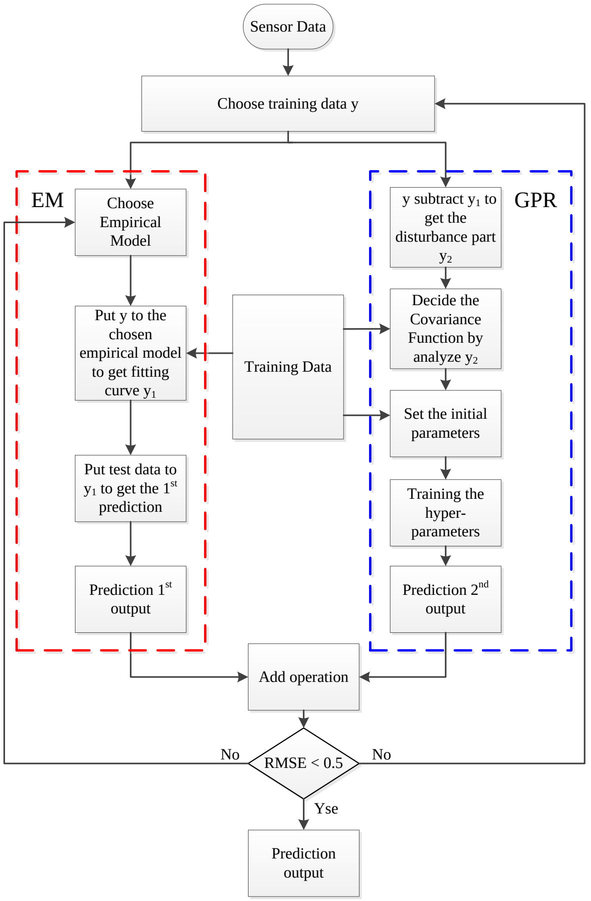

Sensor Data
Choose training data y
EM
Choose Empirical Model
Put y to the chosen empirical model to get fitting curve y_1
Put test data to y_1 to get the 1st prediction
Prediction 1st output
Training Data
GPR
y subtract y_1 to get the disturbance part y_2
Decide the Covariance Function by analyze y_2
Set the initial parameters
Training the hyper-parameters
Prediction 2nd output
Add operation
No
RMSE < 0.5
No
Yse
Prediction output

Figure 1. The fusion EMGPR framework based on combined GPR and Empirical Model

The detail steps of the EMGPR algorithm are as follows:

Step 1. Choose the training data. This step is executed repeatedly until a satisfied training accuracy is obtained. In this work, we assume that the training ends while the Root Mean Square Error (RMSE) value is less than 0.5.

Step 2. Choose a proper empirical model to describe the real physical deterioration of lithium-ion battery. Similarly, this step is carried out repeatedly to identify the suitable empirical model. The criterion of choosing the training data and empirical model is an experienced setting with experiments.

Step 3. Put the training data into the identified empirical model to get a fitting curve y_1 (In this step, the parameters of empirical model are determined).

Step 4. Predict with the model in Step 3 to obtain the 1st prediction output.

Step 5. Subtract y_1 from y to get the disturbance part, denoted as the variable y_2.

Step 6. Analyze the characteristics of disturbance part and choose the covariance function of GPR model.

Step 7. Initialize parameters of mean function and covariance function of GPR model.

Step 8. Train the hyper-parameters of covariance function.

Step 9. Compute the prediction results of disturbance with GPR model (it is as the 2nd prediction output).

Step 10. Fuse the 1st prediction output and the 2nd prediction output together to obtain the final estimated value.

4. Experiments and Discussion

4.1. Raw Data of Lithium-ion Batteries

The data set used in this work to perform the lithium-ion battery prognostics are obtained from the data repository of NASA Ames Prognostics Center of Excellence (PCoE) and the Center for Advanced Life Cycle Engineering (CALCE) of the University of Maryland.

The battery data from NASA were run through 3 different operation profiles (charge, discharge and impedance) at room temperature. Charging was carried out in a constant current mode at 1.5A until the battery voltage reached 4.2V and then continued charging in a constant voltage mode until the charge current dropped to 20mA. Discharging was performed at a constant current level of 2A until the battery voltage falling to 2.7V, 2.5V, 2.2V and 2.5V for batteries B0005, B0006, B0007 and B0018 respectively. Impedance measurement was carried out through an electrochemical impedance spectroscopy frequency sweep from 0.1Hz to 5kHz. Repeated charge and discharge cycles result in accelerated aging of the batteries while impedance measurements provide insight into the internal battery parameters that change with aging processes. The experiments were stopped when the batteries reached end-of-life criteria, which was a 30% fade in rated capacity (from 2Ahr to 1.4Ahr). This data set offers us the discharge capacity of each cycle.

Figure 2 shows the capacity degradation of battery from NASA, assuming that the capacity threshold is 1.41Ah. The horizontal axis represents the number of charge and discharge cycles. The vertical axis represents the capacity (Ah).

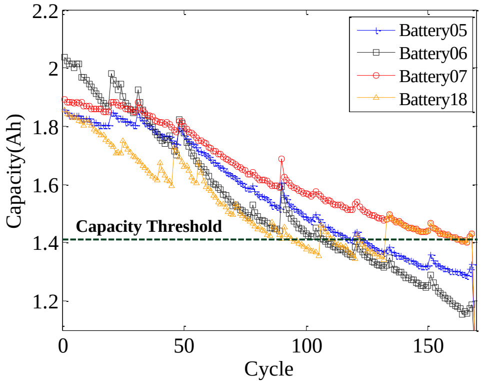


Figure 2. Capacity Degradation of Battery from NASA PCoE

Another data set is obtained from CALCE of the Maryland University, which is tested on the BT2000 lithium-ion battery experimental system. The experiment data contain two groups. The rated capacity is 1.35Ah and 1.1Ah separately. The experiments were done at 20℃ to 25℃, and the time, charging current/voltage, discharging current/voltage and charging/discharging capacity values are offered. Charging was carried out in a constant current mode at 0.675A until the battery voltage reached 4.2V and discharge was carried out at a constant current level of 1.35A until the battery voltage felt to 2.7V. The discharging rate of battery CS2_8, CS2_21, CS2_33 and CS2_34 are 0.5C. The experiments were stopped when the batteries reached end-of-life criteria, which was a 20% fade in rated capacity (from 1.1Ahr to 0.88Ahr).

Figure 3 shows the capacity degradation of battery from CALCE of. Here the capacity threshold we set is 0.88Ah.

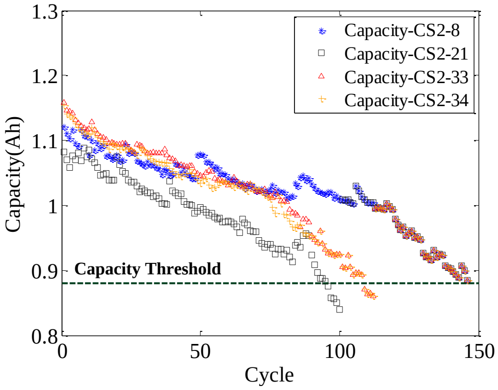


Figure 3. Capacity degradation of battery from CALCE

4.2. RUL prediction with different size of training data

The prediction result with large confidence bound using the non-sufficient available data is analyzed here. Experiments are implemented NASA battery B0007 to determine the better size of the training data. The training data (cycle) are from cycle 2 to 140, from cycle 50 to 140 and from cycle 100 to 140, respectively. The predicted results are shown in Figure 4. The red line of circle is the real test data, and the blue line with plus sign is the prediction result with training data from cycle 20 to cycle 140, the grey line with triangle is the prediction result with training data from cycle 50 to cycle 140 and the green line with square is the prediction result with training data from cycle 100 to cycle 140. The grey shade represents the prediction confidence bound.

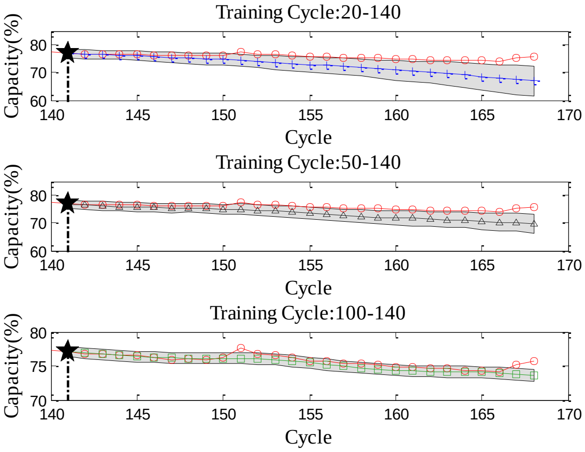


Figure 4. Prediction with different size of training data

Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are utilized to evaluate the accuracy of the estimation, which are defined as Eq. (13) and Eq. (14).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[x(i) - \bar{x}(i)]^{2}} \tag{13}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|x(i) - \bar{x}(i)\right| \tag{14}$$

Here $x(i)$ and $\bar{x}(i)$ represent the actual test data and the predicted result.

The quantified comparison is shown in Table 1.

Table 1 RMSE and MAE with different test data

Training Cycle	RMSE Value	MAE Value
20-140	0.3873	0.0581
50-140	0.2719	0.0430
100-140	0.0676	0.0090

From Table 1, we can find that better prediction result can be obtained and the prediction confidence bound is smaller with local training data set, although the local data set is with less data points.

4.3. RUL prediction with different covariance functions

In this section, experiments are executed to illustrate the influence of covariance function in the lithium-ion battery prognostics based on GPR. Predictions are implemented with two common covariance functions, conSEiso function and covMaterniso function. Based on the experiments above, training data are chosen from 100 to 140 cycles. Prediction result is shown in Figure 5.

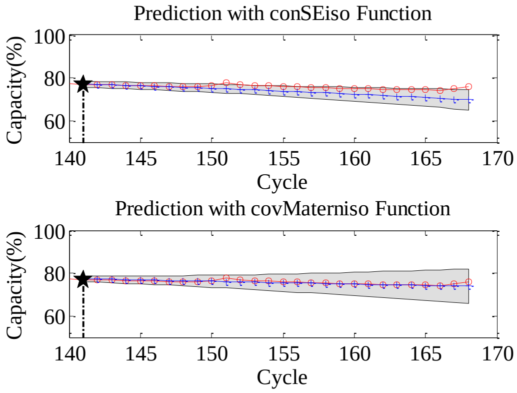


Figure 5. Predicted results with different covariance functions

In detail, the conSEiso function and the covMaterniso function are defined as Eqs. (15) and (16). In Eqs. (15) and (16), θ_1、θ_2、θ_3、ω、l、d are hyper-parameters. The valuables t and t' are the real value and prediction value.

$$k(t,t') = \theta_1^2 \exp(-\frac{(t-t')^2}{2\theta_2^2}) \tag{15}$$

$$k(t,t') = \theta_3 \exp(-\frac{1}{2}\sum_{l=1}^{d}\omega(t-t')^2) \tag{16}$$

The RMSE and MAE are shown in Table 2.

Table 2. Predicted RMSE and MAE with different covariance functions

Covariance function	RMSE	MAE
conSEiso	0.2520	0.0416
conMaterniso	0.0628	0.0083

We can find that the covariance function plays an important role in the prognostics of lithium-ion based on GPR model. Thus, it is necessary to take more efforts to the choice of covariance function.

4.4. Choice of covariance function

We assume that the battery data is composed by two parts: one reflects the inherent degradation regular (self-deterministic part) which can be described and analyzed as double-exponential or Gaussian empirical model. The other part is the disturbance component, which connects with environment factor, operating load etc., which are stochastic and can be estimated with GPR model.

Usually, the lifetime of electronics component degenerates with double-exponential or Gaussian curve. Hence, we firstly use a double-exponential model, described as Eq. (17), or Gaussian model, described as Eq. (18), to approximate the self-deterministic part.

$$y_1 = a_1 e^{b_1 x} + c_1 e^{d_1 x} \tag{17}$$

$$y_1 = a_2 e^{-(\frac{x-b_2}{c_2})^2} \tag{18}$$

Here, parameters a_1, b_1, c_1, d_1, a_2, b_2 and c_2 can be identified by fitting with the training data. The criterion of the choice of the empirical model is whether the prediction RMSE value is less than 0.5.

Then, we can construct the disturbance component (y_2) which can be predicted with GPR,

$$y_2 = y - y_1 \tag{19}$$

where y is the raw data value.

The disturbance parts are indicated in Figure 6.

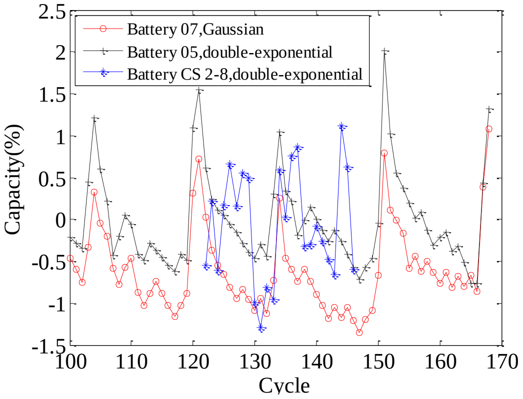


Figure 6. Disturbance parts

For battery B0007, we choose training data from cycle 100 to 140, the prediction output cycle is from 141 to 168 and the empirical model is Gaussian model. For battery B0005, the training data is from cycle 100 to 140, the prediction output cycle is from 141 to 168 and the empirical model is double-exponential model. For battery CS2_8, the training data is from cycle 122 to cycle 130, the prediction output cycle is from 131 to 146 and the empirical model is double-exponential model.

We can conclude that the training number is some extent periodical from Figure 6. Therefore, we use the covariance function as Eq. (20),

$$k_f = \sigma_f^2 \exp(-\frac{2}{l_2^2}\sin^2(\frac{\omega}{2\pi}(x-x'))) \tag{20}$$

With EMGPR framework, the regular of deterministic is represented, and the difficulty of choosing the covariance function of GPR model is overcome.

4.5. RUL Prediction with EMGPR

At last, experiments to predict the RUL of lithium-ion battery are realized. We predict the RUL of NASA batteries B0005 and B0007 and CACLE battery CS2_8 with the proposed EMGPR method.

Figure 7 shows the predicted result with NASA battery B0007. Here, the training data is from cycle 100 to cycle 140, and the estimated output is from cycle 141 to cycle 168. The covariance we use is the periodic type, as shown in Eq. (19), and the empirical model is the Gaussian model.

Figure 8 shows the experiment result on CALCE battery CS2_8. The training data is from cycle 122 to cycle 130. The predicted output is from cycle 134 to cycle 146. The covariance function is the same with that in Eq. (20). And the empirical model is double-exponential model.

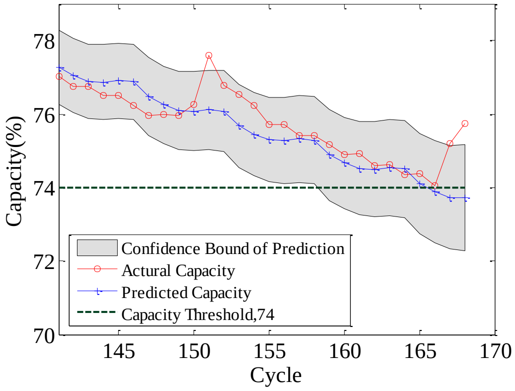


Figure 7. Predicted result of NASA battery B0007 based on EMGPR method

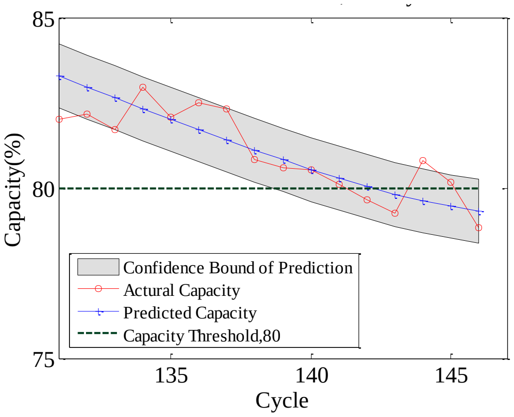


Figure 8. Predicted result of CALCE battery CS2_8 based on EMGPR method

The RMSE and MAE of predicted results for battery B0005, battery B0007 and battery CS2_8 are shown in Table 3.

Table 3. Comparison of RMSE and MAE with different prognostic methods

Battery index	RMSE	MAE
B0005	0.0805	0.0112
B0007	0.0663	0.0089
CS2_8	0.0314	0.0067

From the results above, it can be concluded that the prognostic framework of EMGPR can predict the RUL of lithium-ion battery satisfied. With this method, the estimated result is offered with uncertainty. The uncertainty expression parameter confidence bound is small. Moreover, the covariance function is easy to choose.

5. CONCLUSIONS

In this paper, a fusion prognostic framework of the combination of the GPR model and the empirical model (EMGPR) is proposed. The main contribution of this paper can be concluded as follows. (1) The GPR characteristics is studied and experiments have implemented to illustrate that the confidence bound is smaller if the training data keep closer to the test data. Thus, an important step of prediction with EMGPR approach is to choose the proper training data. (2) In the framework of the EMGPR approach, the training data are divided into two parts. One is self-deterministic which can be approximated by the empirical model (indicates the degradation trend). The other part is the disturbance components which turns out to be periodic and can be predicted with the GPR model. The periodicity of the disturbance components has a positive influence on the decision of the covariance function of GPR. As a result, the challenging selection of the covariance function can be solved. (3) The empirical models, proved to be effective in RUL prognostics of lithium-ion battery, are double-exponential model and Gaussian model. In actual application, the flow of choosing the proper empirical model should be considered. (4) Experimental results with both data from NASA PCoE and CALCE show that the EMGPR prognostic framework can predict the RUL of lithium-ion battery with high performance as well as indicated its uncertainty.

6. FUTURE WORK

In future, we will explore more effective empirical models. The more specific theory will be studied to choose the empirical model directly. Moreover, the idea of EMGPR can be extended. Other techniques such as filter, smooth theory can be combined with GPR method or EMGPR for lithium-ion battery prognostics. In addition, uncertainty representation such as probability density function (PDF) may be utilized.

ACKNOWLEDGEMENT

This work is supported partly by National Natural Science Foundation of China under Grant No. 61301205, Twelfth Government Advanced Research Fund under Grant No. 51317040302, Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20112302120027, Fundamental Research Funds for the Central Universities under Grant No. HIT.NSRIF.2014017 and China Scholarship Council.

REFERENCES

Chen, C. C., & Pecht M. (2012). Prognostics of Lithium-ion Batteries Using Model-based and Data-driving Methods, *IEEE Conference on Prognostics and System Health Management (PHM)*, June 18-21, Beijing, China.

Cristianini, N., & Taylor J. S. (2000). *An introduction to support vector machine and other kernel-based learning*. Cambridge MA: MIT Press.

Dalal, M., Ma, J., & He, D. (2011). Lithium-ion battery life prognostic health management system using particle filtering framework, *Institution of Mechanical Engineers*, vol. 225, pp. 81-90.

Do, D. V., Forgez, C., Benkara, K. E. K., & Friedrich, G. (2009). Impedance Observer for a Li-Ion Battery Using Kalman Filter. *IEEE Transaction on Vehicular Technology*, vol. 58, pp. 3930-3937.

Erdinc, O., Vural B., &Uzunoglu, M. (2009). A dynamic lithium-ion battery model considering the effects of temperature and capacity fading. *International Conference on Clean Electric Power*, June 9-11, Capri, Italy, pp.383-386.

Gao, L. J., Liu, S. Y., & Dougal, R. A. (2002). Dynamic lithium-ion Battery Model for System Simulation. *IEEE Transaction on Components and Packaging Technology*, vol. 25, pp.495-505.

Goebel, K., Saha, B., & Saxena, A. (2009). A Comparison of Three Data-Driven Techniques for Prognostics, *http://ti.arc.nasa.gov/m/pub-archive/1442h/1442%20(Goebel).pdf*.

Goebel, K., Saha, B., Saxena, A., Celaya, J. R., & Christophersen, J. P. (2009). Prognostics in Battery Health Management, *IEEE Instrumentation &Measurement Magazine*. Vol. 8, pp. 33-40.

He, W., Williard, N., Osterman, M., & Pecht, M. (2011). Prognostics of Lithium-ion Batteries Based on Dempster-Shafer Theory and the Bayesian Monte Carlo Method, *Journal of Power Sources*, vol. 196, pp. 10314-10321.

He, W., Williard, N., Osterman, M. & Pecht, M. (2011). Prognostics of Lithium-ion Batteries using Extended Kalman Filtering, *http://www.prognostics.umd.edu/calcepapers/Prognostics_Lithium-ion_Batteries_Kalman_Filtering.pdf.*

Li, J., & Zhang, Y. P. (2010). Chaotic Time Series Single-step and Multi-step Prediction Based on Gaussian Process. Acta Phys. Sin. Vol. 60, pp. 1-10.

Liu, D., Pang, J., Zhou, J., Peng, Y. (2010). Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression, Microelectronics Reliability, Vol. 53, pp. 832-839

Mohanty, S., Das, S., Chattopadhyay, A., & Peralta, P. (2009). Gaussian Process Time Series Model for Life Prognosis of Metallic Structures. *Journal of Intelligent Material Systems and Structures*, vol. 20, 887-896.

Rasmussen, C. E. (2006). *Gaussian Processes in Machine Learning*. The MIT Press Cambridge MA.

Rong, P. (2006). An Analytical Model for Predicting the Remaining Battery Capacity of Lithium-ion Batteries. *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, vol. 15, pp.441-451.

Saha, B., Goebel, K., & Christophersen, J. (2009). Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of Institute of Measurement and Control*, vol. 31, pp:293-308. doi: 10.1177/0142331208092030

Saha, B., Goebel, K., Poll, S. & Christophersen, J. (2009). Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework, *IEEE Transactions on Instrumentation and Measurement*, vol. 58, pp. 291-297.

Scholkopf, B., & Smola A. J. (2002). *Learning with Kernels*. Cambridge MA: MIT Press.

Snelson, E. L. (2007). Flexible and efficient Gaussian process models for machine learning, *Doctoral dissertation*, University of London.

Widodo, A., Shim, M. C., Caesarendra, W., & Yang, B. S. (2011). Intelligent prognostics for battery healthy monitoring based on sample entropy. *Expert System with Application*, vol. 38, pp. 11763-11769.

Xing, Y. J., Ma, E. W. M., Tsui, K. L., & Pecht, M. (2012). A Case Study on Battery Life Prediction Using Particle Filtering, *IEEE Conference on Prognostics and System Health Management (PHM)*, June 18-21, Beijing, China, pp. 1-6.

Yang, Z. G., Ye, F., Guo, H., & Ma, C. F. (2012). Progress of space power technology. *Chemical Chemical Industry and Engineering Progree*, vol. 31, pp. 1231-1237.

Zhang, J. L., & Lee, J. (2011). A review on prognostics and health monitoring of Li-ion battery. *Journal of Power Sources*, vol. 196, pp. 6007-6014.

Particle Filter Prognostic Applied in Landing Gear Retraction

Paula Borges Olivio Cerdeira[1], Roberto Kawakami Harrop Galvão[2] and João Pedro Pinheiro Malère[1]

[1] *EMBRAER S.A., São José dos Campos, São Paulo, 12227-901, Brazil*

paula.olivio@embraer.com.br
joao.malere@embraer.com.br

[2] *ITA – Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo, 12228-900, Brazil*

kawakami@ita.br

ABSTRACT

The PHM (Prognostics and Health Monitoring) applications play an increasingly important role on the aeronautical industry and can provide a wide range of benefits for complex systems, such as aircraft landing gears (LDG). Indeed forecasting the RUL (Remaining Useful Life) of the landing gear subsystems can enable condition–based maintenance, improve the aircraft availability and reduce unscheduled events. The purpose of this work is to investigate nominal and degraded simulated retraction times of a landing gear and to apply a prognostics approach, specifically the particle filter (PF) algorithm, from which the RUL can be predicted at a given confidence level.

1. INTRODUCTION

The future vision for complex systems, such as an aircraft, is the self-monitoring and control. The exploitation of Prognostics and Health Monitoring (PHM) may lead to important competitive advantages in terms of maintenance and operations (ACARE, 2010). Over the past years, the health monitoring community has vastly extended its capability to monitor systems for the improvements on forecasting and prediction (Khuzadi, 2008).

Many authors have already discussed about new paradigms for product life-cycle support. Camci, Valentine & Navarra (2007), Papakostas et al, (2010), Kalgren et al (2007) and Rodrigues, Yoneyama & Nascimento (2012) label the importance of the PHM use in aeronautical maintenance and for the decision making, by enabling effectiveness on troubleshooting, improved logistics and increased fleet availability. In other words PHM is the basis for decision support in a complex environment enabling better planning and subsequently boosting operational availability.

Figure 1 illustrates the estimation of degradation for a system being monitored and the approximation for the system RUL as a probability density function (PDF).

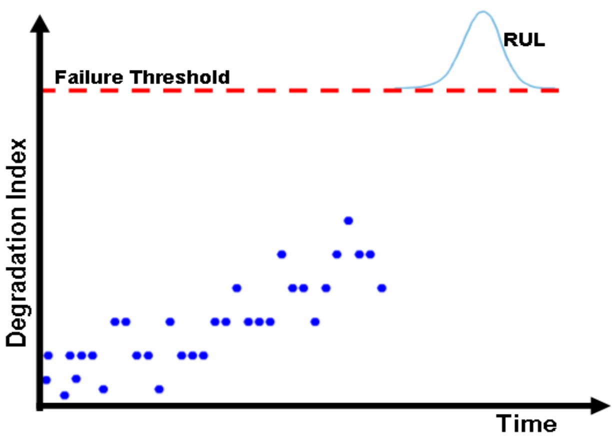


Figure 1. RUL estimation

Large complex systems such as aircraft landing gears (LDG) are composed of multiple systems and subsystems. LDG plays an important role in aircraft safety, comfort and stability. Some examples for the LDG failure modes are the blocked circuit, degraded seals, leak and vibration (Oliva et al, 2012). The aircraft LDG health monitoring may improve the aircraft dispatchability and the operational efficiency, avoiding unscheduled maintenance, Aircraft-on-Ground (AOG) events and other impacts. One example was the crash on July 2013 of the flight 345 due to nose LDG collapse during landing at LaGuardia Airport (AIRNATION.NET).

Ji, Zhang & Dong (2011) studied the LDG retraction and extension problem and the operational impact of some components degradation. Zhou, Yunxia and Rui (2011) modeled the LDG dynamics and studied structural problems

Cerdeira, P.B.O. et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

occurred by the effect of the hard landing and suggested the adoption of PHM solution for this issue.

The present work is composed by a PF algorithm for the landing gear retraction subsystem in order to estimate its RUL. A computational model proposed by Denery et al (2006) was used to simulate the dynamics and different degradation conditions of the LDG retraction subsystem.

Regarding Particle Filter (PF), it is considered state of the art in nonlinear non-Gaussian state estimation (NASA; Orchard; Vachtsevanos et al, 2006).Goebel et al (2008) made a comparison between PF and other regression methods for a battery health management, and concluded that PF is a sophisticated technique considering the accuracy for the smaller estimations. Saha & Goebel (2009) modeled Li-ion battery capacity depletion by the use of PF framework for the predictions of the EOD (end-of-discharge) and EOL (end-of-life) effectively.

This paper is organized as follow: in Section 2, a brief description of the landing gear retraction model is presented; in Section 3, the PF algorithm is explained, in Section 4 the simulation and the estimation results are discussed, followed by conclusions in Section 5.

2. Landing Gear Simulation Model

The first step for the generation of landing gear data was the reuse of an existing Landing Gear (LDG) model proposed by Denery et al (2006). The reused model is a faithful description of the right main LDG of HL20 aircraft in terms of physical representation. It was implemented in Simulink® using some blocks of the SimMechanics® tool.

Figure 2 presents the Simulink block diagram employed in the LDG model simulations.

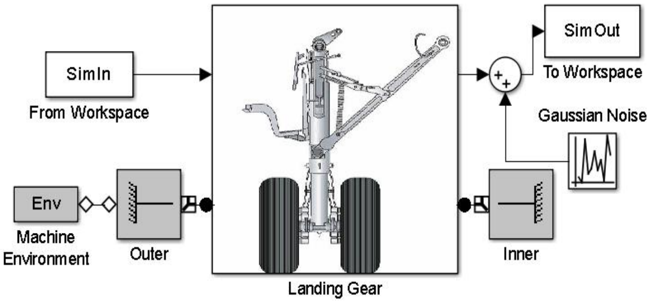


Figure 2. LDG Model

We included three special blocks on the model shown in Figure 2, compared to the original file: SimIn, SimOut and Gaussian Noise.

The SimIn block represents the landing gear actuator force and it was used to introduce degradation effects. Gaussian noise block was added to simulate imperfections in the angular displacement measurements.

The results of each simulation were stored in the Matlab workspace by using the SimOut block.

The blocks in gray are from the original model, the Inner and Outer blocks represent the aircraft structure restrictions and the Env block controls the environment, including dynamic simulation, gravity, tolerance and restrictions of the movement modes. None of the parameters inside these blocks was changed.

2.1. Input Model

Figure 3 represents the actuator input force over LDG cycles, representing the system degradation, due to loss of hydraulic pressure, for example. Each cycle corresponds to a LDG retraction. An initial nominal force retraction of 5600N was defined and in every step a random value between 0N and -50N was added.

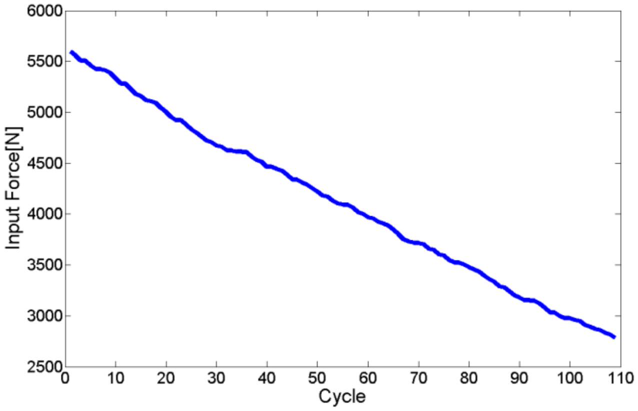


Figure 3. Input Force

The decreasing actuator input force may be associated with some failures modes. Some data observed at field events indicates linear loss of hydraulic pressure, degraded seals or valves and clogged hydraulic lines. Based on this field empirical data, the profile shown on Figure 3 was considered in this study.

2.2. LDG simulation

The LDG model outputs the retraction angle, however an algorithm takes into account this output in order to define the LDG retraction time. In this study it was defined the retraction angle that indicates the maximum degradation effect correlated to the retraction time.

The first simulation step was to establish an index, as shown in equation (1). A similar function is described by Azimi-Sadjadi et al (2000) as an uncorrelated-features assumption by the Fisher discriminant function, which in this case is used to define the angle that maximizes the degradation effect:

$$I = \left. \frac{(\mu_D - \mu_N)^2}{\sigma_D^2 \ + \sigma_N^2} \right|_{n=30} \tag{1}$$

where $\mathbf{\mu_D}$ is the mean retraction time of a set of simulations for each input force, $\mathbf{\mu_N}$ mean nominal retraction time, $\mathbf{\sigma_D}$ degraded retraction time standard deviation, $\mathbf{\sigma_N}$ nominal retraction standard deviation, **I** degradation index, ***n*** number of simulations. Figure 4(a) represents the LDG retraction simulation for the nominal input force, and shows the mean and the standard deviation of the retraction times between 0 and 70 degrees. Figure 4(b) illustrates the LDG retraction simulation with a degraded input force with the mean retraction time and the standard deviation.

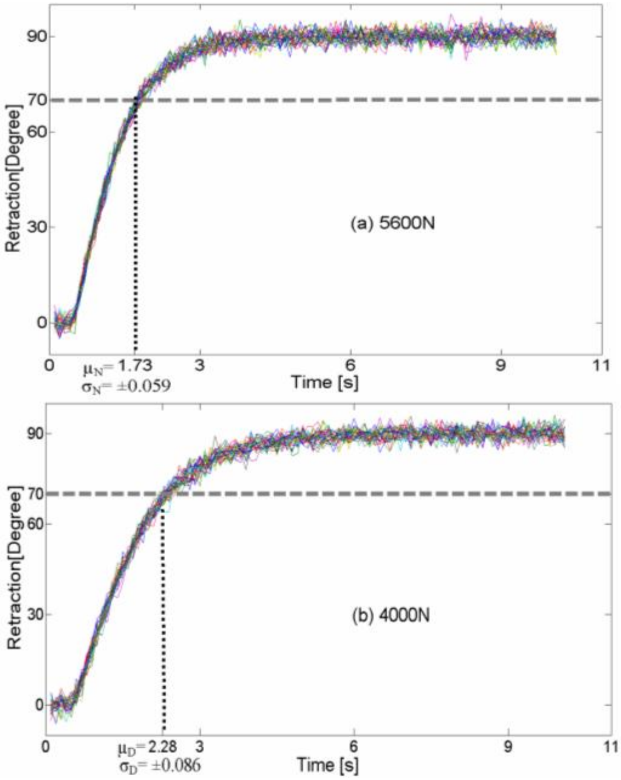


Figure 4. Comparison Simulation for LDG Retraction

Figure 4(b) shows slower retraction times than the ones shown in Figure 4 (a), what illustrates the degradation.

In order to get the best angular displacement which represents the maximum degradation index, four input forces were defined (4000N, 4500N, 5000N and 5600N). Arbitrarily, it was decided that the nominal force was 5600N and the considered degraded conditions were with forces of 5000N, 4500N and 4000N. Note that the model retraction starts with 0° and stops with 90°.

We found reasonable 30 simulations for each input force and a comparison was made among the μ and σ of the nominal force and the degraded ones, as represented in equation (1). According to the results, the 70 degrees presented the maximum index. In other words, the degradation index was established as being the retraction time from 0 to 70 degrees. Figure 5 illustrates the output of the first part of the simulation step results.

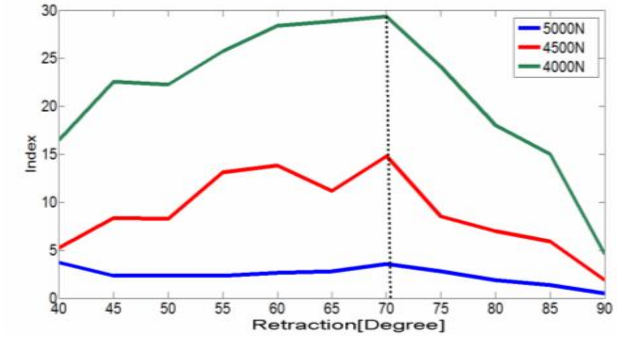


Figure 5. Best Index for Angular Displacement

The second step was to find the failure threshold. We had established the 3000N force as the minimum input force and set 30 simulations in order to get the μ for the failure retraction time and it was around 2.93 seconds. This result is the threshold input for the PF algorithm, to be detailed in Section 4.1. Note that the 3000N force was adopted as being the minimum force needed to perform the LDG retraction.

Finally, the simulations were performed with an initial actuator force and decreased by a delta random force value. The results of the LDG retraction time simulation were recorded to be used on the PF algorithm.

Table 1 shows the initial parameters used in LDG model.

Table 1. LDG Simulation Parameters

Parameter	Value	Unit
Gaussian Noise Mean	0	Degree
Gaussian Noise standard deviation	2	Degree
Time sampling	0.1	Second
Total Simulation time	10	Second
Initial Actuator Force	5600	Newton
Delta Force	(0,-50)	Newton

Figure 6 illustrates the steps of the landing gear simulation.

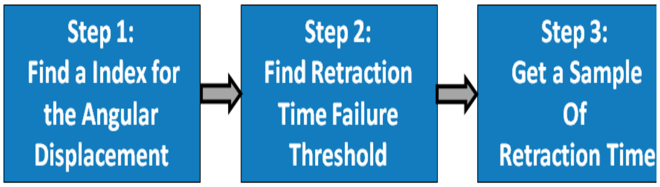

Step 1:
Find a Index for the Angular Displacement
Step 2:
Find Retraction Time Failure Threshold
Step 3:
Get a Sample Of Retraction Time

Figure 6. Schematic of LDG Simulations

3. Particle Filter

Particle Filter in model-based prognostics utilizes a concept of sequential importance sampling and Bayesian theory.

The PF algorithm involves prediction and filtering steps. The prediction step uses both previous state and the process model to generate an a priori probability density function (PDF) for the state at the next time instant, as shown in equation (2).

$$P(x_k \mid z_{1:k-1}) = \int P(x_k \mid x_{k-1}) P(x_{k-1} \mid z_{1:k-1}) dx_{k-1} \qquad (2)$$

The filtering step considers the current observations Z_k and the a priori PDF to generate the posteriori state PDF, using Bayes' formula as shown in equation (3).

$$P(x_k \mid z_{1:k}) = \frac{P(z_k \mid x_k) P(x_k \mid z_{1:k-1})}{P(z_k \mid z_{1:k-1})} \qquad (3)$$

The core idea is to construct a PDF of the state based on all available information. For nonlinear systems the particles are generated and recursively updated from a nonlinear process model that describes the evolution in time, in this case on each cycle, under analysis of a measurement (Z_k) and the *priori* estimate of the state PDF (Goebel et al., 2008).

Figure 7 summarizes the landing gear PF flowchart. The prediction parameters utilize the damage state model, as shown in equation (4), for prediction and filtering step.

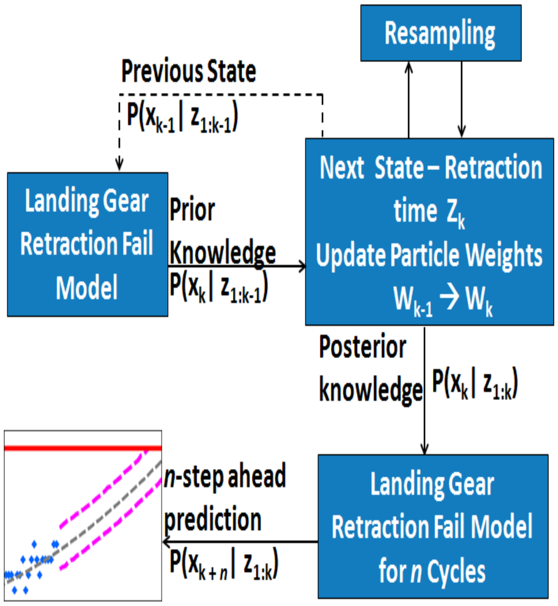


Figure 7. Particle Filter Flowchart

The landing gear retraction fail model is an empirical function, that can be written recursively in terms of the previous step, and it was obtained from the simulation measurements. It is represented in equation (4):

$$x(k) = ae^{bk} \qquad (4)$$

where $x(k)$ is the prediction parameter (LDG retraction time) of k cycles. The coefficients *a* and *b* represents the exponential damage state. More details about this model can be found in Section 4.1.

The particles are generated and updated from the *prior knowledge* of the state PDF, then propagated through landing gear cycles using the nonlinear equation (4) which is recursively updated by using the observed data (measurements). The algorithm than continues with the propagation of the particles until the failure threshold to give the RUL PDF.

Figure 8 summarizes one step for PF algorithm cycle adapted from An, Choi & Kim (2012).

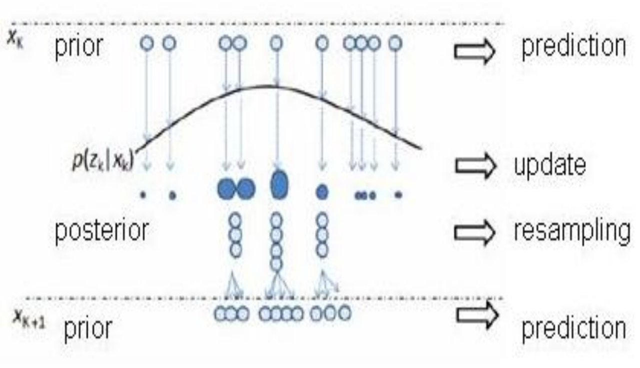


Figure 8. PF algorithm steps

4. Simulation

4.1. Simulation scenario

The LDG model described on section 2 was used to simulate the retraction times given the Gaussian noise (Figure 2) and the degradation profile (Figure 3) We established 60 points as observed data (Z_k). Figure 9 shows the measure values and the retraction time failure threshold.

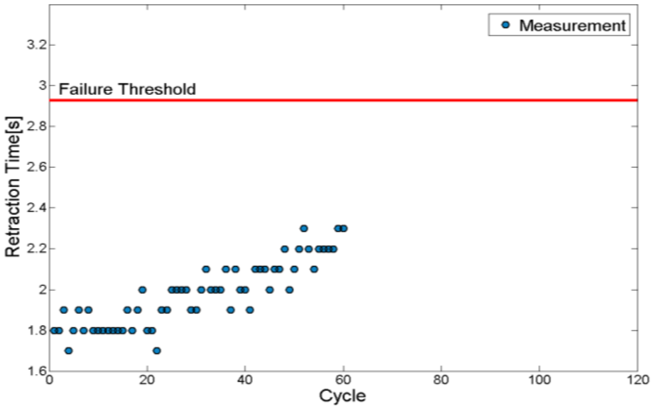


Figure 9. Retraction Time Measurement

The initial estimations for the coefficients b_0 in equation (4) were found by fitting an exponential function through the measurement values. It was adjusted in a C.I. to accommodate the samples inside it.

The initial damage value x_0 was set with the same initial value as the a_0.

Figure 10 shows the polynomial curve, as well as the confidence interval.

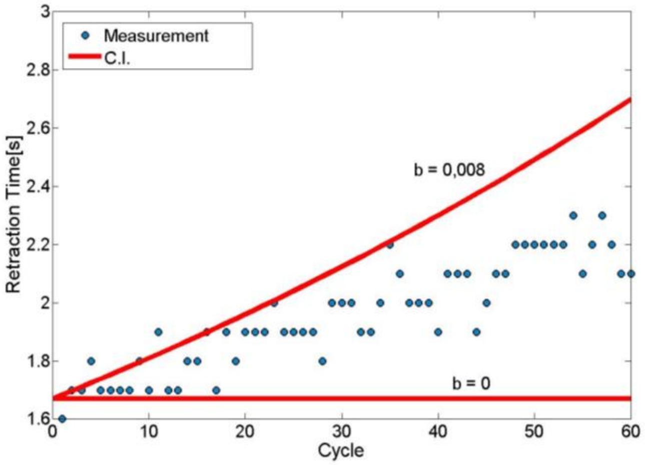


Figure 10. Exponential fitting

Table 2 shows the parameters of this first simulation considering 0.1s for the time sampling. U(α_{min},α_{max}) denotes a uniform probability density function with support [α_{min},α_{max}].

Table 2. PF Simulation Parameters

Parameter	Value	Unit
Threshold	2.93	Second
Particles	10e3	
Significance Level C.I.	95%	
Initial Damage x_0	U(1.6,1.9)	Second
Initial Parameter b_0	U(0,0.008)	
Initial Std Deviation of Measurement Error s_0	U(0,0820, 0,0902)	Second

The distribution of the standard deviation measurement error was obtained by the examination of the first 20 measure values of the LDG retraction time in a bootstrap procedure.

More two scenarios were simulated with different time sampling 0.05s and 0.2s, following the same steps as described in Section 2.2. Table 3 shows the parameters for different time sampling simulation.

Table 3. PF Simulation Parameters

Parameter	Value (0,05s)	Values (0,2s)	Unit
Threshold	2,33	2,8	Second
Initial Damage x_0	U(1,35, 1,55)	U(1,6, 1,8)	Second
Initial Parameter b_0	U(0, 0,008)	U(0, 0,008)	
Initial Std Deviation of Measurement Error s_0	U(0,0579, 0,0636)	U(0,0838, 0,0905)	Second

4.2. Simulation Results

After defining the parameters, we ran a set of simulations adjusting the significance level and compared results with true simulated data until the failure threshold.

The estimation of the RUL for the first simulated scenario, considering the 0.1s as time sampling, resulted on a median value around cycle 49 and the interval considering the significance level was between cycles 40 and 60.

We can observe in Figure 12 the predicted future damage states, which were obtained by propagating the particles though the damage model until it reaches the Failure Threshold.

Figure 12 shows the predictions of the LDG retraction time and true simulated data after cycle 60 until the failure threshold. Most of the data fits inside the prediction interval of the PF.

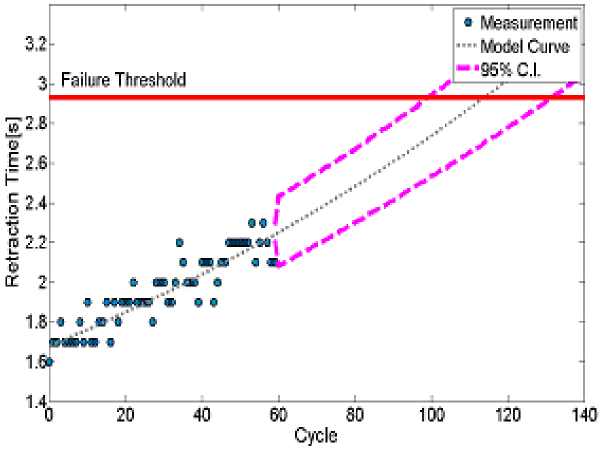


Figure 11. PF Predictions

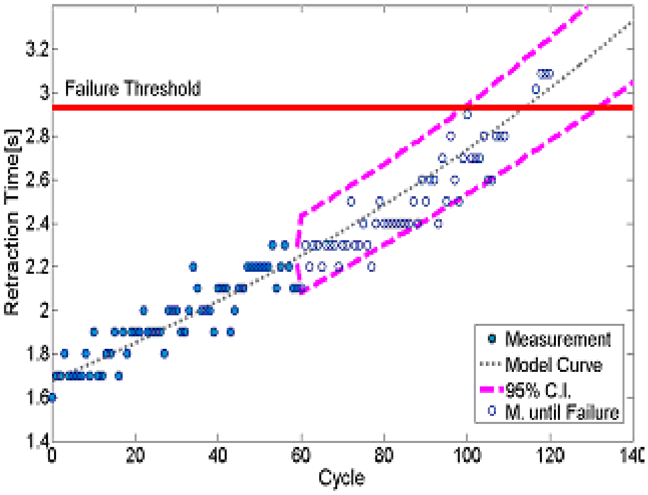


Figure 12. PF Predictions versus Measurement

A set of simulations were done using different time sampling in other to demonstrate how this parameter can interfere into the precision of the RUL. Figure 13 indicates the estimation for 0.05s time sampling and Figure 14 indicates the estimation for 0.2s time sampling.

Figure 13 shows a predicted growth rate slightly steeper and a lower dispersion with respect to the model curve when compared to the Figure 14. This analysis indicates greater dispersion between the values for samples of 0.2s and RUL forward in time.

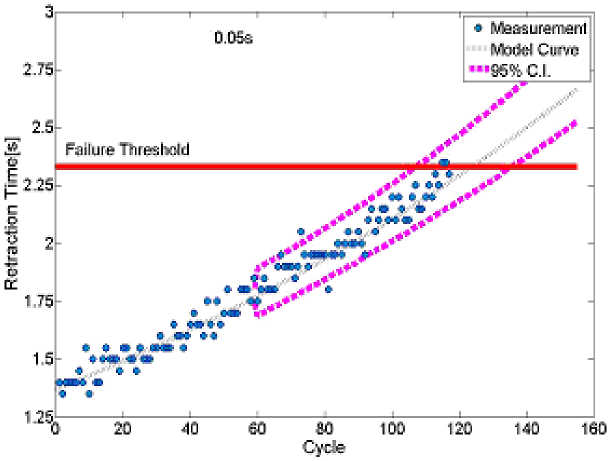


Figure 13. Estimation for 0.05s Time Sampling

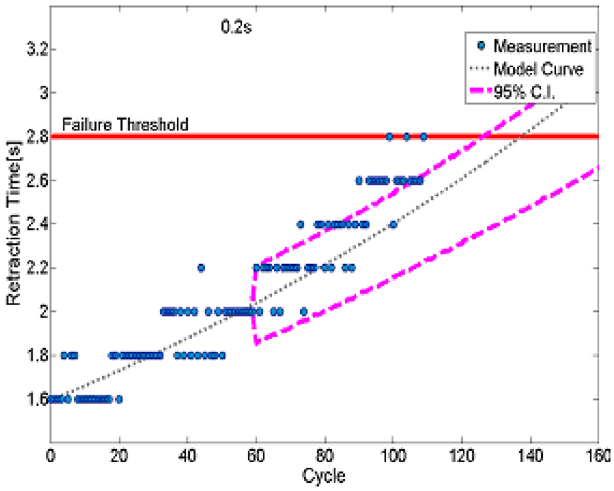


Figure 14. Estimation for 0.2s Time Sampling

Through Figure 15 we can make a comparison between the RUL for the three different time sampling simulations (0.05s, 0.1s and 0.2s). It can be noticed that the higher the interval between samplings (0.2s), the higher is the standard deviation as one can see the measurements outside the C.I crossing the failure threshold in Figure 14.

The estimated failure mean cycle for the time sampling 0.05s, 0.1s and 0.2s is 55 °, 49 ° and 79°, respectively, and the standard deviation is 6.18, 6.36 and 9.39. In this study the time sampling of 0.2s is inappropriate since the failure occurs before the estimation of the RUL.

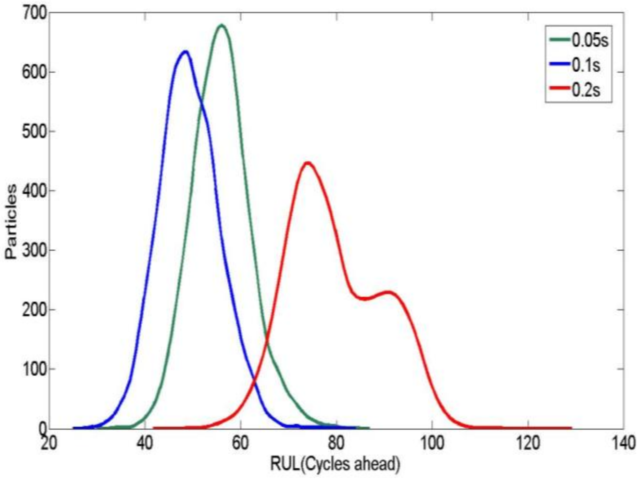


Figure 15. RUL for Different Sampling Times

5. Conclusions

This paper presented a method to obtain the remaining useful life for the landing gear retraction subsystem, based on PF techniques for estimation.

A landing gear model was used to simulate the system dynamic behavior and adapted to establish the degradation index. For the failure mode under investigation a model was presented based on the simulated data.

The results obtained by simulating the PF algorithm with measured values from the LDG model allowed a reasonable prediction level against the true data for the 0.05s and 0.1s time sampling. However the 0.2s time sampling is considered inappropriate for the remaining life prediction.

Future research may extend the proposed models for the LDG extension. To mature the algorithm, aircraft raw data may be analyzed using the PF algorithm. Finally other opportunity could include the comparison of the prediction interval using a different PHM approach.

Acknowledgment

The authors acknowledge the support of FAPESP (grant 2011/17610-0) and CNPq (research fellowship).

References

ACARE << (Advisory Council for Aeronautics Research in Europe), et al. Aeronautics and Air Transport: Beyond Vision 2020 (Towards 2050). 2010>>, http://ec.europa.eu/research/transport/pdf/acare_background_2010_en.pdf, accessed 02/02/2013.

An, D., Choi, J.H. & Kim, N. (2013). Prognostics 101: A Tutorial for Particle Filter-Based Prognostics Algorithms Using Matlab. *Reliability Engineering and System Safety*, vol. 115, pp. 161-169.

Airnation.net<<NTSB Issues Update into Southwest Flight 345 Accident at LaGuardia Airport in New York>>, http://airnation.net/2013/07/26/ntsb-southwest-345-update, accessed 09/03/2013.

Azimi-Sadjadi, M.R. et al. Underwater target classification using wavelet packets and neural networks. *IEEE Transactions on Neural Networks*, vol. 11. n.3, 2000.

Camci, F., Valentine, G., Navarra, K. (2007). Methodologies for Integration of PHM System with Maintenance Data. *IEEE Aerospace Conference,* Big Sky.

Denery, T., et al. (2006). Creating Flight Simulator Landing Gear Models Using Multidomain Modeling Tools. *The MathWorks, Inc.*

Goebel, K., et al. (2008). Prognostics in Battery Health Management. *IEEE Instrumentation and Measurements Magazine, vol. 11(4), pp.33-40.*

Ji, G., Zhang, L. & Dong M. (2011). Dynamic simulation on retraction\ extension system of an aircraft. *Proceedings of Prognostics and System Health Management*

Conference, Shenzhen.

Kalgren, P., et al. (2006). Defining PHM, a lexical evolution of maintenance and logistics. *IEEE Aerospace Conference,* Big Sky.

Khuzadi, M. (2008). On Health Monitoring. *IEEE Aerospace Conference,* Big Sky.

Nasa,« Batteries Prognostic - Particle Filtering», http://ti.arc.nasa.gov/tech/dash/pcoe/battery-prognostics/algorithms/ , accessed 05/11/2013.

Oliva, G. M., et al. (2012). Prognostics Assessment Using Fleet-wide Ontology. *Proceedings of International Conference on Prognostics and Health Management,* Minneapolis.

Orchard, M. & Vachtsevanos, G. (2009). A Particle Filtering Approach for Online Fault Diagnosis and Failure Prognosis. Transactions of the Institute of Measurement and Control, no. 3-4, p. 221–246.

Papakostas, N. et al. (2010). An Approach to Operational Aircraft Maintenance Planning. Decision Support Systems, Vol. 48, Issue 4, Elsevier Science Publishers B.

Rodrigues, L. R., Yoneyama, T. & Nascimento Jr., C. L. (2012). How Aircraft Operators Can Benefit from PHM Techniques. *IEEE Aerospace Conference,* Big Sky.

Saha, B. & Goebel, K. (2009). Modeling Li-Ion battery capacity depletion in a particle filtering framework. *Proceedings of International Conference on Prognostics and Health Management,* San Diego.

Vachtsevanos, G., Lewis, F.L., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent Fault Diagnosis and Prognosis for Engineering Systems.* Hoboken, NJ: John Wiley & Sons, Inc

Zhou, Y., Yunxia, C. & Rui, K. (2011). A Study of Aircraft Landing Gear Testing System on PHM. *Prognostics and System Health Management Conference, Shenzhen.*

BIOGRAPHIES

Paula Borges Olivio Cerdeira holds a bachelor’s degree in Computer Engineering from Centro Universitário e Fundação Santo André (FSA, 2007), Brazil. She is currently pursuing her Master Degree in Aeronautical Engineering from Instituto Tecnológico de Aeronáutica at ITA. She is with EMBRAER S.A. São José dos Campos, São Paulo, Brazil, since 2011. She works as a Development Engineer in an R&T group at EMBRAER performing research on IVHM technology for application to aeronautical systems. Her current research interests are the application of health monitoring techniques for systems and the usage of PHM information for decision making.

Roberto Kawakami Harrop Galvão is an Associate Professor of Systems and Control at the Electronic Engineering Department of ITA. He holds a bachelor’s degree in Electronic Engineering (Summa cum Laude, 1995) from Instituto Tecnológico de Aeronáutica (ITA), Brazil. He also obtained the master’s (1997) and doctorate (1999) degrees in Systems and Control from the same institution. Since 1998 he has been with the Electronic Engineering Department of ITA as a full-time academic. Dr. Galvão is a Senior Member of the IEEE and an Associate Member of the Brazilian Academy of Sciences. He has published more than 200 papers in peer-reviewed journals and conference proceedings. His main areas of interest are fault diagnosis and prognosis, wavelet theory and applications, and model predictive control.

João Pedro Pinheiro Malère holds a bachelor’s degree in Control Engineering from Universidade Estadual de Campinas (Unicamp, 2004), Brazil, and a Master Degree in Aeronautical Engineering from Instituto Tecnológico de Aeronáutica (ITA, 2007), São José dos Campos, São Paulo, Brazil. He is with EMBRAER S.A., São José dos Campos, São Paulo, Brazil, since 2006. He works as a Development Engineer in an R&T group at EMBRAER performing research on IVHM technology for application to aeronautical systems. His current research interest is on integrated health management systems.

Filter Clogging Data Collection for Prognostics

O. F. Eker[1], F. Camci[1], and I. K. Jennions[1]

[1] *IVHM Centre, Cranfield University, UK*
o.eker@cranfield.ac.uk
f.camci@cranfield.ac.uk
i.jennions@cranfield.ac.uk

ABSTRACT

Filtration is a critical process in many industrial systems to obtain the desired level of purification for liquids or gas. Air, fuel, and oil filters are the most common examples in industrial systems. Filter clogging is the main failure mode leading to filter replacement or undesired outcomes such as reduced performance and efficiency or cascading failures. For example, contaminants in fuel (e.g. rust particles, paint chips, dirt involved into fuel while tank is filling, tank moisture rust) may lead to performance reduction in the engine and rapid wear in the pump. Prognostics has potential to avoid cost and increase safety when applied to filters. One of the main challenges of prognostics is the lack of failure degradation data obtained from industrial systems. This paper presents the process of design and building of an experimental rig to obtain prognostics data for filter clogging mechanism and data obtained from the rig. Two types of filters have been used during the accelerated filter clogging and 23 run-to-failure data have been collected. Flow rate and pressure sensors are used for condition monitoring purposes. The filter clogging has been recorded through a camera to evaluate the findings with pressure and flow sensors. The data collected is very promising for development of prognostics methodologies.

1. INTRODUCTION

Filtration is basically described as a unit operation that is separation of suspended particles and fluid utilizing a medium where only the liquid can pass (Cheremisinoff, 1998). Filtration phenomenon is interest of several engineering processes including automotive, chemical, reactor, and process engineering applications. Besides, several industrial applications such as food, pharmaceuticals, metal production, and minerals embrace filtration process (Sparks, 2011).

Sharing an important role with pumps, fuel filters filtrate dirt, contaminants in the fuel system such as rust & dust particulate which has been released by holding tank, pipe work, paint chips, tank moisture, or other numerous type of dirt has been delivered via the supply tanker (Wilfong et al., 2010). Consequences like engine & pump performance reduction due to increased abrasion and inefficient burning in the engine are the main motivation for necessity of fuel filtration. System flow rate and engine performance decreases once a fuel filter is clogged where it doesn’t function well in its desired operation ranges. A picture of clogged and clean filter is shown in Figure 1. The difference in between clogged and clean mesh is visible. Fuel filters replaced & cleaned on a regular basis. Monitoring and implementation of prognostics have the potential to avoid costs and increase safety.

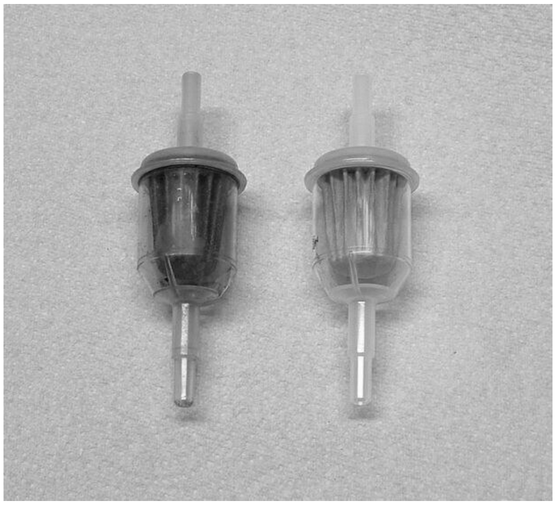


Figure 1. Clogged vs. clean filter (WikiHow, 2013)

The rest of the paper is organized as follows. Section two gives a brief literature review of prognostics in condition-based maintenance (CBM), and literature of physics-based modeling on filter clogging. Section three discusses in detail the filter clogging experimental scenario under accelerated aging conditions. The paper concludes with discussion of the results and future work.

Eker et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. LITERATURE REVIEW

Condition-based maintenance (CBM) is a predictive maintenance strategy, where the maintenance tasks are performed when the need arises. The necessity concept is determined by assessing and trending the health condition of the equipment (Camci and Chinnam, 2010; Eker et al., 2011). Although CBM minimizes the spare part costs, time spent on maintenance and system downtime; it has difficulties such as modeling failures or installing and using of monitoring equipment.

Diagnostics and prognostics are the two major disciplines of CBM. Diagnostics involves identifying degradation and current health status of asset as well as revealing their causes and locations which is relatively mature area compared to prognostics. It aims to stop and schedule a maintenance task for the system once an abnormality has been detected or let the system continue otherwise. In general, incipient failures follow a slow degradation path (Kwan et al., 2003). Detection of failure progression is more valuable compared to the detection of failure once it has reached to severe point. Furthermore, it is a prerequisite for prognostics (Xiong et al., 2008).

Prognostics is forecasting the systems or components future health level by trending the current health level up to failure status and predicting the remaining useful life (RUL). It is considered to be one of the most challenging and key enabling technology among the other phases of CBM (Zhang et al., 2006; Peng et al., 2010; Daigle and Goebel, 2010).

Several studies on modeling of filtration process and fuel system investigations have found in the literature. (Roussel et al., 2007) has experimentally explained the general clogging process as a function of; ratio of particle to mesh hole size (D), solid fraction (Ø), and the number of grains arriving at each mesh hole during one test (N_e). They conducted several clogging experiments and optimized the clogging parameters in their model. (Park, 2002) has investigated F-5F aircraft engine failure caused by erosion-corrosion of a fuel manifold. They claimed that engine failures are caused by sudden pressure drop due to particles (mostly steel and iron) from the welding beads of fuel manifold. Internal welding beads are corroded and metal particles spread out which makes the fuel pump failed. Results are obtained by using EDX analysis of related surfaces. A comprehensive investigation of UAV fuel systems has been conducted in IVHM Centre, Cranfield University, UK (Niculita et al., 2012; Niculita et al., 2013). Several failure scenarios including clogged filter and faulty gear pump has been investigated and mostly diagnostics based studies have been conducted. Direct Proportional Valves (DPV) that have the ability to mimic fuel filter blockage have been utilized to imitate the clogging filter scenario. Darcy's Law and Kozeny-Carman Equations are two commonly used formulations applied in in the field of fluid dynamics to model the pressure drop of a fluid flowing through a porous medium (Carman, 1997). (Sappok et al., 2010) worked on the effects of ash accumulation in diesel particulate filters (DPV). They presented the results of their detailed measurements with formulated lubricants, correlating ash properties to individual lubricant additives and their effects on filter pressure differentiation. (Pontikakis et al., 2001) developed a mathematical model for dynamic behavior of filtering process in ceramic foam filter. The model is able to estimate the filtration efficiency, accumulation of particle mass in the filter, and pressure drop throughout a filter.

3. FILTER CLOGGING EXPERIMENTAL RIG

This section discusses in detail the filter clogging experimental scenario under accelerated aging conditions. We study the effects of different pump speed and solid ratio. Accelerated degradation data collecting mechanism for prognostic purposes is discussed in this section.

3.1. Design & Installation

Filter clogging system was designed as cyclical in order to have continuous flow so that system flow continues till the filter has fully clogged. It's not necessary to add particles or water as shown in Figure **3** and Figure **4**. Figure **3** is the top down perspective view of the real test rig with the components tags.

A peristaltic pump was installed in the system to maintain the flow of the suspension prepared. The pump sucks the suspension with a desired flow rate and pumps it through the filter and letting the suspension pour into the tank back. Pressure sensors across the filter and a flow meter was installed in the system to measure the pressure difference and flow rate during the clogging.

A colorful (necessary for image processing), solid type of particles with a desired size range, having the features of; low water absorption level, closer to water density level, and water-insoluble, was determined to be the optimal way for the particle selection process. Polyetheretherketone (PEEK) particles have been selected among several other types of solids. The 450PF PEEK particle size distribution is shown in Figure 2. A stirrer was installed in order to sustain constant amount of particles flowing through the system. Stirrer was necessary since the PEEK particles sink after a while leaving the water clean due to the density difference.

Pressure and flow rate values are considered to be health indicators of filter degradation. Pressure difference across the filter is expected to increase and flow rate is expected to be decreased during a clogging process. A high quality macro lens camera was installed in the system and macro pictures of the filter were taken in every two seconds. A macro picture and zoomed view of a filter mesh is shown in Figure **5**.

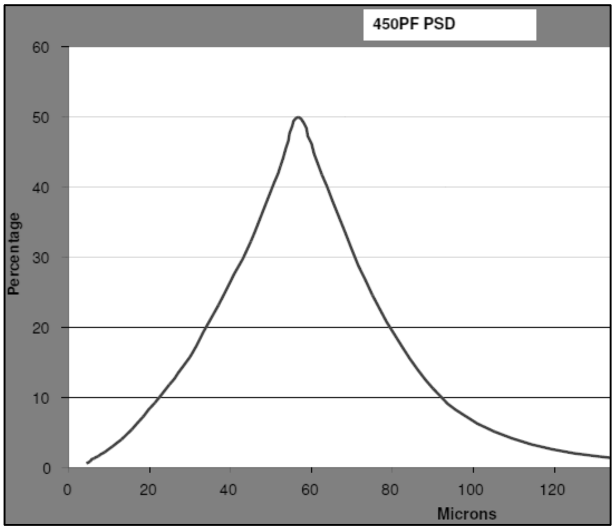


Figure 2. PEEK particle size distribution

The mesh inside the filter can clearly be captured and it can be utilized in image processing applications for clogging rate calculations which gives the ground truth information of clogging. Other sensory information can be compared with the clogging rate obtained from the macro pictures dataset.

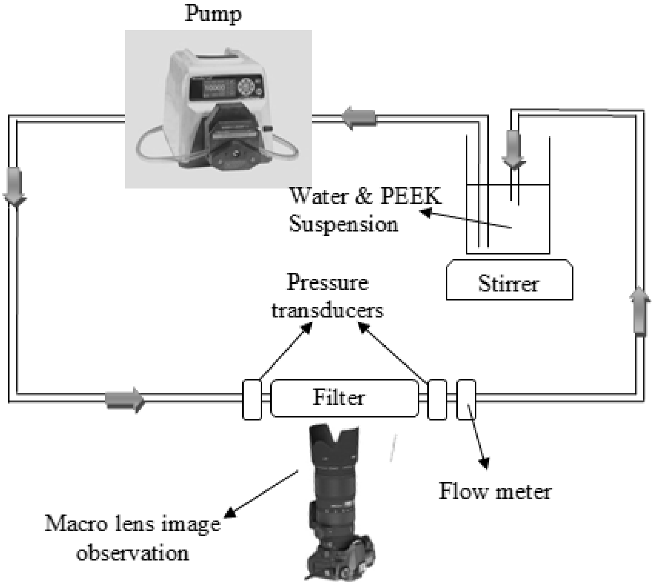


Figure 3. Filter clogging rig system design

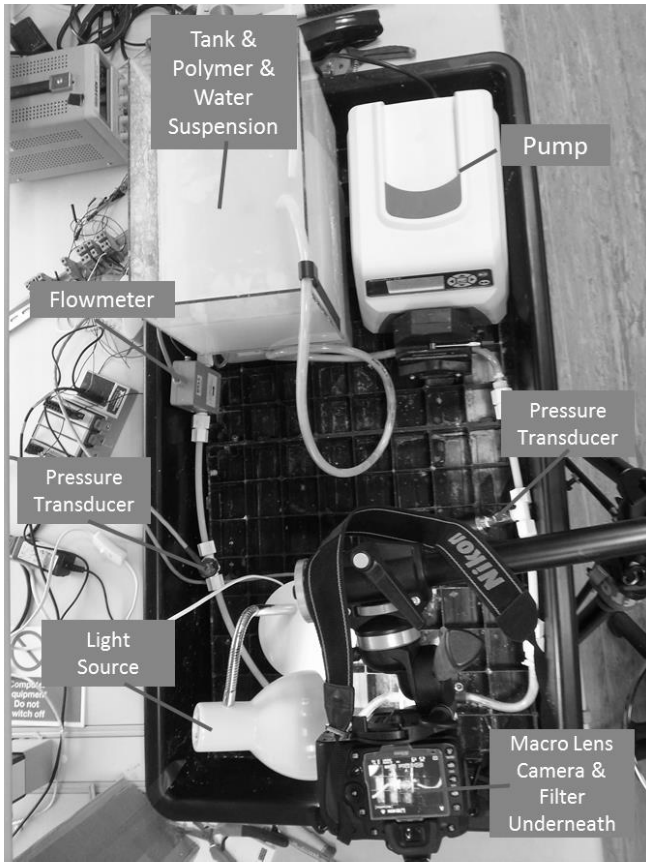


Figure 4. Filter clogging test rig & components

Components of the system were selected so that no other component will degrade other than the filter due to the solid particles inside the liquid. A peristaltic pump and magnetic flow meter were selected since they don’t get affected by the debris inside the liquid. Peristaltic pump compresses the flexible tube connected to the rotor in which the pump does not interconnect with the suspension. Similarly, magnetic flow meters don’t have moving parts inside where existences of particles are not harmful to the sensor at all. The system design shown in Figure 3 was selected among several other types of designs since it promises better solution for degradation process and requires less human labor.

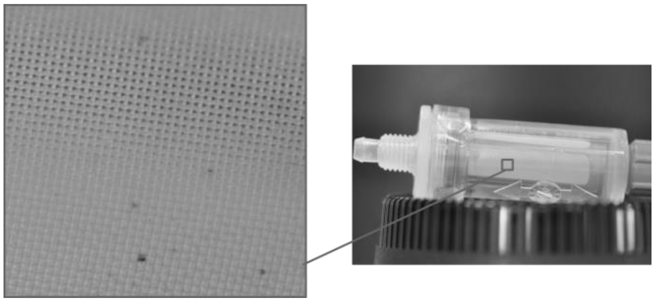


Figure 5. Zoomed capture of a filter mesh

125 micron pore sized car fuel filters have been selected to be used in the system as shown in Figure **6**. The ratio of filter pore size to the mean size of particles is higher than 1 which means the clogging process is stochastic since the majority of particles can flow through the mesh without being captured. Bridges are being formed and clogging is occurred when the considerable amount of particles approaches to each hole (Roussel et al., 2007).

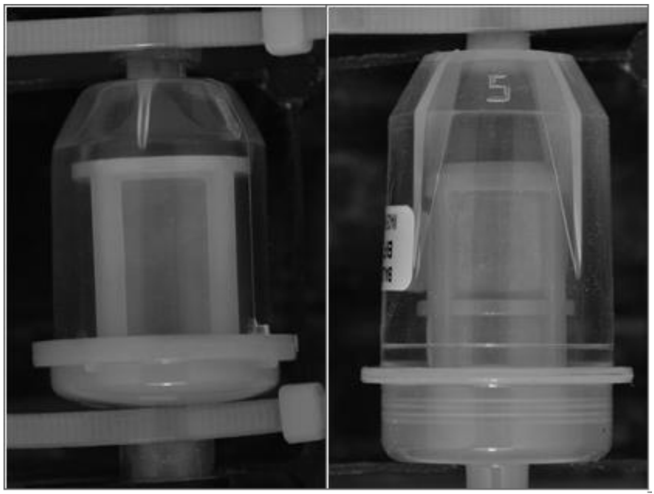


Figure 6. Filters type 1 and type 2

A box was designed to cover the filter area. The interior side of box was covered with a white colored material and a light source was directed inside the box to provide a constant uniform light so that the filter is isolated from varying environmental light.

3.2. Data Collection

This section provides the details of accelerated clogging experiments of two types of fuel filters.

125 micron pore sized two type of fuel filters have been utilized for clogging experiments in the lab environment and pressure and flow rate measurements have been collected. Details of the operating conditions are shown in Table 1. Maximum amount of particle capturing levels for the filters have been measured and the suspension solid ratio have been prepared considering these values. 10 and 12 run-to-failure experiments have been conducted for filters type 1 and type 2 respectively. Each clogging experiment has run and monitored till the pressure drop (e.g. Differential Pressure, ΔP) has reached to peak value and remains stable for a while as shown in Figure **7**.

Basically ΔP values calculated as subtracting the pressure values obtained from pressure sensor 1 (i.e. upstream pressure) from the pressure sensor 2 (i.e. downstream pressure) values. Upstream pressure and ΔP values show similar degradation profile since pressure values collected from pressure sensor 2 are quite low range compared to upstream pressure. Figure 7 plots show the raw measurements obtained from these three sensors. Upstream and downstream pressure values show linear degradation for the majority of lifetime and exponential degradation behaviors are seen after that as shown in plot 1 and plot 2, Figure **7**. Exponential degradation behavior of downstream pressure measurements can only be seen in the zoomed version (i.e. plot 2, Figure **7**). On the other hand, flow rate values remain same for a long period of time and show a dramatic drop at the end of experiments due to clogging.

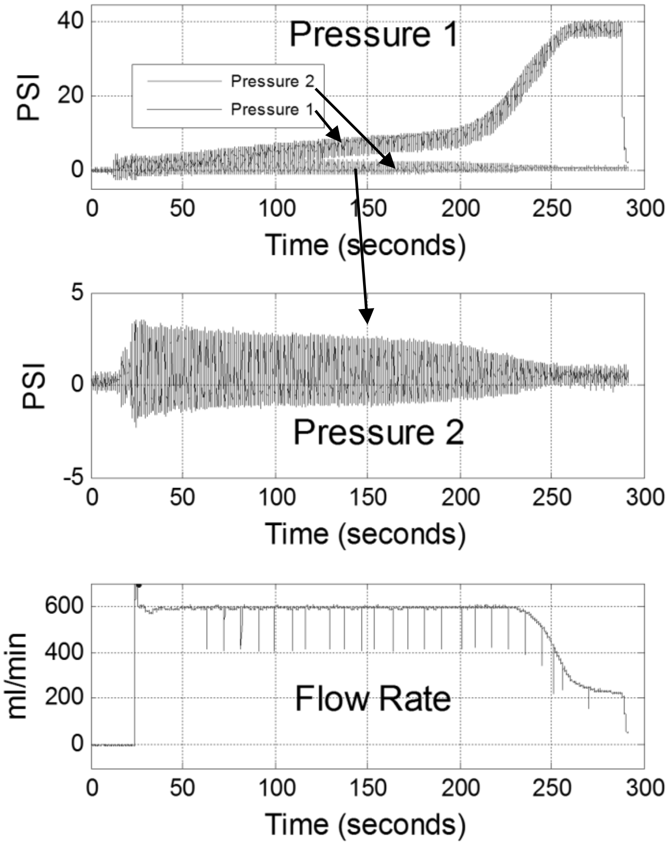


Figure 7. Raw sensor measurements

Macro lens camera was set to take pictures once in two seconds during each clogging test. Experiments were stopped due to prevent potential failures in the other components of the system. Examples of the pictures during the filter clogging are shown in Figure 10. Clogging of the filter can be seen clearly by examining the pictures. Image processing techniques will be implemented on these pictures to obtain the ground truth information regarding the filter clogging level. Accelerated aging experiments took around four minutes with the operating conditions mentioned in Table 1. At the end of experiments, flow rate values are dropped 57% and 48% in average for filters type1 and type2 respectively.

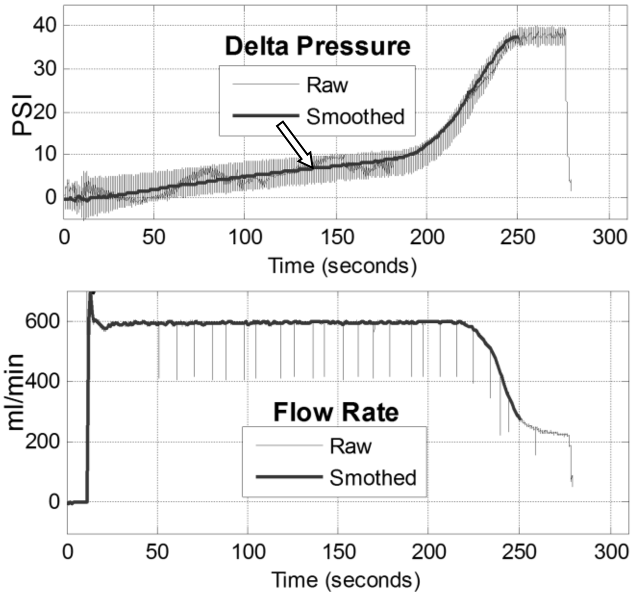


Figure 8. Filter type1 experiment 6 smoothed vs. raw data

Table 1. Operating conditions

Filter Type	Type 1	Type 2
Pore Size (micron)	125	125
Mean Particle Size (micron)	60	60
Max. amount of particles captured by the filter (g)	5.2	7.0
Total weight of particles in the tank (g)	26.0	21.0
Water Weight in the tank (g)	7507.9	7500
Solid Ratio (beginning)	0.35%	0.28%
Solid Ratio (end of clogging)	0.28%	0.19%
Solid Ratio change (max)	20%	33%
Flow Rate (RPM)	211	211

Smoothing has been performed on the raw dataset obtained from experiments. It has been done by taking the median values of each 100 data points since the raw data was collected with 100 Hz sample rate. Smoothed vs. raw data plot is shown in Figure 8 and Figure 9. Figure 9 depicts the zoomed two seconds (200 data points) of an experiment and corresponding two smoothed points. In addition, it shows the pressure peak reflections of the pump pulses.

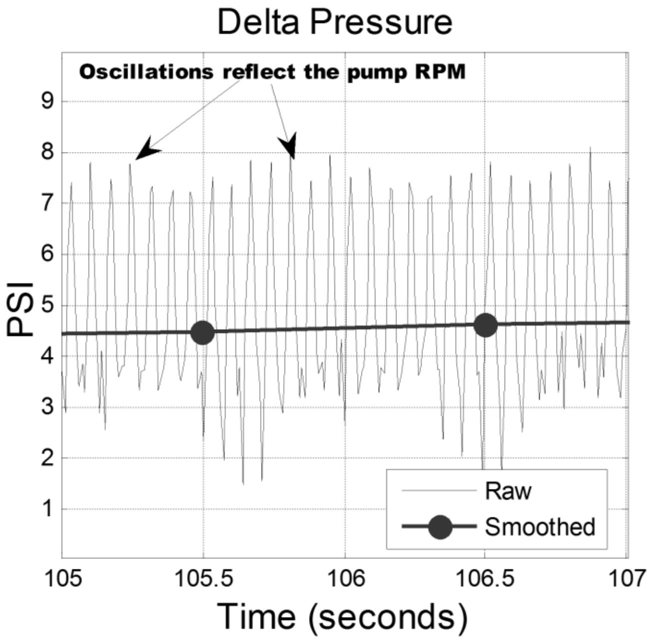


Figure 9. Zoomed pressure plot of a sample

Figure 11 and Figure 12 depict ΔP and flow rate measurements obtained from filter type one and two respectively.

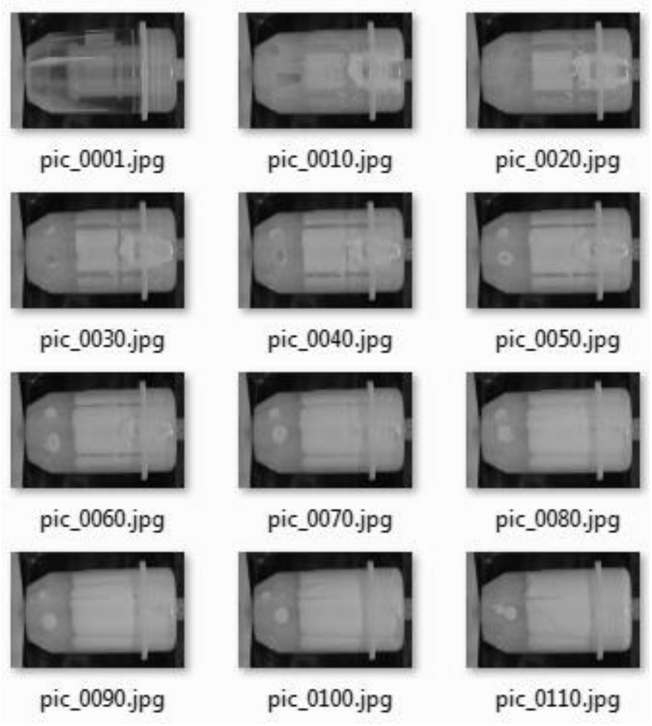


Figure 10. Filter pictures during an experiment

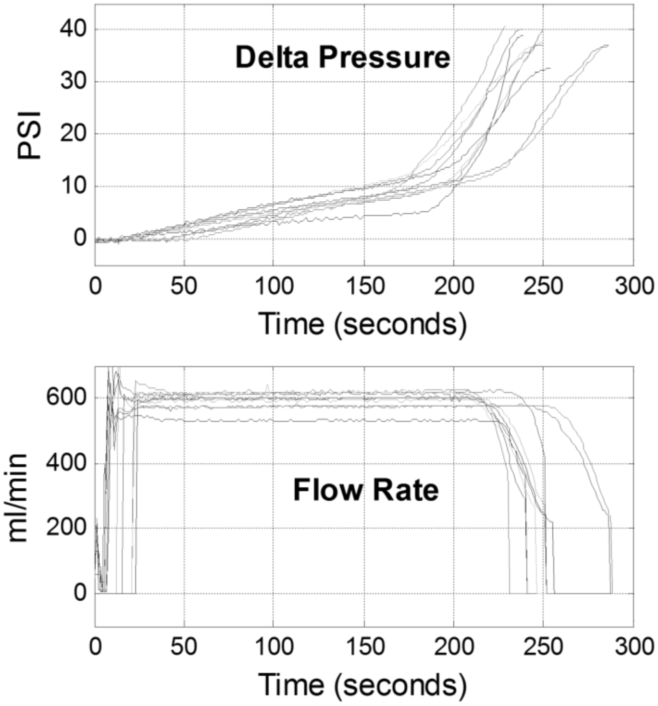


Figure 11. Filter type1 smoothed data

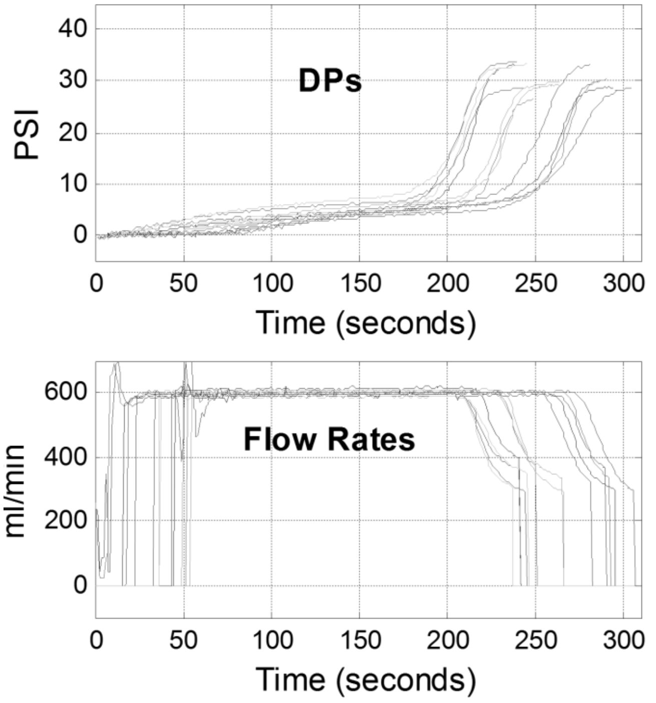


Figure 12. Filter type 2 smoothed data

3.3. Discussions Regarding the Data Collection & New Rig Design and Experiments

After analyzing the dataset we collected, it has been realized that the data collection can be improved more. Discussions about the dataset are summarized in Table 2.

Table 2. Dataset discussions & action table

Challenge	Action	Goal
Picture quality & light reflection problem	Covering the filter area with a chamber	Image processing improvements
Keeping the operating condition same	Several enhancements done listed below	Reproducibility
Change in solid ratio	Putting another tank in the system	Physics-based modelling enhancement
High variety in filter lifetimes (22% - 30%)	Some enhancements done listed below	• Reproducibility • Improvements in prognostic modelling
High oscillation amplitudes in pressure measurements	Putting dummy filters or chambers in the line before upstream pressure sensor	• Lowering oscillation amplitude
Vague clogging pictures	Testing of lower solid ratios	To see the degradation clearly on the macro pictures
Bubbles emerge in the filter	Filling all the system with a clean water prior to tests	• Avoiding the bubble effect on clogging of filter mesh • Reproducibility
Invisible flow movements	Using colourful liquid	To track the flow inside the filter during the clogging

Several other experiments have been conducted by taking into account of these concepts. First we have started changing the design of the test rig by putting another tank in the system. In the main tank we prepared the suspension with a lower solid ratio (i.e. 0.14%) in order to see the clogging process more clearly in the pictures and have longer lifetime for filters. Another tank was initially empty in the beginning of each test. End of tubing has been connected to other tank so that the solid ratio did not change during the experiment. Test rig design was cyclical for the previous experiments. End of tubing was connected to the same tank so solid ratio used to change during the experiment due to considerable amount of particles was being captured by the filter. In the new experiments, suspension was prepared with a specific solid ratio so that the main tank became almost empty at the end of each experiment ended up with a clogged filter. We utilized only filter type 2 since the mesh material quality and robustness is quite higher than the type 1 filter.

It has been seen that the length of soft tubing was affecting the life variety of experiments, particle capture rate, and also the pressure measurement oscillation amplitude. We didn't have chance to avoid using soft tubing since the peristaltic pump mechanism requires soft material in order to pump it by squeezing it. Hereupon we have made the length of soft tubing as short as possible and constant for each experiment and results got better and have become reproducible as shown in Figure 13. In that figure smoothed pressure and flow rate measurements of six different run-to-failure experiments are shown. And the lifetime variation in these experiments are reduced to 12% (variation was 25% in average before).

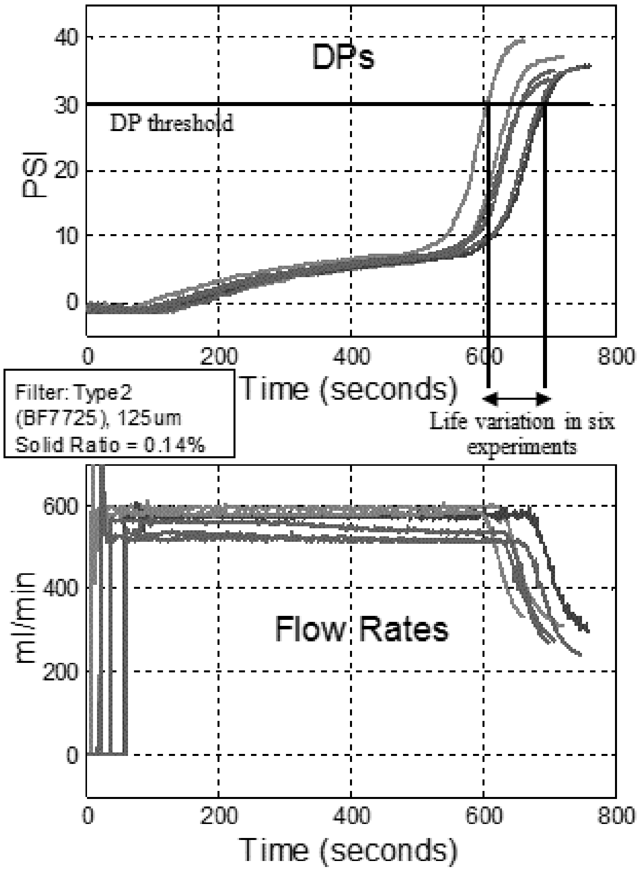


Figure 13. New experiments with constant solid ratio

After these experiments we have tried to reduce the fluctuation amplitude which would make problems in physics-based modeling and reproducibility concerns. In order to do that we installed dummy filters in the system just before the upstream pressure sensor. Basically a dummy filter is an empty chamber which helps to reduce the pump pulse effects. Addition of dummy filter in the line helped to decrease the fluctuation amplitudes in pressure measurements. Comparison of pressure values with or without dummy filter is shown in Figure 14. Pressure fluctuation due to the pump reduces dramatically with a dummy filter. Adding a second dummy doesn't change as the way the first one did. That amplitude (~2PSI) of fluctuation in pressure is similar to the flow created by the atmospheric pressure. We did an experiment with the atmospheric pressure, where the pump wasn't used. It gave the similar amplitude to the experiments done with one or two dummy filters.

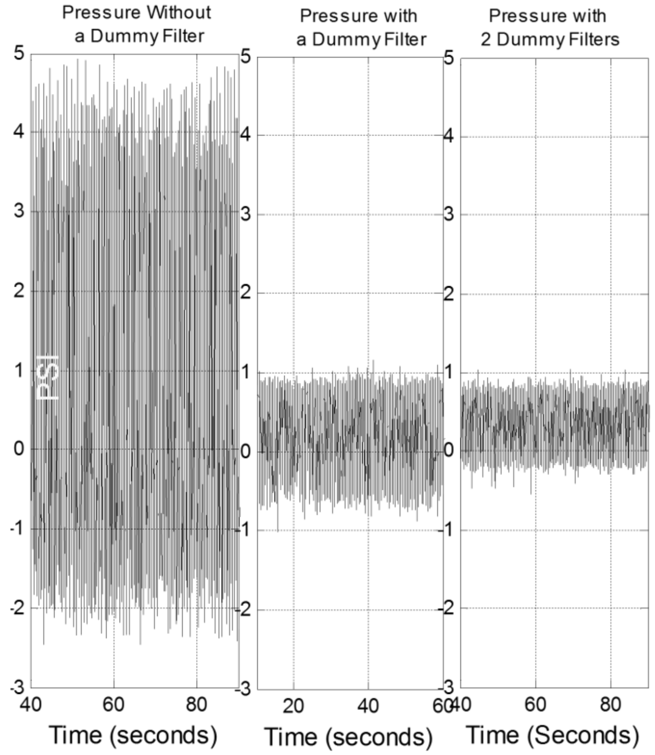


Figure 14. Dummy filter effect comparison

Finally we have tried to get rid of bubbles occur inside the filter. In order to do that, we filled all the system with clean water prior to the tests. We made sure that no air left in the system. Then we managed to get rid of bubbles. However fluctuations got back to the same level similar to the system without a dummy filter which means the dummy filter with full of water started reflecting the pressure pulses generated by the pump.

4. CONCLUSION & FUTURE WORK

This paper presents design & data collection process for filter clogging phenomenon to be used in prognostic application. Accelerated filter clogging experiments have been conducted. Pressure and flow rate signals have been collected during the experiments and macro pictures of filters have been taken too. The experiments were continued till each filter got clogged and considerable amount of reduction in the flow rate was monitored at the end of each clogging. Operating and environmental conditions are kept same. Two different types of fuel filters have been clogged using PEEK polymer particles & water suspension. Several enhancements have been done on data collection & design to improve the reproducibility and to help the physics-based modeling. Results show that the enhancement actions taken provide better reproducibility and an improved dataset. Accelerated filter clogging experiments are planned to be conducted with the latest operating conditions and test rig design. Data-driven, physics-based and hybrid modeling studies will be done on the final dataset to be collected.

Acknowledgement

This research was supported by the IVHM Centre, Cranfield University, UK and its industrial partners.

References

Camci, F. and Chinnam, R. B. (2010), "Health-state estimation and prognostics in machining processes", *IEEE Transactions on Automation Science and Engineering,* vol. 7, no. 3, pp. 581-597.

Carman, P. G. (1997), "Fluid flow through granular beds", *Chemical Engineering Research and Design,* vol. 75, no. 1 SUPPL., pp. S32-S46.

Cheremisinoff, N. P. (1998), *Liquid Filtration* , Second Edition ed, Elsevier Inc.

Daigle, M. and Goebel, K. (2010), "Model-based prognostics under limited sensing", *IEEE Aerospace Conference Proceedings,* .

Eker, O. F., Camci, F., Guclu, A., Yilboga, H., Sevkli, M. and Baskan, S. (2011), "A simple state-based prognostic model for railway turnout systems", *IEEE Transactions on Industrial Electronics,* vol. 58, no. 5, pp. 1718-1726.

Kwan, C., Zhang, X., Xu, R. and Haynes, L. (2003), "A novel approach to fault diagnostics and prognostics", *Proceedings - IEEE International Conference on Robotics and Automation,* Vol. 1, pp. 604.

Niculita, O., Irving, P. and Jennions, I. K. (2012), "Use of COTS Functional Analysis Software as an IVHM Design Tool for Detection and Isolation of UAV Fuel System Faults", *Annual Conference of the Prognostics and Health Management Society 2012,* Vol. 3, Sep 22-27, Minneapolis, USA, pp. Paper #105.

Niculita, O., Jennions, I. K. and Irving, P. (2013), "Design for diagnostics and prognostics: A physical-functional approach", *Aerospace Conference, 2013 IEEE,* pp. 1.

Park, M. (2002), "Engine failure caused by erosion-corrosion of fuel manifold", *Engineering Failure Analysis,* vol. 9, no. 6, pp. 673-681.

Peng, Y., Dong, M. and Zuo, M. J. (2010), "Current status of machine prognostics in condition-based maintenance: A review", *International Journal of Advanced Manufacturing Technology,* vol. 50, no. 1-4, pp. 297-313.

Pontikakis, G. N., Koltsakis, G. C. and Stamatelos, A. M. (2001), "Dynamic filtration modeling in foam filters for diesel exhaust", *Chemical Engineering Communications,* vol. 188, pp. 21-46.

Roussel, N., Nguyen, T. L. H. and Coussot, P. (2007), "General probabilistic approach to the filtration process", *Physical Review Letters,* vol. 98, no. 11.

Sappok, A., Rodriguez, R. and Wong, V. (2010), "Characteristics and effects of lubricant additive chemistry on ash properties impacting diesel particulate filter service life", *SAE International Journal of Fuels and Lubricants,* vol. 3, no. 1, pp. 705-722.

Sparks, T. (2011), *Solid-Liquid Filtration: A User's Guide to Minimizing Cost &amp; Environmental Impact, Maximizing Quality &amp; Productivity,* First Edition ed, Elsevier Science & Technology Books.

Wilfong, D., Dallas, A., Yang, C., Johnson, P., Viswanathan, K., Madsen, M., Tucker, B. and Hacker, J. (2010), "Emerging challenges of fuel filtration", *Filtration,* vol. 10, no. 2, pp. 107-117.

WikiHow, 2013, http://www.wikihow.com/Replace-the-Fuel-Filter-on-an-Aircooled-Volkswagen-Beetle

Xiong, Y., Cheng, X., Shen, Z. J., Mi, C., Wu, H. and Garg, V. K. (2008), "Prognostic and warning system for power-electronic modules in electric, hybrid electric, and fuel-cell vehicles", *IEEE Transactions on Industrial Electronics,* vol. 55, no. 6, pp. 2268-2276.

Zhang, L., Li, X. and Yu, J. (2006), "A review of fault prognostics in condition based maintenance", *Proceedings of SPIE - The International Society for Optical Engineering,* Vol. 6357 II, .

Biographies

Omer Faruk Eker is a PhD student in School of Applied Sciences and works as researcher at IVHM Centre, Cranfield University, UK. He received his B.Sc. degree in Mathematics from Marmara University and M.Sc. in Computer Engineering from Fatih University, Istanbul, Turkey. He got involved in a project funded by TUBITAK, and Turkish State Railways. His research interests include failure diagnostics and prognostics, condition based maintenance, pattern recognition and data mining.

Dr. Fatih Camci works as a faculty at IVHM Centre at Cranfield University since 2010. He has worked on many research projects related to Prognostics Health Management (PHM) in USA, Turkey, and UK. His PhD work was supported by National Science Foundation in USA and Ford Motor Company on development of novelty detection, diagnostics, and prognostics methods. He worked as senior researcher at Impact Technologies, world-leading SME on PHM, for two years. He has involved in many projects funded by US Navy and US Air Force Research Lab. These projects involve development of maintenance planning and logistics with PHM. He then worked as Asst. Prof. in Turkey. He has led a research project, funded by TUBITAK (The Scientific and Technological Research Council of Turkey) and Turkish State Railways, on development of prognostics and maintenance planning systems on railway switches. In addition to PHM, his research interest involves decision support systems and energy.

Ian Jennions Ian’s career spans over 30 years, working mostly for a variety of gas turbine companies. He has a Mechanical Engineering degree and a PhD in CFD both from Imperial College, London. He has worked for Rolls-Royce (twice), General Electric and Alstom in a number of technical roles, gaining experience in aerodynamics, heat transfer, fluid systems, mechanical design, combustion, services and IVHM. He moved to Cranfield in July 2008 as Professor and Director of the newly formed IVHM Centre. The Centre is funded by a number of industrial companies, including Boeing, BAe Systems, Rolls-Royce, Thales, Meggitt, MOD and Alstom Transport. He has led the development and growth of the Centre, in research and education, over the last three years. The Centre offers a short course in IVHM and the world’s first IVHM MSc, begun in 2011.

Ian is on the editorial Board for the International Journal of Condition Monitoring, a Director of the PHM Society, contributing member of the SAE IVHM Steering Group and HM-1 IVHM committee, a Fellow of IMechE, RAeS and ASME. He is the editor of the recent SAE book: IVHM – Perspectives on an Emerging Field.

Post Implementation CBM Benefit Analysis – U. S. Army AH-64D Apache Helicopter Main Transmission Accessory Sprag Clutch Endurance Project

Matthew C. Carter

U. S. Army Aviation & Missile Command Logistics Center, Supportability and Sustainment Directorate, Sustainment Optimization & Analysis – Analysis Division A-Team, Redstone Arsenal 35808 AL,USA – Direct Army Contractor Support and Employee Owner at Modern Technology Solutions, Inc.

matt.carter@mtsi-va.com

ABSTRACT

Declining defense budgets have required the U. S. Military to develop innovative ways to cut the cost of readiness. Avoiding costs within operations and support activities is achievable by reducing the frequency of maintenance events. This must be done without sacrificing the safety of personnel, pose no harmful effect on aircraft airworthiness, or decline mission readiness. Condition Based Maintenance technology has long been presented as a major enabler for realizing cost avoidance. For more than a decade, the Department of Defense has steadily become more interested and involved in the development, installation and utilization of Condition Based Maintenance technology. The dedicated utilization of the technology has, in several instances, been instrumental in lowering the cost of operational readiness and has reduced soldiers' maintenance burden. The Army's employment of the technology, particularly by the Apache Attack Helicopter Project Management Office and AH-64 aviation units, has supported the Army Aviation and Missile Command Logistics Center in its daily mission to achieve the goal of Cost-Wise Readiness. As an example, the Army is benefiting from an ongoing project that has extended the Apache AH-64D Main Transmission time between overhaul and its Sprag Clutch retirement change life limit. Benefit metrics from flight hour extensions have been calculated and validated. This paper presents the substantiated positive results achieved from one ongoing project and the dedicated utilization of deployed Condition Based Maintenance technology. In terms of cost-avoidance, a 9.2:1 return on investment from the project has occurred, within 24 months.

1. INTRODUCTION

Declining defense budgets have required the U. S. Military to develop innovative ways to cut the cost of readiness. Avoiding costs within operations and support (O&S) activities is achievable by reducing the frequency of maintenance events. This must be done without sacrificing the safety of personnel, pose no harmful effect on aircraft airworthiness, or decline mission readiness. Condition Based Maintenance (CBM) technology has long been presented as a major catalyst for realizing cost avoidance. For more than a decade, the Department of Defense (DOD) has steadily become more interested and involved in the development, installation and utilization of CBM. The dedicated utilization of the technology has, in several instances, been instrumental in lowering the cost of operational readiness and has reduced soldiers' maintenance burden.

The Army's use of CBM technology, chiefly by the Apache Attack Helicopter (AAH) Project Management Office (PMO) and its AH-64 aviation units, has supported the goal of Cost-Wise Readiness (CWR). CWR is a top objective of the Army Aviation & Missile Command (AMCOM) Logistics Center (ALC) in its mission. One goal of the Supportability and Sustainment Directorate (SSD), Sustainment Optimization & Analysis (SOA) – Analysis Division is to support this mission by substantiating CWR wins. As an example, the team collaborated with other U. S. Army Aviation offices to substantiate the benefits from an ongoing CBM project, one which has successfully extended the AH-64D Main Transmission time between overhaul (TBO) and its internal Sprag Clutches' retirement change (RC) Life limit. CBM benefit metrics from these flight hour extensions have been identified, calculated, and validated.

2. BACKGROUND – THE ISSUE BEHIND THE COST DRIVER

In the fall of 1999, all 743 U. S. Army Apache Helicopters were grounded in order to inspect the Main Transmission Accessory Gearboxes. This decision followed the June 1999 accident of an Israeli Air Force AH-64, where it was determined that the accident was caused by the dual failure of the primary and secondary Accessory Gearbox Sprag Clutches. U. S. Army AH-64s had experienced six similar

M. Carter. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

failures. Fortunately, each failure occurred while the aircraft weren't flying. In all cases, the clutches experienced unexpected early wear and caused untimely clutch failure. Gearboxes, including the clutch, with greater than 1000 flight hours were immediately replaced. Fig. 1 shows the internal workings of the AH-64D Main Transmission. Gears shown in the forefront of the illustration are the accessory gears driven by the primary and secondary Sprag Clutches.

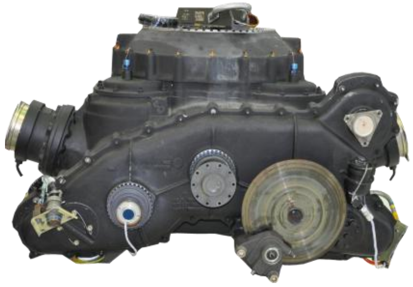


Fig. 1 Apache AH-64D Helicopter Main Transmission

The Sprag Clutches provide the mechanical power input to the helicopter's hydraulic, electrical and pressurized air systems via the accessory gears. Because of the important nature of the clutch's functionality, it is categorized as a Critical Safety Item (CSI) or a flight safety part. Failure of a CSI/flight safety part could cause a catastrophic or critical failure. Such critical failure could result in the loss or serious damage to the aircraft and/or weapons system, an unacceptable risk of personal injury or loss of life, or an uncommanded engine shutdown that jeopardizes safety. The TBO of the transmission, to include its clutches was originally 2000 hours. However, after the entire fleet was grounded for inspection, the clutches were given a highly conservative 1000 hour RC Life limit. This constricted life limit has significantly increased O&S costs for many years. In accordance with Army Regulation 700-82, the clutch is only replaceable at the depot maintenance level. Therefore, the main transmission had to be removed and replaced by Apache units every 1000 hours, at a minimum. In addition, the maintenance burden was increased due to the exceedingly constrictive RC Life limit requirement.

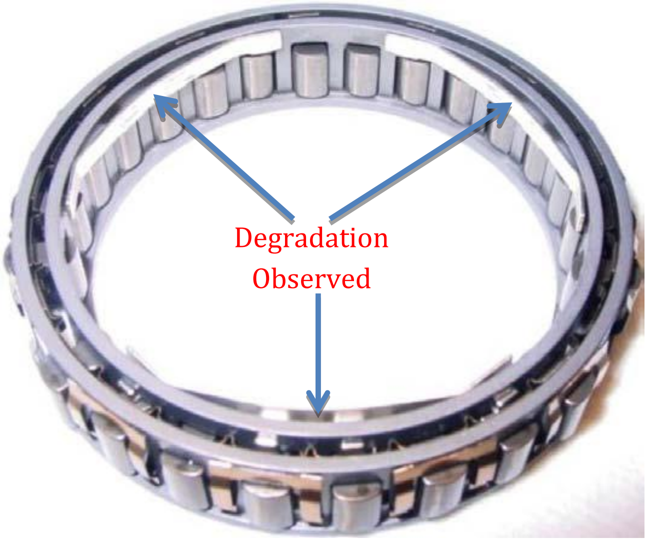


Fig. 2 Transmission Accessory Gearbox Sprag Clutch

3. The O&S Cost Mitigating Remediation Project

The AAH PMO is responsible for planning, directing and controlling the allocation and utilization of all resources authorized for the execution of the Apache AH-64 Program. By authority of the Army Acquisition Executive and the designation of the Program Executive Office (PEO), the AAH PMO performs as the Army's centralized manager for the AAH Project. Within that mission, the PMO directs and controls all phases of research, development, procurement, production, distribution, logistic support, fielding, fiscal and budgetary management of the Apache Attack Helicopter and its sub-products.

This role was exemplified by the AAH PMO's coordination with the AMCOM G3 CBM Office, the Army Aviation and Missile Research Development and Engineering Center (AMRDEC) Aviation Engineering Directorate (AED) CBM office, the Engineering Directorate (ED) Reliability, Availability, Maintainability (RAM) office, the Redstone Test Center (RTC) laboratory, The Boeing Corporation (aircraft OEM), and BorgWarner (Sprag Clutch OEM).

Details of the problem being solved and what approaches were taken: The AAH community at-large suffered from an inflated transmission removal rate due to the unquestionably conservative Sprag Clutch Life Limit. In support of the AAH PMO's number one goal to extend the Sprag Clutch up to 2000 hours which would reflect the Main Transmission's TBO, the AED Apache Division office proposed the Sprag Clutch Endurance Test Project. Funding for the project was approved and provided by the AMCOM G3 CBM Office during FY10. Well-coordinated engineering, testing, and analyses efforts were executed. The approval for the extension was limited exclusively to those AH-64D Helicopters which actively participate in

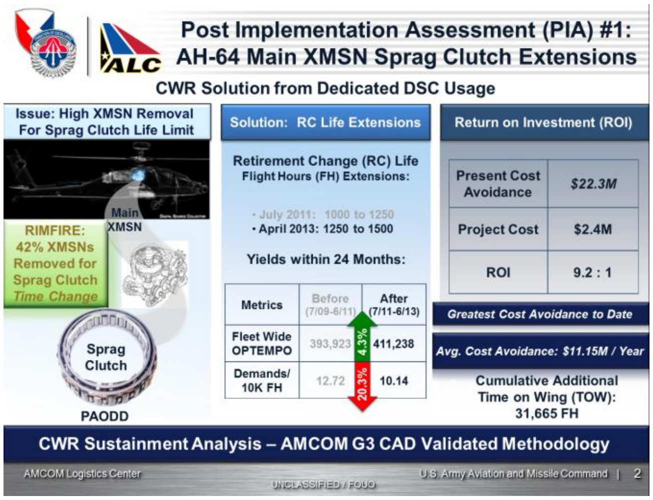
CBM using the MSPU. Such participation is defined in AED's AH-64D CBM Manual. The effort was led by the AED CBM Office and the cost, schedule and performance of the work was overseen by the AMCOM G3 CBM Office. The SSD SOA Office continues its oversight of the project. It has assumed the responsibility for CBM since AMCOM executed a recent reorganization.

4. Results of the Sprag Clutch Endurance Project

The excessively conservative RC Life limit cost driver was significantly mitigated through remediation. Several Army Aviation organizations' collaborative engineering, testing, and analyses efforts resulted in the authorization of the first Sprag Clutch RC Life limit 250 hour extension from 1000 to 1250 hours in July 2011. A second 250 hour extension was authorized in April 2013, from 1250 to 1500 hours. Testing and analysis continues. The goal is to raise the Sprag Clutch RC Life limit to 2000 hours or beyond if possible with no negative effect on airworthiness. Proprietary algorithms are constantly being developed and improved to meet the goal.

Results of this project have been measured and it was determined to be successful. Success has been indicated in terms of avoiding unnecessary cost, exceptional reductions in transmission demands, increased component time on wing, and has resulted in the reduction of soldier-maintainer burden. Greater than $11.14M per year in Cost Avoidance has been calculated during the brief timeframe between July 2011 and June 2013. This has occurred due to extensive development, installation and dedicated utilization of CBM technology. The technology has provided the capability to implement the initial 250 hour extension, without proper utilization, the same outstanding level of success would not have been possible.

The total cost avoidance during this short 24 month timeframe is $22.3M. This figure has been validated by the Army's AMCOM G3 Command Analysis Directorate (CAD) Office. During the same timeframe, unit level demands for the transmission have been reduced by 20.3%, or by 84 total transmissions – from 501 to 417. Therefore, the part was removed and replaced by Apache units much less frequently since the fleet-wide 250 hour extension was authorized. Demand per 10,000 flight hours dropped from 12.72 to 10.14. The AH-64D Main Transmission Time on Wing (TOW) has increased by 31,665 flight hours. This TOW metric was figured by calculating the total number of hours the Sprag Clutch exceeded its former 1000 flight hour RC Life limitation, without exceeding the newly authorized limit. It should be noted that this occurred with a 4.3% increase in total fleet wide flying hours. This increase, without the extension, would have undoubtedly caused greater demands. Figure 3 illustrates the benefits.



Post Implementation Assessment (PIA) #1:
AH-64 Main XMSN Sprag Clutch Extensions

CWR Solution from Dedicated DSC Usage

Issue: High XMSN Removal For Sprag Clutch Life Limit

Main
XMSN
Sprag
Clutch
PAODD

RIMFIRE: 42% XMSNs Removed for Sprag Clutch *Time Change*

Solution: RC Life Extensions

Retirement Change (RC) Life
Flight Hours (FH) Extensions:

- July 2011: 1000 to 1250
- April 2013: 1250 to 1500

Yields within 24 Months:

Metrics	Before (7/09-6/11)	After (7/11-6/13)
Fleet Wide OPTEMPO	393,923	4.3% 411,238
Demands/ 10K FH	12.72	20.3% 10.14

Return on Investment (ROI)

Present Cost Avoidance	*$22.3M*
Project Cost	$2.4M
ROI	9.2 : 1

Greatest Cost Avoidance to Date

Avg. Cost Avoidance: $11.15M / Year

Cumulative Additional Time on Wing (TOW): 31,665 FH

CWR Sustainment Analysis – AMCOM G3 CAD Validated Methodology

AMCOM Logistics Center U.S. Army Aviation and Missile Command | 2

UNCLASSIFIED / FOUO

Fig. 3 Transmission Accessory Gearbox Sprag Clutch

Increased TOW has reduced soldier-maintainer burden in terms of Maintenance Man-Hours (MMH). A conservative estimate of MMH reduction since the first fleet wide AWR was issued in July 2011 is 5,355 hours. It is difficult to put a precise figure on the MMH reduction because of the different situations that the aircraft could be in while the transmission is changed due to Sprag Clutch RC time. Most times when the transmission is replaced the helicopter has been inducted for regularly scheduled Phase Maintenance. This helps reduce the MMH requirement because other parts such as the blades, rotor head, left and right nose gearboxes and all other associated parts that must be removed to allow transmission removal will have already been removed as part of the scheduled maintenance event. The significant quantity of MMHs required to remove and replace the other parts wouldn't count against changing the transmission alone. The transmission in this situation requires 51 MMHs on average, and will vary depending on the experience of the crew and available equipment. The costly Maintenance Test Flight (MTF) required when any of the above parts listed are removed and replaced will not count solely against the transmission replacement. It is understood that an MTF cost approximately $4,000 per total aircraft flight hour.

However, when the aircraft is not already inducted in such scheduled maintenance and all the additional parts must be removed to replace the transmission, the quantity of MMHs and the MTF cost are all counted against the replacement.

It is notable that fleet-wide AH-64D flight hours have increased since July 2011, when the first extension was authorized. This serves to indicate that the project has likely provided benefits to some degree in terms of increases in aircraft uptime, resulting in improved AAH Fully Mission Capable (FMC) rates. The project included engineering, construction and qualification of the Main Transmission test stand located at the RTC. As a result of rigorous testing which included thousands of ground-air-ground cycles, multiple teardowns and detailed analysis, the Apache PMO's Technical Management Division and Fleet Management offices have received authorization from AED to release Accessory Sprag Clutch RC Life and Main Transmission TBO extensions. The extensions have been authorized by publishing Airworthiness Releases (AWR). Each AWR includes within its scope only the AH-64D Helicopters which actively participate in CBM using the MSPU. Such participation is defined in AED's AH-64D CBM Manual.

Army Aviation has been proactive in its establishment of the infrastructure required to achieve these sought after CBM benefits. As a maintenance approach, CBM has begun to enable the optimization of certain DoD maintenance programs. The extensions were engineered by the Aviation Engineering Directorate (AED) in accordance with the Aeronautical Design Standard Handbook for U.S. Army Aircraft CBM Systems. AED authorized a series of five extensions five Airworthiness Releases (AWR). The clutch RC Life has been extended from 1000 to 1500 hours fleet wide and transmission TBO has increased from 2000 to 3000 hours.

5. Conclusion

The AH-64D Main Transmission Sprag Clutch Endurance Test Project has effectively resulted in substantial increases in component Time on Wing, cost Avoidance, demand reductions, and overall increases in the rate of FMC AH-64 Apache Attack Helicopters. It is anticipated that the clutch's CBM enabled benefits will continue to accrue as time passes by and flying hours continue to accumulate fleet wide.

References

Butcher, S. W. (2000), *Assessment of Condition-Based Maintenance in the Department of Defense*. McLean, VA: Logistics Management Institute

U.S. Army Aviation Engineering Directorate (AED), (2013). *ADS-79D-HDBK.* Handbook

http://www.fas.org/man/dod-101/sys/ac/docs/r19991112apache2.htm

day2_sessionviii_01_nrel_keller.pdf

Army Regulation, 700-82, Subject: Joint Regulation Governing the Use and Application of Uniform Source Maintenance and Recoverability Codes, 10 October 2007.

https://www.peoavn.army.mil/SitePages/Apache.aspx

Acknowledgement

The research was supported by the ALC Item Management office, AMRDEC AED & ED offices, AMCOM CAD offices, PEO-A office, and AAH Fleet Management offices.

Nomenclature

AAH	Apache Attack Helicopter
AED	Aviation Engineering Directorate
ALC	AMCOM Logistics Center
AMCOM	Army Aviation & Missile Command
AMRDEC	Army Aviation and Missile Research Development and Engineering Center
AWR	Airworthiness Release
CAD	Command Analysis Directorate
CBM	Condition Based Maintenance
CSI	Critical Safety Item
CWR	Cost-Wise Readiness
DOD	Department of Defense
ED	Engineering Directorate
EE	Electrical Engineer
FMC	Fully Mission Capable
MTF	Maintenance Test Flight
MMH	Maintenance Man-Hours
MTSI	Modern Technology Solutions, Inc.
O&S	Operations and Support
PEO-A	Program Executive Office – Aviation
PMO	Project Management Office
RC	Retirement Change
RAM	Reliability, Availability, Maintainability
RTC	Redstone Test Center
SAC	Sikorsky Aircraft Corporation
SETA	Systems Engineering and Technical Assistance
SOA	Sustainment Optimization & Analysis
SSD	Supportability and Sustainment Directorate
TBO	Time between Overhaul
TOW	Time on Wing

Biography

Matthew Carter has been actively involved in a range of engineering disciplines and Special Operations aspects of Army and Naval Aviation for greater than 22 years. He has worked as a Senior Electrical Engineer (EE) at the Sikorsky Aircraft Corporation (SAC), as Senior Avionics Systems Integrator and Test Lead at L-3 Communications, Inc. and Raytheon, and experienced as Senior Systems Engineer at the Camber Corporation in support of AED – Cargo Helicopter Division. In 2011, Mr. Carter received a Masters Degree in Aeronautical Science in Management. Mr. Carter provides direct professional contractor support to the ALC SSD SOA Analysis Division as a Senior Systems Engineering and Technical Assistance (SETA). He enjoys being a very highly regarded employee-owner at Modern Technology Solutions, Inc. (MTSI). MTSI's headquarters are located in Alexandria, VA. This year the company is celebrating its 20th Anniversary since it was co-founded by current Chief Executive Officer (CEO) Mr. Philip Soucy, retired SR-71 Reconnaissance Systems Officer (RSO).

Bringing the Aircraft System Health Management Tool to Life Using the Informatica Tool Suite

Glenn Peters[1], Mark Mosher[2], and Chris Carlson[3]

[1]*UTC Aerospace Systems, San Diego, CA, 92123, USA*
glenn.peters@hs.utc.com

[2]*Impact Technologies, Rochester, NY, 14623, USA*
mark.mosher@impact-tek.com

[3]*Informatica Corporation, Redwood City, CA, 94063, USA*
ccarlson@informatica.com

ABSTRACT

Earlier this year, UTC Aerospace Systems introduced the Aircraft System Health Management (ASHM) Tool, a web application that takes in Aircraft Condition Monitoring Function (ACMF) reports for selected subsystems and components of an aircraft platform, parses and processes the reported parameters against thresholds and computes estimated or expected values for some key parameters, and serves the report data and the processed results as part of a fleet view available to airline and maintenance users.

The ASHM application uses Informatica PowerCenter to parse and store incoming report data and Informatica RulePoint to apply alert rules and analytic processing to the report data as it is persisted to the ASHM database. This paper describes how UTC Aerospace Systems has leveraged a commercial off-the-shelf tool suite rather than continue to build custom components for the ASHM architecture, with the goals of achieving a short development cycle, robust transaction processing, and scalability to other aircraft systems and other aircraft platforms.

Instead of building tools from scratch that would need to be reworked as the application scales, a set of scalable tools that suit the task at hand and in the future were selected. The full suite of tools, beyond what has been implemented so far, appears to provide capability to address data integrity, reliability and performance as the application grows.

1. BACKGROUND

UTC Aerospace Systems is a supplier of aircraft systems and power, controls and sensing systems for platforms that include commercial and military aircraft, including helicopters, and international space programs. They are a division of UTC Propulsion and Aerospace Systems, which also includes Pratt & Whitney (engines). Their aircraft systems and services include actuation systems, aero structures, air management systems, interiors, landing gear, propeller systems, and wheels and brakes. Their power controls and sensing systems include electric systems, engine components, engine and control systems, fire protection system, intelligence, surveillance and reconnaissance systems, sensors and integrated systems, and space systems.

Historically, UTC Aerospace Systems has not had a proactive capability in place to predict when a critical or important issue might occur for systems other than the APU. Furthermore, what was previously a once-per-day sample of sensor readings for the APU has been moving rapidly to a real-time, over-the-air paradigm.

2. MOTIVATION

Pratt & Whitney AeroPower (formerly Hamilton Sundstrand Power Systems) has been monitoring the A320 and A380 Fleets of many airline customers for over 15 years. Essential requirements developed during this time were:

- The ability to easily accommodate new versions of aircraft reports.
- An alerting engine that is scalable and that allows for the addition of alert processes against any number of parameters.
- A “modern” web based user interface that allows end users to focus on aircraft within the fleet

Glenn Peters et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Except as noted, all content Copyright © United Technologies Corporation, 2013

Informatica Corporation develops and markets data integration software tools that have allowed UTC Aerospace Systems to:

a) Develop positional parsers to quickly react to different report versions
b) Develop an alerting engine for parameters, events and trends
c) Develop a user interface to display alerts generated by the alerting engine.

3. Event Driven Architecture

3.1. Events

Events represent any change(s) in state throughout an enterprise, from the lowest, narrowest level to the highest and broadest.

Events can be sensor reads, social media postings, location changes, financial transactions, database operations and file arrivals, for example. Events are effective triggers for transitions between states.

Missing events are themselves events because something that was expected to happen did not happen. These special events are nonetheless important because they signal transition to anomaly or exception states. For example, a business process step that is not executed by a particular time, an event not received an expected time after or before another event, or no readings, input or feedback received for a specific period of time.

3.2. Event Processing

Event processing is a solution approach that deals with making sense out of events from one or more sources. Events may be combined with other sources of data to define "situations of interest". This provides automatic monitoring of changes in state, reduced decision latency, consistent application of business rules, self-service, and knowledge capture. Figure 1 shows how event processing flows from input event sources to output actions.

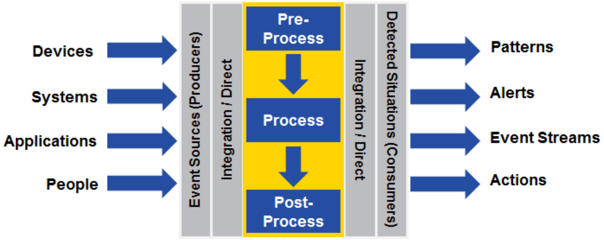


Figure 1. Event Processing Flow

Figure copyright © Informatica Corporation

3.3. Informatica RulePoint

RulePoint is Informatica's event processing solution, and is designed to be deployed as a standalone solution or as part of a broader event-based architecture. For background, RulePoint was developed by Agent Logic, which was acquired by Informatica in 2009, in a move to fill a gap in Informatica's overall data integration suite of products.

RulePoint is focused on the end user / data analyst, and provides a rule-based approach to event processing, and that includes self-service, event-condition-action and temporal/geo-spatial rule handling.

An event-driven architecture is a special type of data driven architecture in which changes in state drive the activity within an environment. Put simply, events drive the execution of logic, or perhaps more correctly, events feed the application of rules and actions based on the outcome of the applications of those rules within the architecture.

4. Application of Informatica Tools to ASHM

Figure 2 shows how the general categories of data, analytics and event processing are handled by the specific Informatica tools that are being used for the ASHM project.

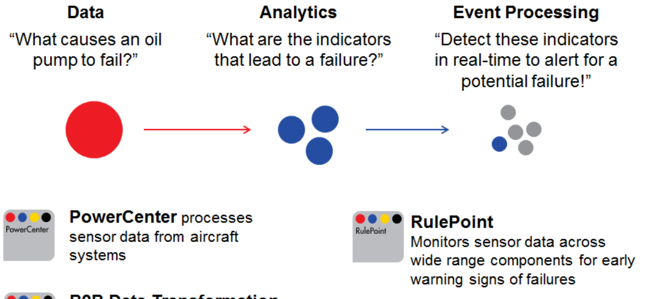

Data
"What causes an oil pump to fail?"
Analytics
"What are the indicators that lead to a failure?"
Event Processing
"Detect these indicators in real-time to alert for a potential failure!"
PowerCenter
PowerCenter processes sensor data from aircraft systems
RulePoint
RulePoint Monitors sensor data across wide range components for early warning signs of failures
B2B Data Transformation
B2B Data Transformation converts different sensor formats into a single data model

Figure copyright © Informatica Corporation

Figure 2. Data Analytics with Event Processing

4.1. High Level Solution Architecture

Figure 3 shows the ASHM architecture at a high level. Standard ATA reports come in for various subsystems of multiple aircraft, are parsed by Informatica PowerCenter using parser templates created for each report type, stored in the application database, acquired and processed by RulePoint for alerts and sent through engineering models for computation of other 'estimated' or 'corrected' parameter values.

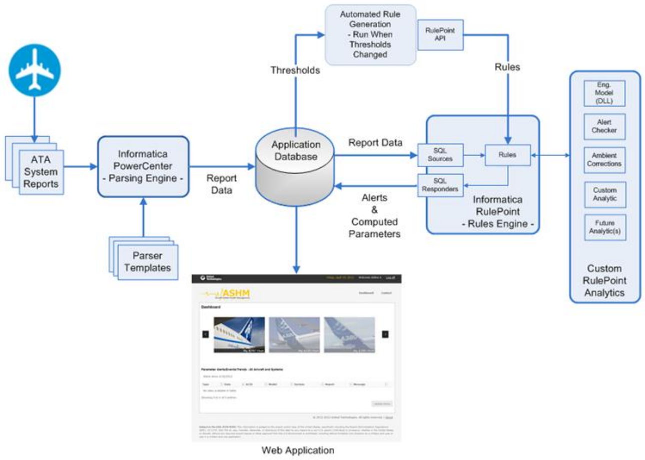


Figure 3. ASHM Architecture

4.2. Data Flow from Aircraft Reports to Database

Figure 4 shows the data flow from incoming aircraft reports to the database. At the top, different report *types* arrive and are bucketed by type into directories for further processing. Each subsystem has one or more report types, and as ASHM grows to process more systems of each aircraft platform, and adds more aircraft platforms, the number of report types will grow accordingly, as indicated by the arrows to the lower right of the diagram.

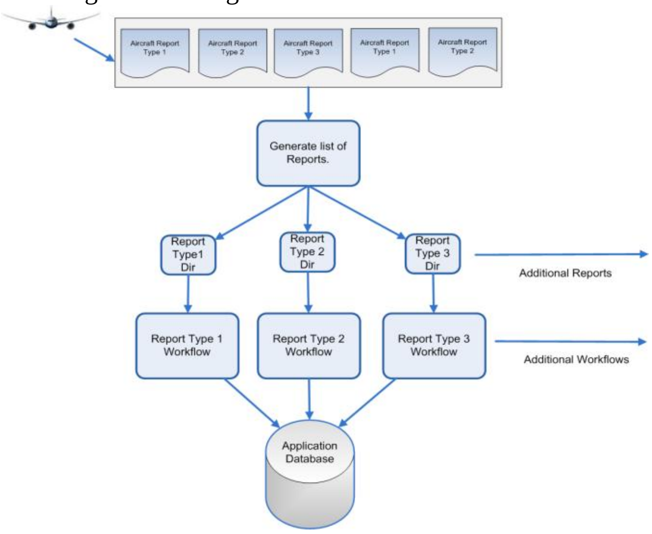


Figure 4. Data Flow – Aircraft Reports to Database

4.3. First Level Parsing to Categorize Reports

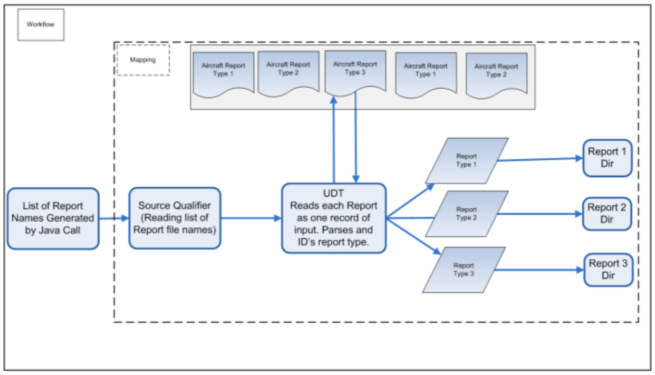


Figure 5. Report Categorization

First, as shown in Figure 5, the reports are parsed at a high level to determine the specific type of report, e.g. subsystem, report type, variant of that report type, and based on that a determination is made and action taken to move that report to the proper staging directory.

4.4. Second Level Parsing to Harvest Report Data

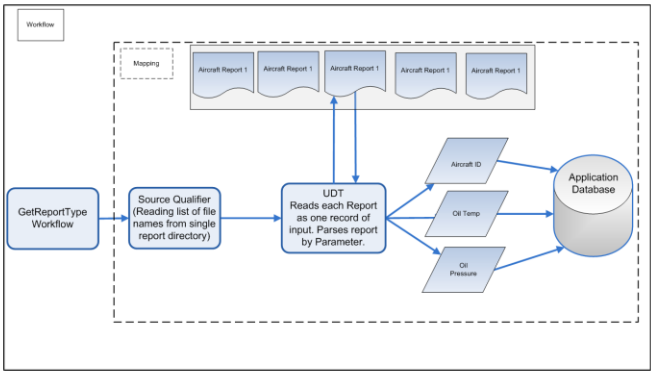


Figure 6. Report Data Transformation

Secondly, as shown in Figure 6, the Data Transformation agent reads each report as one record of input, and parses the parameters from that report. It stores each parameter as part a unique record for that report in the application database.

4.5. RulePoint Workflow in ASHM

Figure 7 shows the flow of data through RulePoint, by report type originating with a *SQL Source* that acquires parameter instance data from the ASHM database and pushes it into a RulePoint *Topic*. A *Rule* references one or more topics and may use data from those topics to 1) determine anomalous conditions, e.g. value out of range, 2) compute new values based on those parameters, 3) send those computed values or detected conditions to a *Responder* that is responsible for storing new data back to the same ASHM database. RulePoint has a wide variety of responders that can send emails, send data to other systems, write to files, et cetera.

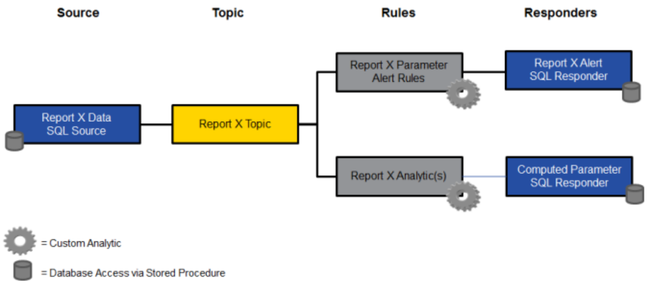

Source
Topic
Rules
Responders
Report X Data SQL Source
Report X Topic
Report X Parameter Alert Rules
Report X Alert SQL Responder
Report X Analytic(s)
Computed Parameter SQL Responder
= Custom Analytic
= Database Access via Stored Procedure

Figure 7. RulePoint Workflow

4.6. Automated Rule Generation

For the ASHM project automatic alert rule generation based on thresholds defined in the database was employed (see Figure 8). This tool uses the RulePoint Java API Adapter to 1) connect to the development RulePoint instance, 2) remove all previously generated (as opposed to hand entered) rules, and 3) generate a new set of rules based on those thresholds. Currently there are 250+ alert rules, with more being added for each new report type. These thresholds are originally defined but are inspected and assessed at the outset and as needed to reduce the occurrence of false alarms.

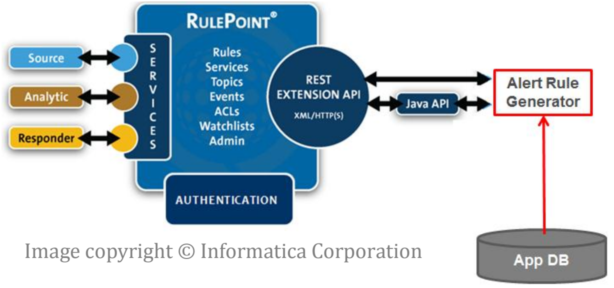


Image copyright © Informatica Corporation

Figure 8. Alert Rule Generation

4.7. Parameter Alerts

The ASHM application checks for out of range "alert" conditions on selected incoming report parameters, looking for warning or alarm conditions that are higher or lower than expected under normal operating conditions. Each "alertable" parameter has its own set of thresholds defined in the database for low and high warning and alarms, for example 0, 1, 99, and 100, for low alarm, low warning, high warning and high alarm thresholds, which would trigger a low alarm condition if the parameter value is at or below 0, for a low warning if the parameter values is above 0 but at or below 1, and likewise a high alarm if the parameter value is at or above 100, and a high warn if the parameter values is at or above 99 but less than 100.

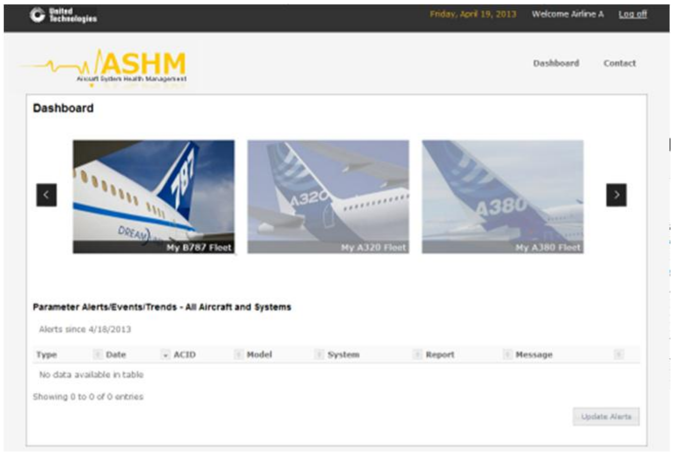
There are also mechanisms in place to define two additional criteria which are when the thresholds are to be ignored, say when some (the same or another) parameter's value meets a certain conditional relationship with a fixed value, e.g. <= some value, = some value, or >= some value.

The parameter alert rules store alert conditions that are detected back to the database, where they are used to display those anomalous conditions to the end user in the web application.

4.8. Analytics

In RulePoint, an *analytic* is a plugin that can be invoked from a rule, and in ASHM a *health (or diagnostic) analytic* is an engineering model of a component or corrections for ambient conditions applied to a set of input parameters. These analytics are used to compute values that would be expected from a normal running system, and these values can be compared against actual values, and also can be processed against thresholds defined in the database, if present.

4.9. Web Application



Friday, April 19, 2013 Welcome Airline A

Dashboard

Parameter Alerts/Events/Trends - All Aircraft and Systems

Alerts since 4/18/2013

Type	Date	ACID	Model	System	Report	Message
No data available in table						

Showing 0 to 0 of 0 entries

Figure 9. ASHM Dashboard

5. Lessons Learned and Best Practices

- By design, the overall application architecture and implementation is heavily dependent on the database architecture and support. This dependence makes it possible to make minor changes that support end-to-end Database to RulePoint to Database functional testing, which are critical to the integrity and proper operation of the tool.

- The integration of the general commercial off-the-shelf tools purpose-built for applications such as ASHM are viewed critical to its continued success. There are other tools in the vendor offering that can be brought in as needed when the platform requires it as it expands in systems, numbers of reports and types of aircraft that are supported.

- Monitoring and logging of the various automation components was and continues to be important for troubleshooting and debugging development and production issues and anomalies.

- As a design goal, the architecture is generic, and so can be modified to accept and process additional types of inputs, e.g. other aircraft such as helicopters, or ground vehicles, or other high value assets.

6. Conclusion

The use of a suite of off-the-shelf commercial tools, whose intended design was consistent with our ASHM design goals, provided the framework for the ASHM architecture. Much of the learning about the application and integration frameworks were encountered and dealt with by the vendor, and that expertise was applied to improve the general purpose set of tools. The tools also provide options for future growth as the application scales to more platforms and systems, as ASHM makes its way towards a ‘big data’ service.

Acknowledgement

The authors would like to thank Rhonda Walthall for her leadership and support on this project, and Rocco LaPorte for his sponsorship of this key UTC Aerospace Systems initiative.

Biographies

Glenn Peters is an IT designer / analyst at UTC Aerospace Systems, and has been involved with the ASHM project since its inception. He played a major role in the initial Airbus report acquisition, parsing and processing, and played a key role in defining the new architecture for the ASHM Tool that resulted in the incorporation of Informatica tools into that event-driven architecture.

Mark Mosher is a project manager at Impact Technologies, a Sikorsky Innovations company that is a world’s leader in the field of complex asset Condition Based Maintenance (“CBM”) and Prognostics & Health Management (“PHM”). Impact provides the most comprehensive range of products and solutions for monitoring, analyzing, detecting, isolating, and predicting the performance, health and readiness of mission critical assets on the market today.

Chris Carlson is the Director of Technology Evangelism for Informatica’s Complex Event Processing business. He joined Informatica with its acquisition of Agent Logic in 2009, where he ran Product Management and new solutions development for the company’s flagship RulePoint complex event processing product. Prior to joining Agent Logic, Chris was Vice President of Product Management at Lucid Security, which was later acquired by Trustwave, a provider of security and PCI DSS compliance management solutions. Prior to that Mr. Carlson held Product Management and Chief Technology positions at network security and application performance management start-ups. He was a Principal Security Consultant at Science Applications International Corporation (SAIC) and held system architecture roles at Booz Allen and Hamilton.

Options for Prognostics Methods: A review of data-driven and physics-based prognostics

Dawn An[1], Nam Ho Kim[2], and Joo-Ho Choi[3]

[1,2]*University of Florida, Gainesville, FL, 32611, USA*
dawnan@ufl.edu
nkim@ufl.edu

[1,3]*Korea Aerospace University, Goyang-si, Gyeonggi-do, 412-791, Korea*
dawnan@ufl.edu
jhchoi@kau.ac.kr

ABSTRACT

Condition-based maintenance is a cost effective maintenance strategy, in which maintenance schedules are predicted based on the results provided from diagnostics and prognostics. Although there are several reviews on diagnostics methods and CBM, a relatively small number of reviews on prognostics are available. Moreover, most of them either provide a simple comparison of different prognostics methods or focus on algorithms rather than interpreting the algorithms in the context of prognostics. The goal of this paper is to provide a practical review of prognostics methods so that beginners in prognostics can select appropriate methods for their field of applications in terms of implementation and prognostics performance. To achieve this goal, this paper introduces not only various prognostics algorithms, but also their attributes and pros and cons using simple examples.

1. INTRODUCTION

Prognostics is to predict future behavior of damage/degradation and the remaining useful life (RUL) of in-service system, which facilitates condition-based maintenance whose schedule is planned according to predicted results based on diagnosis. There are a large number of publications on condition-based maintenance (Jardine, Lin, & Banjevic, 2006; Grall, Bérenguer, & Dieulle, 2002; Yam, Tse, Li, & Tu, 2001) and diagnostics methods (Martin, 1994; Samuel & Pines, 2005; Singh & Al Kazzaz, 2003; Juricic, Znidarsic, & Fussel, 1997; Sugumaran, Sabareesh, & Ramachandran, 2008; Yan & Gao, 2007; Samanta & Al-Balushi, 2003). On the other hand, a relatively small number of reviews on prognostics are available (Si, Wang, Hu, & Zhou, 2011; Srivastava & Das, 2009; Goebel, Saha, & Saxena, 2008; Saha, Goebel, & Christophersen, 2009; Xing, Williard, Tsui, & Pecht, 2011; Zhang & Lee, 2011). Most of them provide a simple comparison of different prognostics methods using a specific application, a summary of pros and cons of algorithms, or a collection of papers. In addition, most review papers focused on algorithms rather than interpreting the algorithms in the context of prognostics. Therefore, the goal of this paper is to provide a practical review of prognostic methods so that beginners in prognostics can select appropriate methods for their field of applications. To achieve this goal, this paper introduces not only various prognostics algorithms, but also their attributes and pros and cons, so that engineers can choose the best algorithm for their field of applications in terms of implementation and prognostics performance.

In general, prognostics methods can be categorized into data-driven, physics-based, and hybrid approaches. Data-driven approaches use information from previously collected data (training data) to identify the characteristic of the currently measured damage state and to predict the future trend. Physics-based approaches assume that a physical model describing the behavior of damage is available, and combine the physical model with measured data to identify model parameters and to predict the future behavior. Hybrid approaches combine the above-mentioned two methods to improve the prediction performance (Mohanty, Teale, Chattopadhyay, Peralta, & Willhauck, 2007; Sankavaram, Pattipati, Kodali, Pattipati, Azam, Kumar, & Pecht, 2009; Cheng & Pecht, 2009; Xu & Xu, 2011; Xing, Miao, Tsui, & Pecht, 2011). The last approach, however, is not mature yet, and will not be considered in this paper.

Dawn An et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data-driven and physics-based approaches have different properties that will contribute to the preference of each algorithm. Providing a standard to select the best algorithm is important for a practical review of prognostic methods, and considering practically available conditions is also important. In this paper, therefore, the following conditions will be considered: the number of data sets, the level of noise and bias in obtained data, availability of loading conditions and physical models, and complexity of damage growth behavior. Typical prognostics algorithms are tested under such conditions, and the algorithms' attributes and pros and cons are provided based on the results.

The paper is organized as follows: in Sections 2 and 3, reviews on the data-driven and physics-based approaches are presented, respectively. In Section 4, several case studies are presented to analyz different prognostic methods, followed by conclusions in Section 5.

2. Reviews on Data-Driven Approaches

Data-driven approaches use information from collected data to identify the characteristics of damage state and predict the future state without using any particular physical model. Instead, mathematical models with weight parameters are employed. The weight parameters are determined based on the training data that are obtained under the various usage conditions. Since the data-driven approaches depend on the trend of data, which often has a distinct characteristic near the end of life, it is powerful in predicting near-future behaviors, especially toward the end of life.

The data-driven approaches are divided into two categories: (1) the artificial intelligence approaches that include neural network (NN) (Chakraborty, Mehrotra, Mohan, & Ranka, 1992; Krogh, 2008; Yao, 1999) and fuzzy logic (Zio & Maio, 2010; Gouriveau, Dragomir, & Zerhouni, 2008), and (2) the statistical approaches that include gamma process (Pandey & Noortwijk, 2004; Dickson & Waters, 1993), hidden Markov model (Rabiner, 1989), and regression-based model such as Gaussian process (GP) regression (Mackay, 1997; Seeger, 2004), relevance vector machine (Tipping, 2001), and least square regression (Tran & Yang, 2009; Bretscher, 1995), etc. Among these algorithms, NN and GP are commonly used for prognostics and will be discussed in the following sections.

2.1. Neural Network (NN)

NN is a representative data-driven method, in which a network model learns a way to produce a desired output, such as the level of degradation or lifespan, reacting to given inputs, such as time and usage conditions. This method mimics the human nervous system, which responds and adapts to a stimulus. Once the network model learns enough the relationship between inputs and output, it can be used for the purpose of diagnosis and prognosis. A typical architecture, feed-forward neural network (FFNN) (Svozil, Kvasnička, & Pospíchal, 1997), is illustrated in Figure 1. In the figure, circles represent nodes (also called a neuron or unit), and each set of nodes in the same column is called a layer. The nodes in the input and output layer, respectively, represent input variables and response variable. The number of nodes in the hidden layer can be adjusted to properly express the mechanism between input and output. Once the network model learned enough, the model is functionalized using transfer functions and weight parameters. Transfer functions characterize the relationship between each layer, and several types of transfer function are available such as sigmoid, inverse, and linear function (Duch & Jankowski, 1999). Weight parameters include weights for the interconnected nodes and biases that are added to inputs of transfer functions (Liu, Saxena, Goebel, Saha, & Wang, 2010; Firth, Lahav, & Somerville, 2003). The process of finding the weight parameters is called training or learning, and to accomplish that, many sets of training data are required.

In general, FFNN is often called a back-propagation neural network (BPNN) because weight parameters are obtained through the learning/optimization algorithm (Rumelhart, Hinton, & Williams, 1986) that adjusts weight parameters through backward propagation of errors between actual output (training data) and the one from the network model based on gradient descent optimization methods. In addition to FFNN, there exists recurrent (Bodén, 2002), fuzzy (Liu & Li, 2004), wavelet (He, Tan, & Sun, 2004), associative-memory (Bicciato, Pandin, Didonè, & Bello, 2001), modular (Happel & Murre, 1994), and hybrid (Psichogios & Ungar, 1992; Rovithakis, Maniadakis, & Zervakis, 2004) neural network.

In the following, three important issues are discussed for NN-based prognostics.

2.1.1. Issue 1: Network Model Definition (the Number of Node and Layer)

The first issue is the definition of the network model that includes selecting the number of hidden nodes, hidden layers and input nodes. Trial-and-error methods are often used to determine a suitable network model. Lawrence, Giles, and Tsoi (1998) and Doukim, Dargham, and Chekima

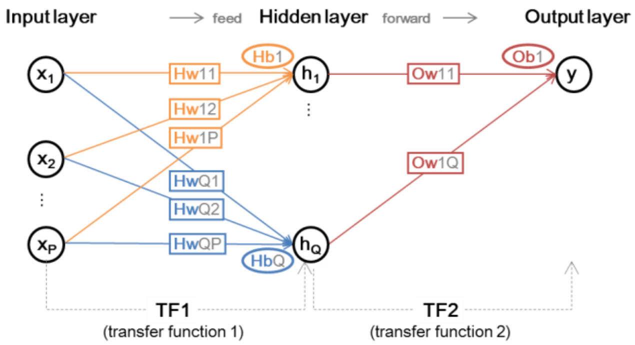


Figure 1. Illustration of typical network model: FFNN

(2010) investigated the usage of mean square error in order to find the optimal number of hidden nodes. Gómez, Franco, and Jérez (2009) used generalization complexity to determine the number of nodes and showed that the results were quite close to the optimum. Although one or two hidden layers are generally used, there is no fixed rule. Ostafe (2005) presented a method using pattern recognition to determine the number of hidden layers. The problem of determining the number of input nodes occurs when input variables affecting the output are not clear; various inputs possibly affecting the response can be considered or only data trace is applicable. Chakraborty et al. (1992) compared the prediction results of flour prices using variable network models. In such a case, actual past values are used for inputs, but how many past values should be used is unclear. In fact, the accuracy of prediction results is not proportional to the number of input nodes. Chang and Hsieh (2011) also researched to select the optimal input layer neurons using particle swarm optimization. Therefore, defining a proper neural network model can be difficult for new users without having much experience.

2.1.2. Issue 2: Optimal Parameters (Finding Weights and Biases)

Once a network model is defined, the next issue is to find weight parameters related with the model. In NN, no matter how complex the relationship between input and output layer is, it is possible to express the relationship by augmenting the number of hidden layers and hidden nodes. However, when the BP algorithm is used, the following problems exist: (1) the global optimum of many weight parameters is extremely difficult to find, and (2) the convergence rate is very low and depends on the initial estimates. For these reasons, there have been many efforts to improve the drawbacks of the BP algorithm. Salomon and Hemmen (1996) presented a dynamic self-adaptation algorithm to accelerate the steepest descent method, and Chen, Lin, Tseng, and Lin (2006) applied the simulated annealing algorithm to search for the best BP parameters such as learning rate, momentum and the number of hidden nodes. Also, Subudhi, Jena, and Gupta (2008) proposed a technique combining the genetic algorithm and differential evolution with BP, and Nawi, Ransing, and Ransing (2007) presented a technique combining the conjugate gradient optimization algorithm with the BP algorithm. There are many ensemble techniques to improve the performance of a single algorithm (Navone, Granitto, Verdes, & Ceccatto, 2001; Jacobs, 1995; Drucker, Cortes, Jackel, LeCun, & Vapnik, 1994; Krogh & Vedelsby, 1995; Perrone & Cooper, 1993; Naftaly, Intrator, & Horn, 1997), and the other efforts are found in the Refs.(Jardine et al., 2006; Salomon and Hemmen, 1996; Nawi et al., 2007) However, finding good weight parameters is still challenging, and the performance of NN algorithm deteriorates with non-optimal weight parameters.

2.1.3. Issue 3: Uncertainty From Data and Optimization Process

Last but not least, uncertainty in noise and bias in training data is an important issue in NN, as most measured data include them. The bias here is different from the bias as weight parameters; here the bias is the error caused by sensors, such as calibration error. In terms of noise, it is common to provide confidence bounds based on nonlinear regression and/or the error between NN outputs and training data (Chryssoloiuris, Lee, & Ramsey, 1996; Veaux, Schumi, Schweinsberg, & Ungar, 1998; Yang, Kavli, Carlin, Clausen, & Groot, 2000; Leonard, Kramer, & Ungar, 1992). Bootstrapping (Efron & Tibshirani, 1994) can also be applied, which can be easily implemented by running Matlab NN toolbox several times because Matlab uses different subsets of given training data for obtaining weight parameters. Furthermore, running NN several times can relieve the concerns about initial weight parameters for optimization by setting different initial parameters automatically. For example, Liu et al. (2010) used the repeating method with 50 attempts to predict battery's RUL with uncertainty. Actually, a basic method to handle uncertainty in NN is the probabilistic neural network (PNN) (Specht, 1990) using Parzen estimator (Parzen, 1962). However, most papers employ PNN for classification or risk diagnosis (Petalas, Spyridonos, Glotsos, Cavouras, Ravazoula, & Nikiforidis, 2003; Giurgiutiu, 2002; Mao, Tan, & Ser, 2000), and prognostics ones are rarely found except for the study by Khawaja, Vachtsevanos, and Wu (2005). They introduced a way to obtain not only confidence bounds but also confidence distribution based on PNN to predict a crack on a planetary gear plate. Unfortunately, bias in measured data cannot be handled with data-driven approaches because the approaches are based on measured data, and there are no parameters related with bias.

2.2. Gaussian Process (GP) Regression

GP is a commonly used method among regression-based data-driven approaches for prognostics, whose conceptual property is illustrated in Figure 2. An outstanding property of GP is that simulated outputs are smoothly constructed making exactly the same value as every measured point (data) as blue-dashed curve in Figure 2. The reason for this can be explained with following GP model that is composed of a global model ($\mathbf{f}\boldsymbol{\beta}$) and departures ($\mathbf{r}\mathbf{R}^{-1}\left(\mathbf{Y}-\mathbf{F}\boldsymbol{\beta}\right)$):

$$y^{*}=\mathbf{f}\boldsymbol{\beta}+\mathbf{r}\mathbf{R}^{-1}\left(\mathbf{Y}-\mathbf{F}\boldsymbol{\beta}\right) \qquad (1)$$

where y^{*} is a simulated GP output at an arbitrary input vector, $\mathbf{x}^{*}$ whose size is $1\times p$ (p is the number of input variables), $\mathbf{f}$ is the known function of $\mathbf{x}^{*}$ and determines polynomial order of global model, $\boldsymbol{\beta}$ is the regression coefficient and obtained by $\left(\mathbf{F}^{T}\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}^{T}\mathbf{R}^{-1}\mathbf{Y}$, and $\mathbf{r}$ is a

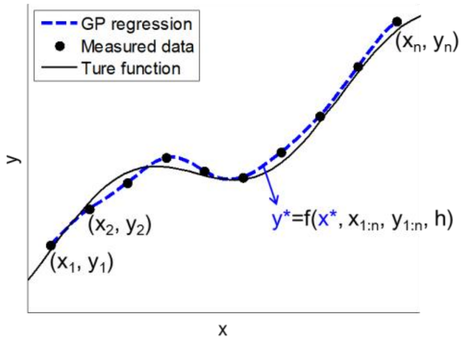


Figure 2. Illustration of GP regression

$1\times n$ (n is the number of measured data) vector, which represents a correlation between $\mathbf{x}^*$ and a $n\times p$ input matrix of all measured points, $\mathbf{X}$. The rest capital letters $\mathbf{R}$, $\mathbf{Y}$, and $\mathbf{F}$ that are the same property as their small letters' have the same size in terms of row vector as their small letters, but the size of column vector is the same as number of measured data, n.

If correlation terms ($\mathbf{r}$ and $\mathbf{R}$) and departures term are ignored in Eq. (1), it becomes that $y^*=\mathbf{f}\boldsymbol{\beta}$ with $\boldsymbol{\beta}=\left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{Y}$, which is the ordinary least square regression (Tran & Yang, 2009). Consequently, GP is distinguishable from least squarie in terms of that simulated outputs penetrate every measured point with two assumptions that (1) GP model is a combination of global model and local departures, and (2) the error between every points is correlated. Assumption (1) is already reflected in the GP model, and assumption (2) is determined by a type of covariance function and scale parameters (or hyperparameters) related with them. Once scale parameters are obtained based on the measured points (training data) using optimization algorithm, GP model can be used to predict future behavior of damage. Lastly, the name, Gaussian process comes from the assumption that each point is normally distributed. Based on this assumption with multivariate normal distribution, more mathematical/probabilistic derivation can be done in terms of probabilistic parameters of Gaussian distribution; the mean expressed in Eq. (1) and the variance found in the Refs. (Gelman, Carlin, Stern, & Rubin, 2004; Santner, Williams, & Notz, 2003).

2.2.1. Issue 1: Model Problem (Covariance Function)

In common with NN, the performance of GP largely depends on models. In this case, covariance function and polynomial order of global model are related with GP model, but order of global model is less important as it is often handled with constant value. There are various types of covariance functions such as radial basis (or squared exponential), rational quadratic, neural network, Matern, periodic, constant and linear (Rasmussen & Williams, 2006; Williams, 1997). Mohanty, Das, Chattopadhyay, and Peralta (2009) compared the prediction results of crack length under variable loading from radial basis function (RBF) covariance function and neural network based (NN-based) covariance function, and showed that RBF-based GP model outperformed NN-based one in their application. As part of an effort to resolve the selection of covariance function, research on nonstationary covariance functions that is a model to adapt to variable smoothness and can be constructed by adding or multiplying simple covariance functions has been conducted. Paciorek and Schervish (2004) introduced a class of nonstationary covariance functions so that the model adapts to variable smoothness, and compared the results from stationary GP. From their research, it was concluded that the results from nonstationary GP are better than stationary GP, but pointed out that simplicity loss of the algorithm occurs as the nonstationary GP requires more parameters than a stationary GP. Brahim-Belhouari and Bermak (2004) used nonstationary GP to predict respiration signal, and compared with a GP model with an exponential covariance function, and Liu, Pang, Zhou, and Peng (2012) used the combination of three covariance functions to predict lithium-ion battery degradation (state of health, SOH).

2.2.2. Issue 2: Optimization Problem (Finding Scale Parameters)

Determining the scale parameters related with covariance function is also important, since they determine the smoothness of regression model. In general, the parameters are obtained based on equivalent likelihood function (Sacks, Welch, Mitchell, & Wynn, 1989) via optimization algorithm. It, however, is a difficult task to search their optimum values, and even if they are found they are not always the best selection (An & Choi, 2012). Since the scale parameters are seriously affected by input and output values, input and output values are applied as normalized values in most cases. Mohanty et al. (2009), however, studied the performance to prediction crack growth according to three different types of scaling. Neal (1998) considered the scale parameters as distributions rather than deterministic values, and An and Choi (2012) showed that the GP models with scale parameters identified as distributions outperform the one using optimal deterministic parameter.

2.2.3. Issue 3: Data Problems (Num. of Data and Uncertainty)

Even though large number of training data is usually profitable for increasing accuracy of prediction results, it's not always acceptable for GP because it also increases computational costs to calculate the inversion of the covariance matrix (Eq. (1)) as well as generates singularity. It is suggested that inversion by direct method may become prohibitive when the number of the data points is greater than 1000 (MacKay, 1997). As a solution to relieve such problem, the methods to select a subset of data points are usually employed (Lawrence, Seeger, & Herbrich, 2003; Smola & Bartlett, 2001; Foster, Waagen, Aijaz, Hurley, Luis, Rinsky, Satyavolu, Way, Gazis, & Srivastava, 2009). While Melkumyan and Ramos (2009) suggested new covariance function based on cosine function that inherently provides a sparse covariance matrix. In terms of uncertainty, it's determined with Gaussian noise as mentioned before. Mohanty et al. (2007) and Liu et al. (2012), respectively, showed the predictive confidence interval of crack length and SOH of lithium-ion battery using GP.

3. Reviews on Physics-Based Approaches

Physics-based approaches combine the physical damage model with measured data to predict future behavior of damage and the RUL, which is illustrated in Figure 3. The behavior of the physical model depends on the model parameters that are estimated and updated in company with damage state based on the measured data. Finally, the RUL is predicted by progressing the damage state until it reaches a threshold as the dashed curves in the Figure 3. Similar issues are addressed in this approach as the previous ones.

3.1. Issue 1: Model Problem (Physical Model Accuracy)

Since the physics-based approaches employ a physics model describing the behavior of damage, it has advantages in predicting long term behaviors of damage. However, model validation should be carried out since such models contain many assumptions and approximations. There have been much literature on model validation using statistical methods such as hypothesis test and Bayesian method to calibrate and improve the model by comparing with observation (Rebba, Huang, Liu, & Mahadevan, 2006; Rebba, Mahadevan, & Huang, 2006; Kleijnen, 1995; Sargent, 2009). In general, the number of model parameters increases as model complexity increases, which makes it difficult to identify the model. Recently, Coppe, Pais, Haftka, and Kim (2012) showed that the issue of model accuracy can be relieved by identifying equivalent model parameters of the simpler model. They showed that a simple Paris model with an assumed stress-intensity factor can be used for predicting crack growth of complex geometries by adjusting the model parameters to compensate for the error in the simple model. Although this is limited to the case of a similar model form, cumbersome efforts to validate the model accuracy can be eased off.

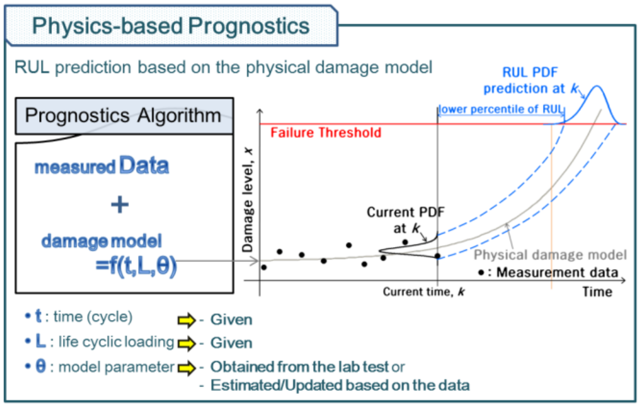


Figure 3. Illustration of physics-based prognostics

3.2. Issue 2: Model Parameter (Physical Model Parameters, Noise and Bias)

3.2.1. Introduction to Physics-Based Algorithms

Once a physical model is available, model parameter identification becomes the most important issue, which is performed with an estimation algorithm based on measured data with a usage condition. In fact, estimation algorithms become criteria to classify physics-based approaches. There are several algorithms such as Kalman filter (Kalman, 1960), extended Kalman filter (Julier & Uhlmann, 2004), particle filter (PF) (Doucet, De Freitas, & Gordon, 2001), and Bayesian method (BM) (Kramer & Sorenson, 1998). These algorithms are based on the Bayesian inference (Bayes, 1763), in which observations are used to estimate and update unknown parameters in the form of a probability density function (PDF). The updated PDF is called the posterior distribution, which is obtained by multiplying the prior distribution that is prior knowledge or information of the unknown parameters and the likelihood function that is the PDF value of measured data conditional on the given parameters.

There are several researches dealing with parameter estimation in terms of prognostics. DeCastro, Tang, Loparo, Goebel, and Vachtsevanos (2009) used extended Kalman filter to estimate model parameters and predict RUL for crack growth on a planetary carrier plate. Orchard and Vachtsevanos (2007) estimated the crack closure effect using PF for RUL prediction of a planetary carrier plate based on vibration-based feature. Daigle and Goebel (2011) used PF to estimate wear coefficients by considering multiple damage mechanisms in centrifugal pumps. An, Choi, Schmitz, and Kim (2011) estimated wear coefficients to predict the joint wear volume of slider-crank mechanism based on BM. Among the aforementioned algorithms, PF is the most commonly used for prognostics. In the following, PF and BM are discussed.

- ***Particle filter (PF)***

PF is the most commonly used algorithm in the prognostics field, in which the posterior distribution of model parameters is expressed as a number of particles (or samples) and their weights as shown in Figure 4. There are three steps in PF process: (1) prediction step - posterior distributions of the model parameters ($\boldsymbol{\theta}$) at the previous ($k-1$th) step are

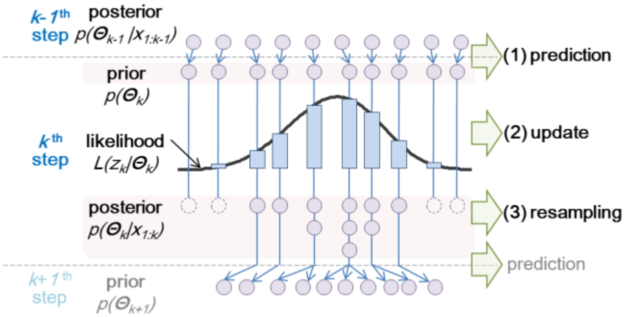


Figure 4. Illustration of PF process

used for the prior at the current (k th) step, and the damage state at the current time is transmitted from the previous one based on a damage model (physical model), (2) update step - model parameters and damage state are updated based on the likelihood function combined with measurement data (x), and (3) resampling step - particles of the prior distribution are resampled based on their weights expressed as vertical-rectangular in Figure 4 by duplicating or eliminating the samples with high or low weight, respectively. The resampled result corresponds to the posterior distribution at the current step and is also used as the prior distribution at the next ($k+1$th) step. That means the Bayesian update is processed in a sequential way with particles in PF.

Since prediction results from PF depends on initial distributions of parameters (posterior distribution at $k=0$), correct information or proper assumption for initial distributions is one of the most important issues to find model parameters. Another important point is accumulated sampling error that occurs during the update process. In other words, the particle depletion problem can occur since those particles with a very small weight are eliminated, while those particles with a high weight are duplicated. There have been researches to recover the particle diversity during update process. A common practice is to add random sample ξ_{k-1} from arbitrary distribution during the prediction step; that is, $\theta_k = \theta_{k-1} + \xi_{k-1}$ so that identical particles are not generated (Higuchi, 1997; Kitagawa, 1987; Wang, Liao, & Xing, 2009). This method, however, can change probabilistic characteristic of parameters as well as increase the variance of parameters. Gilks and Berzuini (2001) proposed a resample-move algorithm based on PF and Markov Chain Monte Carlo (MCMC) method (Andrieu, Freitas, Doucet, & Jordan, 2003), Kim and Park (2011) introduced the maximum entropy particle filter and demonstrated the effectiveness of the proposed technique by applying it to highly nonlinear dynamical systems.

- ***Bayesian method (BM)***

The Bayesian update is processed with an overall way in BM; the posterior distribution is obtained as an equation by multiplying all the likelihood function given by k number of data. Once the posterior distribution is available, a sampling method can be used to draw samples from the posterior distribution. Therefore, the estimation performance in BM depends on sampling methods, MCMC method which has been recognized as a computationally effective means is usually employed. MCMC is based on a Markov chain model of random walk as shown in Figure 5. It starts from generating an arbitrary initial sample (old sample) and a new sample from an arbitrary proposal distribution with the centered at the old sample. The two samples are compared with each other based on an acceptance criterion, from which either one is selected. In Figure 5, two circles with dashed line means new samples not selected, and in this case, the old one is selected. This process is repeated as many as the number of particles in PF.

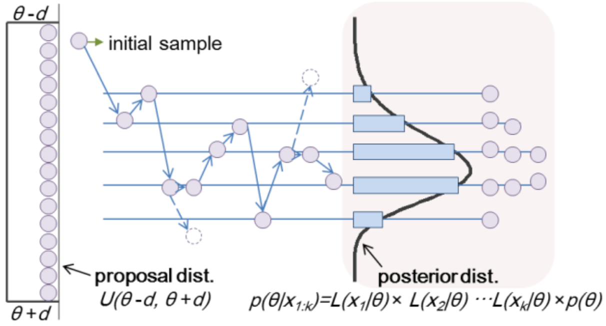


Figure 5. Illustration of BM process with MCMC.

Even though there is no accumulated sampling error in BM, there still exists some error caused by sampling method, random walk. Initial sample, proposal distribution for new sample, and acceptance ratio to the old sample have an effect on the sampling results; with improper setting, it could be not converged or show stationary chain that old sample is selected continually. There are some researches to reduce those effects by utilizing marginal density function for proposal distribution (Rubin, 1998; An & Choi, 2013). Gelfand and Sahu (1994) presented two distinct adaptive strategies to accelerate the convergence of a MCMC algorithm. More literatures are found in the Ref. (Andrieu et al. 2003).

3.2.2. Correlation Issue between Model Parameters

One of the most challenging parts in model parameter identification is correlation between model parameters. Without properly identifying correlation, the predicted RUL can be significantly different from reality. An, Choi, and Kim (2012) studied the correlation in a crack growth problem, in which correlation between the parameters was well identified, but eache parameter was not accurately identified under a large level of noise because of the correlations: correlation between the two Paris parameters and correlation between bias and the initial crack size. The prediction results of damage growth and RUL, however, were reliable since many combinations of the correlated parameters can yield the same prediction results.

3.3. Issue 3: Uncertainty From Data (Noise and Bias)

Since damage cannot be directly measured in many cases, a damage quantification process is required from sensor measurement data, which is called structural health monitoring (SHM). SHM data could include a large level of noise and bias due to sensor equipment, and there are several researches dealt with the analysis of noise and bias in SHM data. Gu, Azarian, and Pecht (2008) presented a prognostics approach which detects the performance degradation of multilayer ceramic capacitors under temperature-humidity-bias conditions. Coppe, Haftka, Kim, and Yuan (2009) showed that the uncertainty in structure-specific damage growth parameters can be progressively reduced in spite of noise and bias in sensor measurements. Guan, Liu, Saxena, Celaya, and Goebel (2009) considered various uncertainties from measurements, modeling, and parameter estimations to describe the stochastic process of fatigue damage accumulation based on a maximum entropy-based general framework. It is concluded that convergence with large noise becomes slow, and positive and negative bias, respectively, effect on early and late prediction.

4. Case Study to Select Pertinent Method

Prognostics algorithms including NN, GP, PF and BM are analyzed and compared, so that engineers can choose the best algorithm for their field of applications. Since there are many variations each algorithm, the most common and basic ones are employed.

4.1. Problem Definition

4.1.1. Given Information for Case Study

Paris model (Paris & Erdogan, 1963) and Huang's model (Huang, Torgeir, & Cui, 2008) are, respectively, employed for a simple behavior of damage growth and complex behavior of damage growth, which are shown in Figure 6. In each model, there are ten sets of data under different loading conditions. Different level of noise and bias are artificially added to the data in Figure 6. Bias is considered as -2 mm, and noise is uniformly distributed between $-u$ mm and $+u$ mm. Three different levels of u are considered: 0, 1, and 5 mm. Ten data sets are numbering, one data set (usually #8) will be used for the set to be predicted and the other sets will be used for training data.

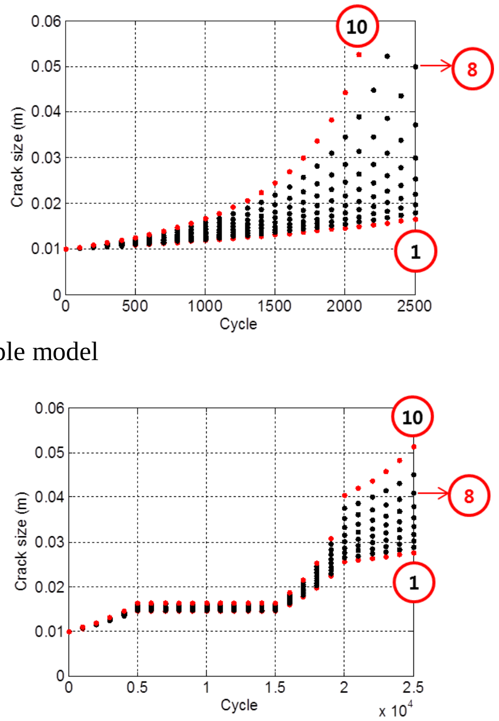
4.1.2. Definition of Algorithm Conditions

For the case of NN, the network model is constructed based on FFNN with three input nodes, one hidden layer with two nodes. Then, the number of total weight parameters become 11 including eight weights ($3\times2+2\times1$) and three biases (2+1). Since there is one hidden layer, two transfer functions are required, and as a common way, the tangent sigmoid and pure linear functions are employed. For GP model, linear or second order polynomial function is employed for the global model, and one parameter radial basis covariance function is employed as follow:

$$R\left(\mathbf{x}^i,\mathbf{x}^j\right)=\exp\left[-\left(d/h\right)^2\right],\ d=\left|\mathbf{x}^i-\mathbf{x}^j\right|,\quad i,j=1,\ldots,n \qquad (2)$$

where h is a scale parameter to be identified, $\mathbf{x}^i,\mathbf{x}^j$ are vector of input variables, n is total number of training data; in this case, $n=k-3$, and k is the current time index.

For the input variable in NN and GP, the previous three damage data (x_{k-3}, x_{k-2}, x_{k-1}) are used, and the current damage data (x_k) becomes the output. Also, loading conditions can be utilized by adding to the input nodes. In this case, loading condition at the current cycle is added to the forth input node. If $k=16$, 13 sets of input and output data are available, which are used to obtain weight parameters, and then future damages ($x^p_{k+1},x^p_{k+2},x^p_{k+3},\ldots$) are predicted based on the obtained parameters and the previous damage data. According to the previous damage data used as inputs, prediction methods can be divided into short term prediction and long term prediction. Short term prediction is one-step ahead prediction since it uses only measured data for input, e.g., x_k, x_{k+1}, x_{k+2} are inputs to predict x^p_{k+3}. On



a) simple model

b) complex model

Figure 6. Problem definition for case study

the other hand, long term prediction is multi-step ahead prediction since it utilizes predicted results as inputs, e.g., x_k, x_{k+1}^p, x_{k+2}^p are inputs to predict x_{k+3}^p . In terms of prediction uncertainty, while noise in training data can be handled during GP, bootstrapping is employed by repeating NN 30 times to handle uncertainty from data noise and optimization process in NN.

Models for physics-based approaches are the same as Paris model and Huang's model employed to generate data sets. There are, respectively, two and six model parameters in Paris model and Huang's model, and the data sets in Figure 6 were generated with following true model parameters:

Paris model ture parameters: m=3.8, $C = 1.5\text{e-}10$

Huang's model true parameters: $m = 3.1$, $C = 5.5\text{e-}11$,

$m = 3.1,\ C = 5.5\text{e-}11,\ \Delta K_{th} = 5.2,\ \beta = 0.2,\ n = 2.8,\ \sigma_y = 580$

4.2. Case Study Results

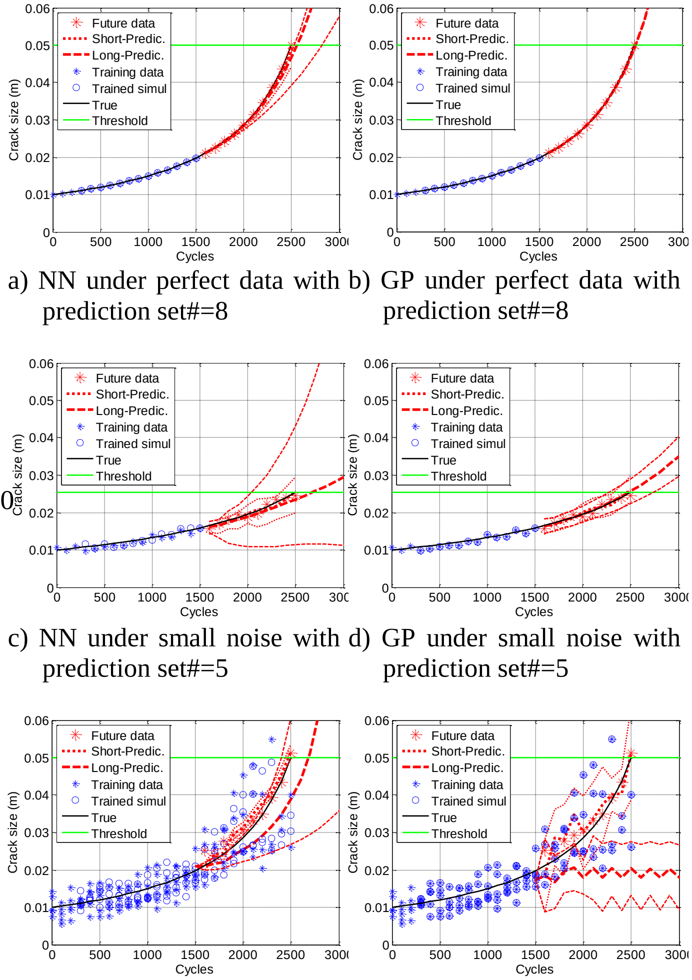
4.2.1. Data-Driven Results

At first, the performance of training data is compared. Figure 7 shows the comparison between NN and GP with different levels of noise. In the figure, the red and blue star markers are, respectively, future damage data and training data up to the current cycle (1500 cycles), and circle markers are simulation results from each algorithm. Thick dotted curve and thick dashed curve are, respectively, medians of short term prediction and long term prediction, and their thin curves represent 90% confidence intervals.

GP show exact result under perfect data (no noise) and outperform NN under small noise in terms of both short term and long term prediction. Long term prediction using GP is not available under large noise and many sets of training data. In this case, NN outperform GP in both short term and long term prediction, and long term prediction results get better as the number of cycles increases. The reason why GP is better than NN for small noise and small number of data is because of correlation property. Large data affect adversely the covariance matrix manipulation. Also, GP model penetrates every training data points assuming each data are correlated, which means that the GP behaves poorly when the data include large noise because the noisy relation is reflected to the predictions. While the reason for better results from NN under large noise and many data is that increasing data have no effect on network model but gives more information. Also, combination of transfer function is much less restricted to the level of noise.

4.2.2. Physics-Based Results

As mentioned before, bias in obtained data cannot be handled with data-driven approaches. In contrast, physics-based one can do it by adding bias to unknown parameter, whose results are shown in Figure 8. In the figure, measurement data up to the current cycle (blue star makers) are biased; measured crack size is consistently less than the true one. The medians of prediction results (dashed curves) at the current cycle are close to the true one, which means bias is well identified and compensated. Since physical model describes behavior of damage data accurately, the difference between model and data can be obtained as a constant at any cycle, which satisfies bias in measurement data is a constant. Further study for noise and bias in physic-based approaches is found in the Ref. (An et al., 2012).



a) NN under perfect data with prediction set#=8

b) GP under perfect data with prediction set#=8

c) NN under small noise with prediction set#=5

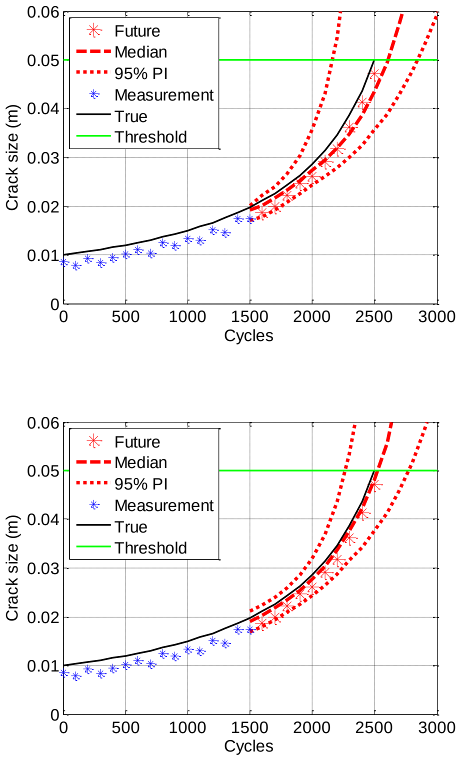
d) GP under small noise with prediction set#=5

e) NN under large noise with prediction set#=8 and training set#=6,7,9,10

f) GP under large noise with prediction set#=8 and training set#=6,7,9,10

Figure 7. Results from data-driven approaches with the case of simple damage growth

The difference between PF and BM is negligible in view of the prediction results because the two methods have the same foundation with the same physical model. The only differences are the way of updating distributions and generating samples. BM is faster than PF because the



a) PF

b) BM

Figure 8. Results from physics-based approaches with the case of simple damage growth under small noise and negative bias

posterior distribution is given as a single equation and there is no accumulated sampling error. Even if PF has an accumulated sampling error during the updating process, it predicts well because the updating process occurs along with damage propagation. However, BM is not practical for them because of tremendous computational costs (Gilks & Berzuini, 2001; An, Choi, & Kim, 2012; Storvik, 2002). This is a key difference between the two methods, and more detailed comparison between PF and OBM can be found in the reference by An et al. (2012).

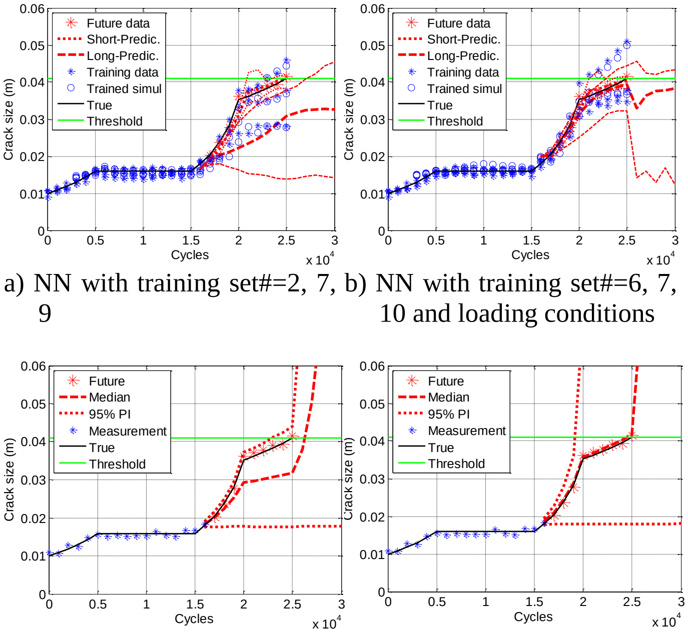
4.2.3. Case Study for Comparison between NN and PF

For data-driven approaches, NN is considered to predict damage growth in a complex model, because it is difficult to use a proper correlation function for GP to predict future due to retardation portion in obtained data. To find out how many data sets are required to obtain proper prediction results, different numbers of training sets are randomly selected. The results from this case study are presented in Figure 9 (a) and (b). Based on the training data sets, NN well predicts future damage as shown in Figure 9 (a). It, however, is limited to short term prediction, and proper results for long term prediction could not be found with different attempts. If there are available loading conditions, medians of short term and long term prediction become similar to each other with at least three sets of training data as shown in Figure 9 (b).

If there are available physical model as well as loading conditions, it might be clear that using physics-based approaches for the case of complex model outperforms data-driven ones. Since BM has a difficulty for the complex model due to extremely expensive computational costs, PF is only considered, and the results are shown in Figure 9 (c). The median of prediction is still not accurate, but uncertainty covers that. Also, since this physical model largely depends on the initial damage, if the initial distribution of damage is also updated, median will close to the true one. For example, Figure 9 (d) shows the results with true value of initial damage.

4.3. Results summary

In terms of algorithms, results from case studies can be summarized as follow: GP works well when the correlation function can be well defined such as the case of small noise data and simple models. It is easy to implement and fast to calculate. NN is proper to apply for the case of large noise and complex models with many training data sets. Even so, NN can be applied for small noise and simple models, which has a wide range of applications. It is, however, challenging to obtain many sets of data in realistic applications. PF and BM are less affected by the level of noise and model complexity, but they can be employed only if a physical model and loading conditions are given. The



a) NN with training set#=2, 7, 9

b) NN with training set#=6, 7, 10 and loading conditions

c) PF with distributed initial damage size

d) PF with true value of initial damage size

Figure 9. Comparison of NN and PF with the case of complex damage growth under small noise

results from the two methods are not much different, but PF and BM, respectively, have advantage in terms of wide range of applications and a fast calculation. Further, the case of no loading conditions and no physical models, short term prediction can be done by using data-driven approaches with at least three data sets. For long term predictions, loading conditions are additionally required.

5. Conclusions

This paper provides a practical review of both data-driven and physics-based approaches for the purpose of prognostics. As common prognostics algorithms, NN, GP, PF and BM are introduced and employed for case studies under practical conditions to discuss about attributes, pros and cons, and applicable conditions. Even if advanced algorithms are available, the basic algorithms are employed in this study, and the results are analyzed focusing on their intrinsic properties. This will be helpful for the beginners in prognostics to choose the best algorithm for their field of applications.

Acknowledgments

This work was supported by a grant from the International Collaborative R&D Program (0420-2011-0161) of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), funded by the Korean government's Ministry of Knowledge Economy.

References

An, D., Choi, J. H., Schmitz, T. L., & Kim, N. H. (2011). In-situ monitoring and prediction of progressive joint wear using Bayesian statistics. *Wear*, vol. 270(11-12), pp. 828-838.

An, D., & Choi, J. H. (2012). Efficient reliability analysis based on Bayesian framework under input variable and metamodel uncertainties. *Structural and Multidisciplinary Optimization*, vol. 46, pp. 533-547.

An, D., Choi, J. H., & Kim, N. H. (2012). A comparison study of methods for parameter estimation in the physics-based prognostics. *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, April 23-26, Honolulu, Hawaii.

An, D., Choi, J. H., & Kim, N. H. (2012). Identification of correlated damage parameters under noise and bias using Bayesian inference. *Structural Health Monitoring*, vol. 11(3), pp. 293-303.

An, D., & Choi, J. H. (2013). Improved MCMC method for parameter estimation based on marginal probability density function. *Journal of Mechanical Science and Technology*, vol. 27(6).

Andrieu, C., Freitas, de N., Doucet, A., & Jordan, M. (2003). An introduction to MCMC for machine learning. *Machine Learning*, vol. 50(1), pp. 5-43.

Bayes, T. (1763.) An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, vol. 53, pp. 370-418.

Belhouari, S. B., & Bermak, A. (2004). Gaussian process for nonstationary time series prediction. *Computational Statistics & Data Analysis*, vol. 47, pp. 705-712.

Bicciato, S., Pandin, M., Didonè, G., & Bello, C. D. (2001). Analysis of an associative memory neural network for pattern identification in gene expression data. *Biokdd01, Workshop on Data Mining in Bioinformatics*, August 26, San Francisco, CA.

Bodén, M. (2002). A guide to recurrent neural networks and backpropagation. in *The DALLASproject. Report from the NUTEK-supported project AIS-8: Application of Data Analysis with Learning Systems, 1999-2001.* Holst, A. (ed.), SICS Technical Report T2002:03, SICS, Kista, Sweden.

Bretscher, O. (1995). *Linear Algebra with Applications*, 3rd ed. Upper Saddle River NJ: Prentice Hall.

Chakraborty, K., Mehrotra, K., Mohan, C. K., & Ranka, S. (1992). Forecasting the behavior of multivariate time series using neural networks. *Neural Networks,* vol. 5, pp. 961-970.

Chang, J. F., & Hsieh, P. Y. (2011). Particle swarm optimization based on back propagation network forecasting exchange rates. *International Journal of Innovative Computing, Information and Control*, vol. 7(12), pp. 6837-6847.

Chen, S. C., Lin, S. W., Tseng, T. Y., & Lin, H. C. (2006). Optimization of back-propagation network using simulated annealing approach. *IEEE International Conference on Systems, Man, and Cybernetics*, October 8-11, Taipei, Taiwan.

Cheng, S., & Pecht, M. (2009). A fusion prognostics method for remaining useful life prediction of electronic products. *5th Annual IEEE Conference on Automation Science and Engineering*, August 22-25, Bangalore, India.

Chryssoloiuris, G., Lee, M., & Ramsey, A. (1996). Confidence interval prediction for neural network models. *IEEE Transactions ON Neural Networks*, vol. 7(1), pp. 229-232.

Coppe, A., Haftka, R. T., Kim, N. H., & Yuan, F. G. (2009). Reducing uncertainty in damage growth properties by structural health monitoring. *Annual Conference of the Prognostics and Health Management Society*, September 27-October 1, San Diego, CA.

Coppe, A., Pais, M. J., Haftka, R. T., & Kim, N. H. (2012). Remarks on using a simple crack growth model in predicting remaining useful life. *Journal of Aircraft*, vol. 49, pp. 1965-1973.

Daigle, M., & Goebel, K. (2011). Multiple damage progression paths in model-based prognostics. *IEEE Aerospace Conference*, March 05-12, Big Sky, Montana.

DeCastro, J. A., Tang, L., Loparo, K. A., Goebel, K., & Vachtsevanos, G. (2009). Exact nonlinear filtering and prediction in process model-based prognostics. *Annual Conference of the Prognostics and Health Management Society*, September 27-October 1, San Diego, CA.

Dickson, D. C. M., & Waters, H. R. (1993). Gamma processes and finite time survival probabilities. *Astin Bulletin*, vol. 23(2), pp. 259-272.

Doucet, A., De Freitas, N., & Gordon, N. J. (2001). *Sequential Monte Carlo methods in practice*. Springer-Verlag.

Doukim, C. A., Dargham, J. A., & Chekima, A. (2010). Finding the number of hidden neurons for an MLP neural network using coarse to fine search technique. *10th International Conference on Information Science, Signal Processing and their Applications*, May 10-13, Kuala Lumpur, Malaysia.

Drucker, H., Cortes, C., Jackel, L. D., LeCun, Y., & Vapnik, V. (1994). Boosting and other ensemble methods. *Neural Computation*, vol. 6(6), pp. 1289-1301.

Duch, W., & Jankowski, N. (1999). Survey of neural transfer functions. *Neural Computing Surveys*, vol. 2, pp. 163-212.

Efron, B., & Tibshirani, R. J. (1994). *An introduction to the bootstrap*. FL: Chapman & Hall/CRC.

Firth, A. E., Lahav, O., & Somerville, R. S., (2003). Estimating photometric redshifts with artificial neural networks. *Monthly Notices of the Royal Astronomical Society*, vol. 339, pp. 1195-1202.

Foster, L., Waagen, A., Aijaz, N., Hurley, M., Luis, A., Rinsky, J., Satyavolu, C., Way, M. J., Gazis, P., & Srivastava, A. (2009). Stable and efficient gaussian process calculations. *Journal of Machine Learning Research*, vol. 10, pp. 857-882.

Gelfand, A. E., & Sahu, S. K. (1994). On Markov chain Monte Carlo acceleration. *Journal of Computational and Graphical Statistics*, vol. 3, pp. 261-276.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian data analysis*. NY: Chapman and Hall.

Gilks, W. R., & Berzuini, C. (2001). Following a moving target-Monte Carlo inference for dynamic Bayesian models. *Royal Statistical Society B*, vol. 63, Part 1, pp. 127-146.

Giurgiutiu, V. (2002). Current issues in vibration-based fault diagnostics and prognostics. *SPIE's 9th Annual International Symposium on Smart Structures and Materials and 7th Annual International Symposium on NDE for Health Monitoring and Diagnostics*, March 17-21, San Diego, CA.

Goebel, K., Saha, B., & Saxena, A. (2008). A comparison of three data-driven techniques for prognostics. *Proceedings of the 62nd Meeting of the Society For Machinery Failure Prevention Technology (MFPT)*, May 6-8, Virginia Beach, VA.

Gómez, I., Franco, L., & Jérez, J. M. (2009). Neural network architecture selection: can function complexity help?. *Neural Processing Letters*, vol. 30, pp. 71-87.

Gouriveau, R., Dragomir, O., & Zerhouni, N. (2008). A fuzzy approach of online reliability modeling and estimation. *Proceedings of the 34th European Safety Reliability & Data Association, ESReDA Seminar and 2nd Joint ESReDA/ESRA Seminar on Supporting Technologies for Advanced Maintenance Information Management*, May 13-14, San Sebastian, Spain.

Grall, A., Bérenguer, C., & Dieulle, L. (2002). A condition-based maintenance policy for stochastically deteriorating systems. *Reliability Engineering and System Safety*, vol. 76(2), pp. 167-180.

Gu, J., Azarian, M. H., & Pecht, M. G. (2008). Failure prognostics of multilayer ceramic capacitors in temperature-humidity-bias conditions. *2008 International Conference on Prognostics and Health Management*, October 6-9, Denver, Colorado.

Guan, X., Liu, Y., Saxena, A., Celaya, J., & Goebel, K. (2009). Entropy-Based probabilistic fatigue damage prognosis and algorithmic performance comparison. *Annual Conference of the Prognostics and Health Management Society*, September 27-October 1, San Diego, CA.

Happel, B. L. M., & Murre, J. M. J. (1994). The design and evolution of modular neural network architectures. *Neural Networks*, vol. 7, pp. 985-1004.

He, Y., Tan, Y., & Sun, Y. (2004) Wavelet neural network approach for fault diagnosis of analogue circuits. *IEE Proceedings-Circuits, Devices and Systems*, vol. 151(4), pp. 379-384.

Higuchi, T. (1997). Monte Carlo filter using the genetic algorithm operators. *Journal of Statistical Computation and Simulation*, vol. 59(1), pp. 1-23.

Huang, X., Torgeir, M., & Cui, W. (2008). An engineering model of fatigue crack growth under variable amplitude loading. *International Journal of Fatigue*, vol. 30(1), pp. 2-10.

Jacobs, R. A. (1995). Methods for combining experts' probability assessments. *Neural Computation*, vol. 7(5), pp. 867-888.

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, vol. 20(7), pp. 1483-1510.

Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, vol. 92(3), pp. 401-422.

Juricic, D., Znidarsic, A., & Fussel, D. (1997). Generation of diagnostic trees by means of simplified process models and machine learning. *Engineering Applications of Artificial Intelligence*, vol. 10(1), pp. 15-29.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transaction of the ASME-Journal of Basic Engineering*, No. 82, pp. 35-45.

Khawaja, T., Vachtsevanos, G., & Wu, B. (2005). Reasoning about uncertainty in prognosis: a confidence prediction neural network approach. *2005 Annual Meeting of the North American Fuzzy Information Processing Society*, June 22-25, Ann Arbor, Michigan.

Kim, S., & Park, J. S. (2011). Sequential Monte Carlo filters for abruptly changing state estimation. *Probabilist Engineering Mechcnics*, vol. 26, pp. 194-201.

Kitagawa, G. (1987). Non–Gaussian state space modeling of nonstationary time series (with discussion). *Journal of the American Statistical Association*, vol. 82(400), pp. 1032-1063.

Kleijnen, J. P. C. (1995). Statistical validation of simulation models. *European Journal of Operational Research*, vol. 87, pp. 21-34.

Kramer, S. C., & Sorenson, H. W. (1998). Bayesian parameter estimation. *IEEE Transactions on Automatic Control*, vol. 33(2), pp. 217-222.

Krogh, A. (2008). What are artificial neural networks?. *Nature Biotechnology*, vol. 26(2), pp. 195-197.

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In Tesauro G., Touretzky D., & Leen T. (Eds.), *Advances in Neural Information Processing Systems* (231-238). The MIT Press.

Lawrence, N., Seeger, M., & Herbrich, R. (2003). Fast sparse Gaussian Process methods: the information vector machine. In Becker S., Thrun S., & Obermayer K. (Eds.), *Advances in Neural Information Processing Systems*(625-632), Vol. 15, MIT Press.

Lawrence, S., Giles, C. L., & Tsoi, A. C. (1998). *What size neural network gives optimal generalization? convergence properties of backpropagation*. Technical Reports. UM Computer Science Department, UMIACS.

Leonard, J. A., Kramer, M. A., & Ungar, L. H. (1992). A Neural network architecture that computes its own reliability, *Computers in Chemical Engineering*, vol. 16(9), pp. 819-835.

Liu, D., Pang, J., Zhou, J., & Peng, Y. (2012). Data-driven prognostics for lithium-ion battery based on Gaussian process regression. *2012 Prognostics and System Health Management Conference*, May 23-25, Beijing, China.

Liu, J., Saxena, A., Goebel, K., Saha, B., & Wang, W. (2010). An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries. *Annual Conference of the Prognostics and Health Management Society*, October 10-16, Portland, Oregon.

Liu, P., & Li, H. (2004). *Fuzzy Neural Network Theory and Application*. Singapore: World Scientific.

MacKay, D. J. C. (1997). Gaussian processes-a replacement for supervised neural networks?, *Tutorial lecture notes for NIPS*, UK, http://www.inference.phy.cam.ac.uk/mackay/BayesGP.html.

Mackay, D. J. C. (1997). *Introduction to Gaussian processes*. Technical report. Cambridge University, UK, http://www.cs.utoronto.ca/~mackay/gpB.pdf

Mao, K. Z., Tan, K.-C., & Ser, W. (2000). Probabilistic neural-network structure determination for pattern classification. *IEEE Transactions on Neural Networks*, vol. 11(4), pp. 1009-1016.

Martin, K. F. (1994). A review by discussion of condition monitoring and fault diagnosis in machine tools. *International Journal of Machine Tools and Manufacture*, vol. 34(4), pp. 527-551.

Melkumyan, A., & Ramos, F. (2009). A sparse covariance function for exact gaussian process inference in large datasets. *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*, July 11-17, Pasadena, CA.

Mohanty, S., Teale, R., Chattopadhyay, A., Peralta, P., & Willhauck, C. (2007). Mixed Gaussian process and state-space approach for fatigue crack growth prediction. *International Workshop on Structural Heath Monitoring*, vol. 2, pp. 1108-1115.

Mohanty, S., Das, D., Chattopadhyay, A., & Peralta, P. (2009). Gaussian process time series model for life prognosis of metallic structures. *Journal of Intelligent Material Systems and Structures*, vol. 20, pp. 887-896.

Naftaly, U., Intrator, N., & Horn, D. (1997). Optimal ensemble averaging of neural networks. *Network: Computation in Neural Systems*, vol. 8(3), pp. 283-296.

Navone, H. D., Granitto, P. M., Verdes, P. F., & Ceccatto, H. A. (2001). A Learning algorithm for neural network ensembles. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*. No. 12, pp 70-74.

Nawi, N. M., Ransing, R. S., & Ransing, M. R. (2007). An improved conjugate gradient based learning algorithm for back propagation neural networks. *International Journal of Computational Intelligence*, vol. 4(1), pp. 46-55.

Neal, R. M. (1998). Regression and classification using Gaussian process priors. In Bernardo J. M., Berger J. O., Dawid A. P., & Smith A. F. M. (Eds.), *Bayesian statistics* (475-501), Vol. 6, NY: Oxford University Press.

Orchard, M. E., & Vachtsevanos, G. J. (2007). A particle filtering approach for on-line failure prognosis in a planetary carrier plate. *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 7(4), pp. 221-227.

Ostafe, D. (2005). Neural network hidden layer number determination using pattern recognition techniques. *2nd Romanian-Hungarian Joint Symposium on Applied computational Intelligence*, May 12-14, Timisoara, Romania.

Paciorek, C., & Schervish, M. J. (2004). Nonstationary covariance functions for gaussian process regression. In

Thrun S., Saul L., & Schölkopf B. (Eds.), *Advances in neural information processing systems*, Vol. 16, MIT Press.

Pandey, M. D., & Noortwijk, J. M. V. (2004). Gamma process model for time-dependent structural reliability analysis. In Watanabe E., Frangopol D. M., & Utsonomiya T. (Eds.), *Bridge maintenance, safety, management and cost* (101-102). London: Taylor and Francis Group.

Paris, P. C., & Erdogan, F. (1963). A critical analysis of crack propagation laws. *Transactions of the ASME, Journal of Basic Engineering, Series D*, vol. 85(3), pp. 528-534.

Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, vol. 33(3), pp. 1065-1076.

Perrone, M. P., & Cooper, L. N. (1993). When networks disagree: ensemble method for neural networks. In Mammone R. J. (Ed.), *Neural Networks for Speech and Image processing*, Chapman-Hall.

Petalas, P., Spyridonos, P., Glotsos, D., Cavouras, D., Ravazoula, P., & Nikiforidis, G. (2003). Probabilistic neural network analysis of quantitative nuclear features in predicting the risk of cancer recurrence at different follow-up times. *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis*, September 18-20, Rome, Italy.

Psichogios, D. C., & Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, vol. 38(10), pp. 1499-1511.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, vol. 77(2), pp. 257-286.

Rasmussen, C. E., & Williams, C. K. I., (2006). *Gaussian Processes for Machine Learning*. Cambridge, MA: The MIT Press

Rebba, R., Huang, S., Liu, Y., & Mahadevan, S. (2006). Statistical validation of simulation models. *International Journal of Materials and Product Technology*, vol. 25(1/2/3), pp. 164-181.

Rebba, R., Mahadevan, S., & Huang, S. (2006). Validation and error estimation of computational models. *Reliability Engineering and System Safety*, vol. 91, pp. 1390-1397.

Rovithakis, G. A., Maniadakis, M., & Zervakis, M., (2004). A hybrid neural network/genetic algorithm approach to optimizing feature extraction for signal classification. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 34(1), pp. 695-702.

Rubin, D. B. (1998). Using the SIR algorithm to simulate posterior distributions. In Bernardo J. M., DeGroot M. H., Lindley D. V., & Smith A. F. M. (Eds.), *Bayesian statistics*(395-402), vol. 3, Cambridge, MA: Oxford University Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. *Parallel Distributed Processing: Ex-plorations in the Microstructure of Cognition*, vol. 1: foundations, MIT Press, pp. 318-362.

Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, vol. 4(4), pp. 409-423.

Saha, B., Goebel, K., & Christophersen, J. (2009). Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement & Control*, vol. 31(3-4), pp. 293-308.

Salomon, R., & Hemmen, J. L. V. (1996). Accelerating backpropagation through dynamic self-adaptation. *Neural Networks*, vol. 9(4), pp. 589-601.

Samanta, B., & Al-Balushi, K. R. (2003). Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. *Mechanical Systems and Signal Processing*, vol. 17(2), pp. 317–328.

Samuel, P. D., & Pines, D. J. (2005). A review of vibration-based techniques for helicopter transmission diagnostics. *Journal of Sound and Vibration*, vol. 282(1-2), pp. 475-508.

Sankavaram, C., Pattipati, B., Kodali, A., Pattipati, K., Azam, M., Kumar, S., & Pecht, M. (2009). Model-based and data-driven prognosis of automotive and electronic systems. *5th Annual IEEE Conference on Automation Science and Engineering*, August 22-25, Bangalore, India.

Santner, T. J., Williams, B. J., & Notz, W. I. (2003). *The design and analysis of computer experiments*. NY: Springer Verlag.

Sargent, R. G. (2009). Verification and validation of simulation models. *Proceedings of the 2009 Winter Simulation Conference*, December 13-16, Austin, TX.

Seeger, M. (2004). Gaussian processes for machine learning. *International Journal of Neural Systems*, vol. 14(2), pp. 69-106.

Si, X. S., Wang, W., Hu, C. H., & Zhou, D. H (2011). Remaining useful life estimation – A review on the statistical data driven approaches. *European Journal of Operational Research*, vol. 213, pp. 1-14.

Singh, G. K., & Al Kazzaz, S. A. S. (2003). Induction machine drive condition monitoring and diagnostic research-a survey. *Electric Power Systems Research*, vol. 64, pp. 145-158.

Smola, A. J., & Bartlett, P. L. (2001). Sparse greedy Gaussian process regression. In Leen T. K., Dietterich T. G., & Tresp V. (Eds.), *Advances in Neural Information Processing Systems* (619-625), Vol. 13. MIT Press.

Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, vol. 3, pp. 109-118.

Srivastava, A. N., & Das, S. (2009). Detection and prognostics on low-dimensional systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 39(1), pp. 44-54.

Storvik, G. (2002). Particle filters in state space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, vol. 50(2), pp. 281-289.

Subudhi, B, Jena, D., & Gupta, M. M. (2008). Memetic differential evolution trained neural networks for nonlinear system identification. *IEEE Region 10 Colloquium and the Third International Conference on Industrial and Information Systems*, December 8 -10, Kharagpur, India.

Sugumaran , V., Sabareesh, G. R., & Ramachandran, K. I. (2008). Fault diagnostics of roller bearing using kernel based neighborhood score multi-class support vector machine. *Expert Systems with Applications*, vol. 34, pp. 3090-3098.

Svozil, D., Kvasnička, V., & Pospíchal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, vol. 39, pp. 43-62.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, vol. 1, pp. 211-244.

Tran, V. T., & Yang, B. S. (2009). Data-driven approach to machine condition prognosis using least square regression tree. *Journal of Mechanical Science and Technology*, vol. 23, pp. 1468-1475.

Veaux, R. D., Schumi, J., Schweinsberg, J., & Ungar, L. H. (1998). Prediction intervals for neural networks via nonlinear regression. *Technometrics*, vol. 40(4), pp. 273-282.

Wang, W. P., Liao, S., & Xing, T. W. (2009). Particle filter for state and parameter estimation in passive ranging. *IEEE International Conference on Intelligent Computing and Intelligent Systems*. November 20-22, Shanghai, China.

Williams, C. K. I. (1997). Computing with infinite networks. In Mozer M. C., Jordan M. I., & Petsche T. (Eds.), *Advances in neural information processing systems*, Cambridge, MA: MIT Press.

Xing, Y., Miao, Q., Tsui, K.-L., & Pecht, M. (2011). Prognostics and health monitoring for lithium-ion battery,” *2011 IEEE International Conference*, July 10-12, Beijing, China.

Xing, Y., Williard, N., Tsui, K.-L., & Pecht, M. (2011). A comparative review of prognostics-based reliability methods for lithium batteries. *Prognostics and System Health Management Conference*, May 24-25, Shenzhen, China.

Xu, J., & Xu, L. (2011). Health management based on fusion prognostics for avionics systems,” *Journal of Systems Engineering and Electronics*, vol. 22(3), pp. 428–436.

Yam, R. C. M., Tse, P. W., Li, L., & Tu, P. (2001). Intelligent predictive decision support system for condition-based maintenance. *International Journal of Advanced Manufacturing Technology*, vol. 17(5), pp. 383-391.

Yan, R., & Gao, R. X. (2007). Approximate entropy as a diagnostic tool for machine health monitoring. *Mechanical Systems and Signal Processing*, vol. 21, pp. 824–839.

Yang, L., Kavli, T., Carlin, M., Clausen, S., & Groot, P. F. M. (2000). An evaluation of confidence bound estimation methods for neural networks. *European Symposium on Intelligent Techniques 2000*, September 14-15, Aachen, Germany.

Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, vol. 87(9), pp. 1423-1447.

Zhang, J., & Lee, J. (2011). A review on prognostics and health monitoring of li-ion battery. *Journal of Power Sources*, vol. 196, pp. 6007-6014.

Zio, E., & Maio, F. D. (2010). A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliability Engineering and System Safety*, vol. 95, pp. 49-57.

Dawn An received the B.S. degree and M.S. degree of mechanical engineering from Korea Aerospace University in 2008 and 2010, respectively. She is now a joint Ph.D. student at Korea Aerospace University and the University of Florida. Her current research is focused on the Bayesian inference, correlated parameter identification and the methodology for prognostics and health management and structural health monitoring.

Nam Ho Kim received the B.S. degree of mechanical engineering from Seoul National University in 1989, the M.S. degree and Ph.D. degree of mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1991 and the University of Iowa in 1999, respectively. He worked as a Postdoctoral Associate at the University of Iowa from 1999 to 2001. He joined the Dept. of Mechanical & Aerospace Engineering at the University of Florida, in 2002 and is now Associate Professor. His current research is focused on design under uncertainty, design optimization of automotive NVH problem, shape DSA of transient dynamics (implicit/explicit) and structural health monitoring.

Joo-Ho Choi received the B.S. degree of mechanical engineering from Hanyang University in 1981, the M.S. degree and Ph.D. degree of mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1983 and 1987, respectively. During the year 1988, he worked as a Postdoctoral Fellow at the University of Iowa. He joined the School of Aerospace and Mechanical Engineering at Korea Aerospace University, Korea, in 1997 and is now Professor. His current research is focused on the reliability analysis, design for life-time reliability, and prognostics and health management.

Experimental Tests of Wind Turbine Main Shaft Motion on a Laboratory Test Rig

Dirk Lutschinger[1], Ian Howard[2]

[1,2]*Curtin University, Perth, Western Australia, 6845, Australia*

d.lutschinger@curtin.edu.au
I.Howard@exchange.curtin.edu.au

Abstract

This paper introduces research and discusses findings dealing with failure modes of gearboxes in wind turbines. These gearboxes fail in general after five years which is far below the expected design life of twenty years of a wind turbine. The research is taking a more holistic approach towards finding typical behaviour of the main shaft taking the transient nature of the wind into consideration. In this research, a small scale wind turbine test rig has been designed and manufactured with displacement sensors installed to observe the displacement of the main shaft at specific points, namely the main bearing locations of the forward framework of a wind turbine nacelle, where the main shaft is installed. The experimental data measured from the test rig is being analysed with common beam bending, statistical and fatigue theories to draw conclusions for long term loading in service. Aspects of the turbulent nature of the wind driving the wind turbine have been taken into consideration as being part of the aerodynamic loading onto the rotor and eventually the gearbox, transmitted through the main shaft. The purpose of the test rig at this stage is to obtain a quantitative insight into the motion of the main shaft. The deliberately chosen softer aluminum material and the more slender geometry for the components should provide exaggerated displacements which help to make motion and deformations more obvious. At this point, no resemblance to real size wind turbines has been established.

1. Introduction

Gearboxes of horizontal wind turbines have a tendency to fail in service before the end of the design life of the wind turbine itself (Van Rensselar, 2010). According to Musial, Butterfield and McNiff (2007), the failure of the gearbox begins in the bearings of the gears inside the gearbox. In this project an experimental approach is conducted to determine the typical motion of the main shaft that connects the rotor hub with the gearbox. The shaft is in general rigidly connected with the carrier arm of the first planetary stage through a ring that applies pressure onto the surfaces of both components (Rexroth Bosch Group, 2010). Figure 1 shows a sketch of a typical gearbox with three planetary stages used in wind turbines. The above mentioned rigid connection is visible here as well. Misalignments between the main shaft and the first planetary stage are expected to contribute to the failure problems. Also indicated are typical gear shaft or carrier arm bearing locations. Moser (2010) states that one reason for the gearbox failure is the turbulent nature of the wind. The flow field around a wind turbine in real wind situations is turbulent. It has randomly distributed eddies with small and large sizes. These eddies hit the three blades of a rotor, a three bladed rotor assumed, with different speeds and under different angles at different locations along the blades. Hence they are expected to cause different local aerodynamic forces which apply a resultant moment onto the hub of the rotor which is changing in magnitude and direction over time.

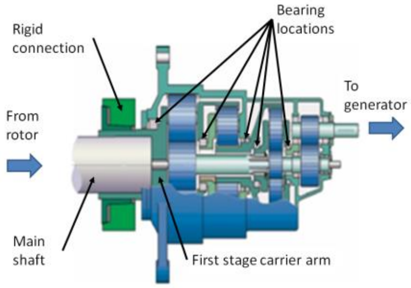


Figure 1. Typical gearbox layout (Rexroth Bosch Group, 2010)

The experimental design in this research has a main shaft realised with two main bearings holding the shaft. The displacements of these bearings in the plane perpendicular to the main shaft have been measured. In particular the displacements along two axes, orthogonal to each other have been recorded over time. The entire experiment has taken place in a test set up in a laboratory. The set up comprised

Dirk Lutschinger et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

of three main components, a fan, an aerodynamic element, the function of which will be explained later in this paper, and a wind turbine test rig. As a fourth component, a board containing the electrical devices for the measurement equipment has been constructed. Figure 2 illustrates the overall experimental setup of the four components.

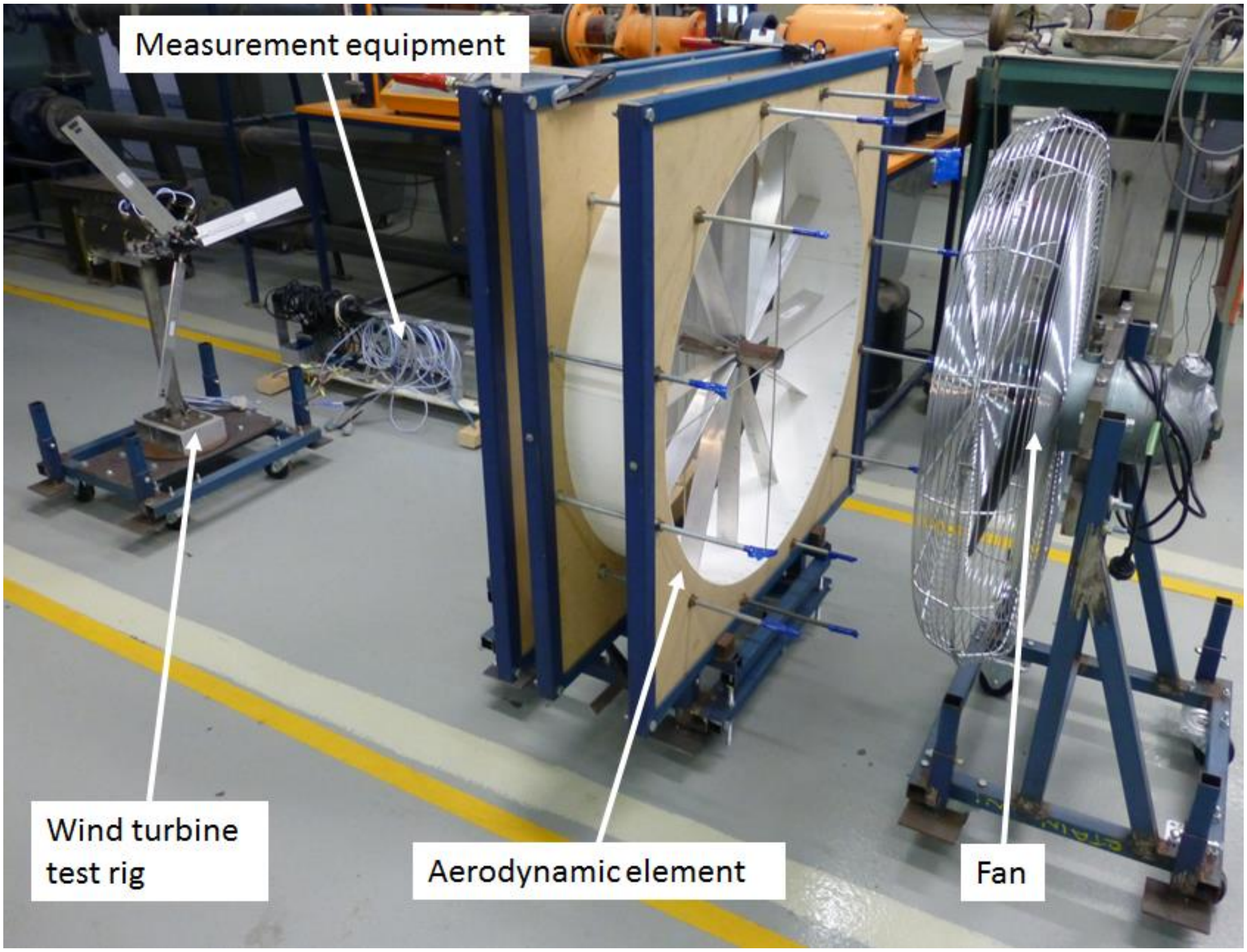


Figure 2. Wind Turbine test set up

2. Experimental Tests

2.1. Wind Simulation

The tests were performed inside a laboratory with a conventional industrial fan as the flow source. The flow behind a fan, according to Eck (1973) is built up as a concentrated flow with rotation and a vortex core inside, parallel to the flow direction and centered around the central line of the hub of the fan. To straighten the flow again and to get rid of the vortex, an aerodynamic construction has been installed behind the fan. It consists of a stator and vertical vanes. This design has been taken over from general arrangements of low-speed wind tunnels as described by Pope & Harper (1966). This component, that is actually acting as sort of an aerodynamic filter can be seen in Figure 2 in the center of the image.

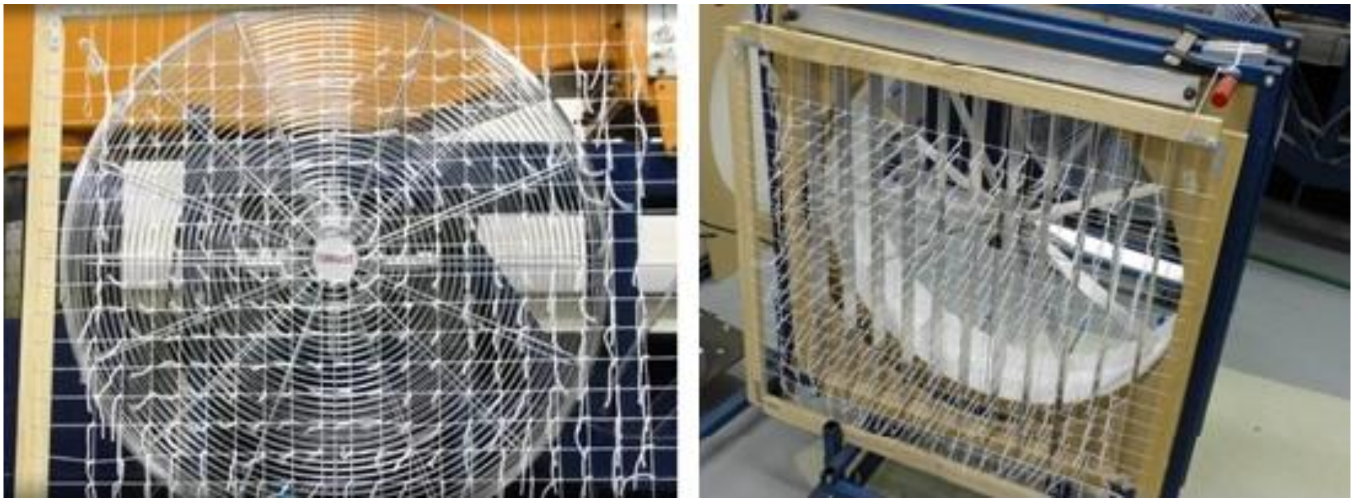


Figure 3: Flow vortex behind fan (left) and straightened flow with aerodynamic element (right)

After placing the “aerodynamic filter” downstream of the fan, a more even flow was obtained. In the left side of Figure 3, the flow with the vortex after the fan, visualized with cotton strips, can be seen whereas on the right side of the figure, the straightened flow has been made obvious with the cotton strips. Test campaigns have been performed to investigate the wind speeds downstream at designated locations. For that a micromanometre has been used. It was connected to a pitot tube that has been placed onto the central line at the height of the hub of the fan. Measurements were taken downstream in 1 meter steps up to 4 meters away from the “aerodynamic filter”. Measured data has been put together in Table **1**. The values represent overall wind speeds and their fluctuations in meters per seconds. The first column divides the table into three different fan rotational speed levels with 1 being the slowest and 3 the fastest.

The measurements showed that for distances greater than 3 meters the fluctuation of the speeds significantly increased. This was also experienced with a simple hand test placing the hand into the flow and “feeling” the change of the pressure induced onto the hand’s surface by the flow. Through this test a more turbulent flow in this general area was assumed. The measurements of main shaft displacements were conducted with the wind turbine test rig being positioned 3 metres behind the straightener vanes, in the transient wind speed zone.

2.2. Main Shaft Motion

The test rig was designed in a way so that the basic structure of a main shaft and gearbox installation has been realised with two main shaft bearings and the corresponding brackets to hold them as well as the bracket to hold the gearbox. This structure could be seen in Figure 4 in a design example from Rexroth Bosch. The gearbox in the experiment has not been reproduced with gears but only the front wall and the bearing holding the main shaft in the front wall of the gearbox has been installed. To simulate the total inertia of the gearbox a basic calculation was performed with methods from machine dynamics. As a geometrical reference, the gear arrangement sketched in Figure 1 has been used. A fly wheel with a corresponding inertia has been manufactured and installed onto the shaft. The two main shaft brackets with the main shaft bearings, shown in Figure 5 as Bearing 1 and Bearing 2, have been designed with two blocks each on extensions with a relative angle towards each other of 90° and 45° towards the horizontal. This has the purpose to measure displacements of the blocks with proximity probes. The sizes of the blocks have been chosen according to requirements from the proximity probe handbook to build up the necessary eddy current to function properly. The distance of the blocks from the rotational axis has been chosen for the same reason. There are two proximity probes for each main shaft bearing, measuring the displacements in two axes perpendicular to each other. The probes themselves have been mounted onto steel brackets to keep their movements negligible in relation to the aluminium shaft and bracket arrangement. The shaft continues through the gearbox wall and the bearing where the main shaft enters the gearbox.

Table 1. Flow speeds and fluctuations in [m/s] at different distances behind the straightener vanes for different fan speed settings

Fan level	1 [m]	2 [m]	3 [m]	4 [m]
1	6.0 ± 0.3	6.2 ± 0.5	5.1 ± 0.6	4.2 ± 0.7
2	6.6 ± 0.2	6.9 ± 0.4	5.4 ± 0.5	4.5 ± 0.8
3	7.0 ± 0.5	7.1 ± 0.2	5.7 ± 0.4	4.9 ± 1.1

Figure 4. Main shaft and bearing framework example (Rexroth Bosch Group, 2010)

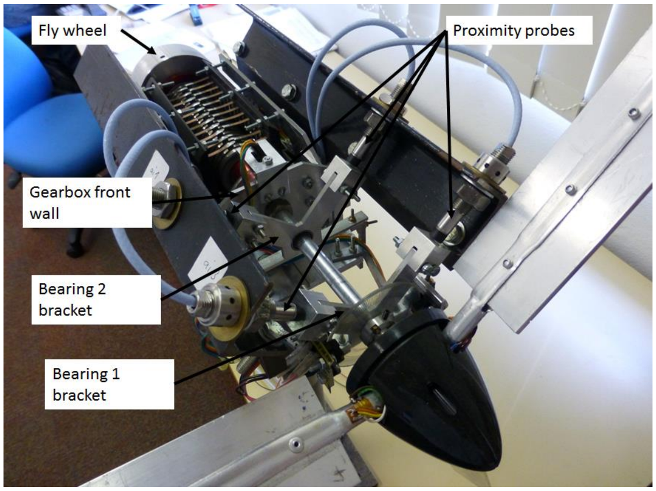


Figure 5. Wind turbine test rig component arrangement

3. Theoretical Consideration

3.1. Forces on blades and resulting moment in rotor hub

The theoretical approach taken in this project to deal with the aerodynamic forces on each blade is to assume a resulting aerodynamic force in the so called pressure point of the blade, depending on the flow around it. The flow environment is considered highly turbulent, with eddies of different sizes and flow velocities at different locations in the flow field at one point in time, as can be seen simply sketched in Figure 6. Every blade will experience a different flow and hence a different magnitude and direction of the resulting aerodynamic force and also the pressure point will be at a different location or radius from the rotational axis of the rotor. This resulting force can then be split up into its components in the rotational plane of the rotor and perpendicular to this plane, parallel to the rotational axis, as shown in Figure 7**.**

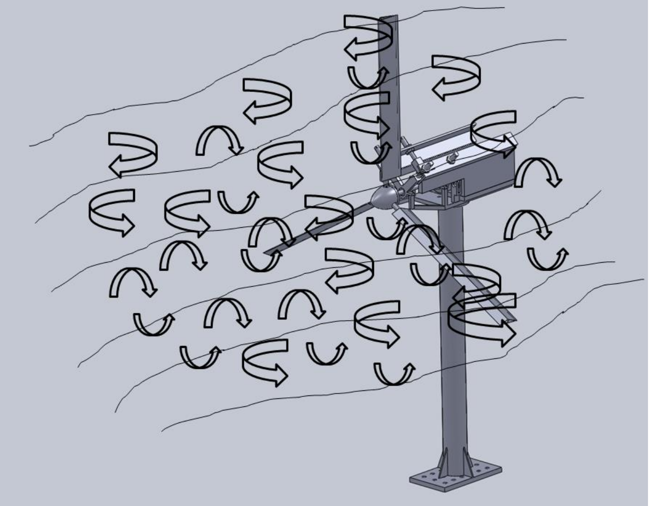


Figure 6. Sketched turbulent wind flow with eddies

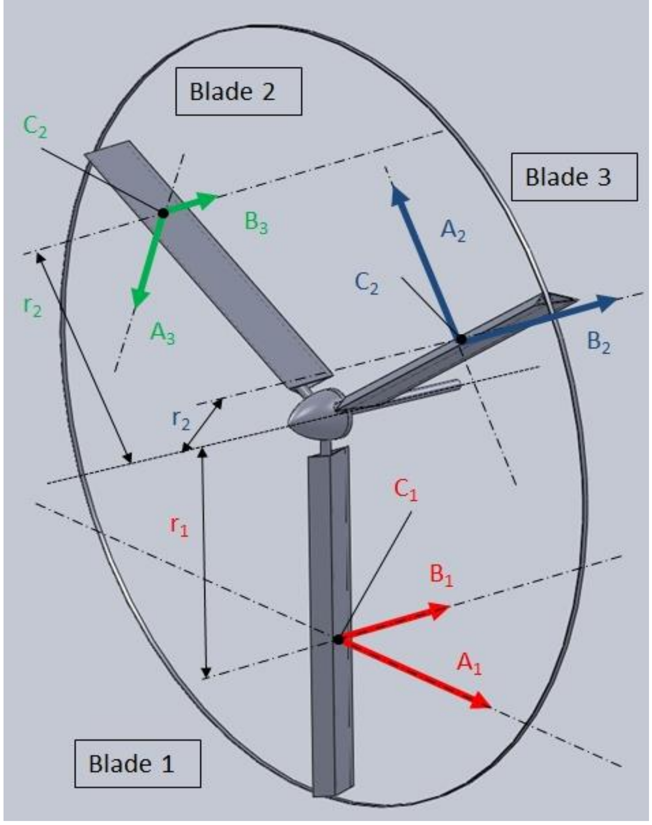


Figure 7. Aerodynamic force components on individual blades

Here the pressure point has been labeled as C, the corresponding radii are shown with r and the two force components, one in-plane, A, and out-of-plane, B, all individually numbered for the three blades. When only the B force components are considered, then together with their radii, they create a resulting moment in the rotor hub, in a plane through the main shaft axis, oriented in space depending on the magnitude of the individual moments of each blade.

Components A contribute to the rotational motion of the rotor. The effect that this moment has on the main shaft is, as expected from simple beam bending theory, a deformation of the framework holding the main shaft. This deformation will be restrained at the bearing locations on the main shaft. However, since the framework has a certain stiffness and is not assumed rigid here, certain displacements of the bearings can be observed. A free body diagram of the main shaft with the external loading moment M_a and the bearing supports along with their displacements δ_1 and δ_2 for the two bearings can be seen in Figure 8.

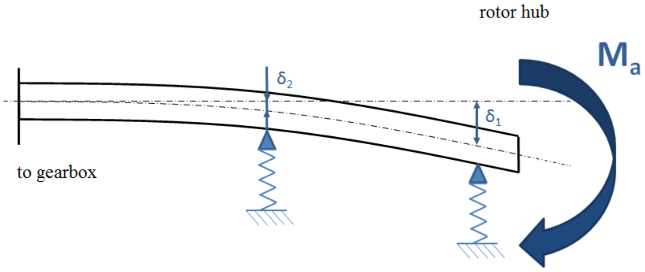


Figure 8. Free body diagram of main shaft

In this figure, the stiffness of the framework and the stiffness of the bearing brackets in particular, have been represented as springs.

4. TEST RESULTS AND ANALYSIS

For a more common method of viewing the main shaft movement, the displacement data can be mathematically transformed into horizontal and vertical components.

Figure 9 and Figure 10 represent orbit plots, where the vertical displacement component of a bearing is plotted against its corresponding horizontal displacement component. This is shown in Figure 9 for Bearing 1 and in Figure 10 for Bearing 2. When plotted in this fashion, it is obvious that the shaft is mainly oscillating in the vertical plane through the main shaft central axis. For Bearing 1 the magnitude of the horizontal amplitude around the middle axis is only about 15% of the magnitude of the amplitude in the vertical plane. For Bearing 2, this relation is 11.5%. In the plots, a certain deviation from the vertical plane is visible.

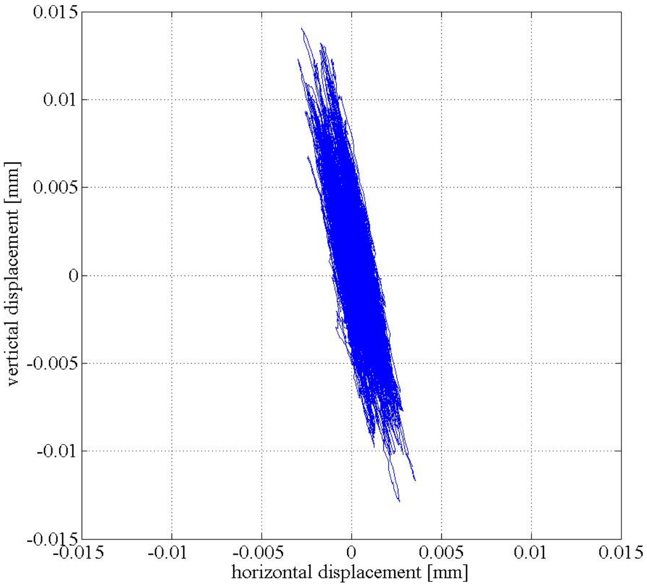


Figure 9. Orbit plot Bearing 1

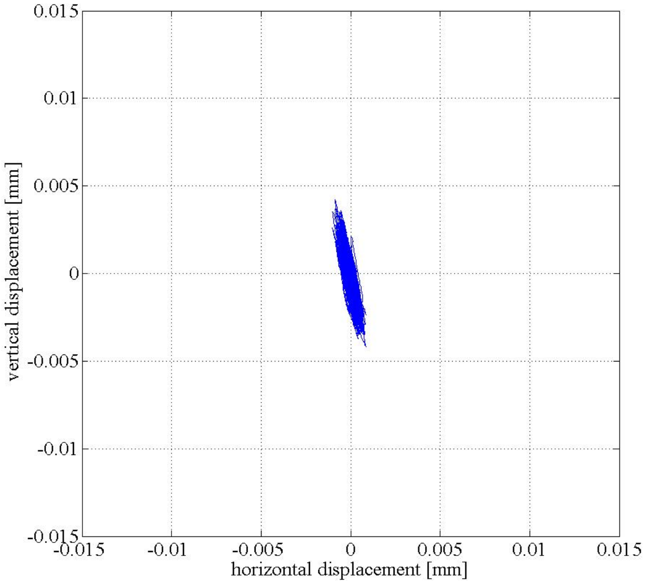


Figure 10. Orbit plot Bearing 2

It has been assumed to be caused by an asymmetrical geometrical error of the test rig construction. Further investigation will be conducted to shed light onto this effect in following tests. Disregarding this effect, it still allows for the simplified approach of assuming the main shaft being alternatingly bent in more or less a vertical plane.

Figure 11 is an extract of the data from this experiment. The curves represent the vertical displacements for Bearing 1 and Bearing 2. It is clearly seen that the direction of displacement for the two bearings relative to each other is the same. When Bearing 1 is moving upward, Bearing 2 does the same movement and there is no counteracting movement of the two bearings.

Figure 12 shows a close up of the structure that holds the bearings and hence the main shaft. By inspection, and not quantifying any values, the design suggests that the stiffness in the horizontal plane of this framework is much higher than in the vertical plane of the main shaft. It is suspected that this fact contributes to the orbital plots appearing the way they do with a larger amplitude in the vertical plane compared to the horizontal. A moment that is applied at the hub, resulting from aerodynamic loads, as described in section 3.1, will see its vertical component having a higher bending effect onto the main shaft than its horizontal component. Therefore, in further investigations in this paper, for simplifying reasons, only the vertical displacements are being considered.

Two different effects can be considered for the displacements of the main shaft, when it is continued through the gearbox wall.

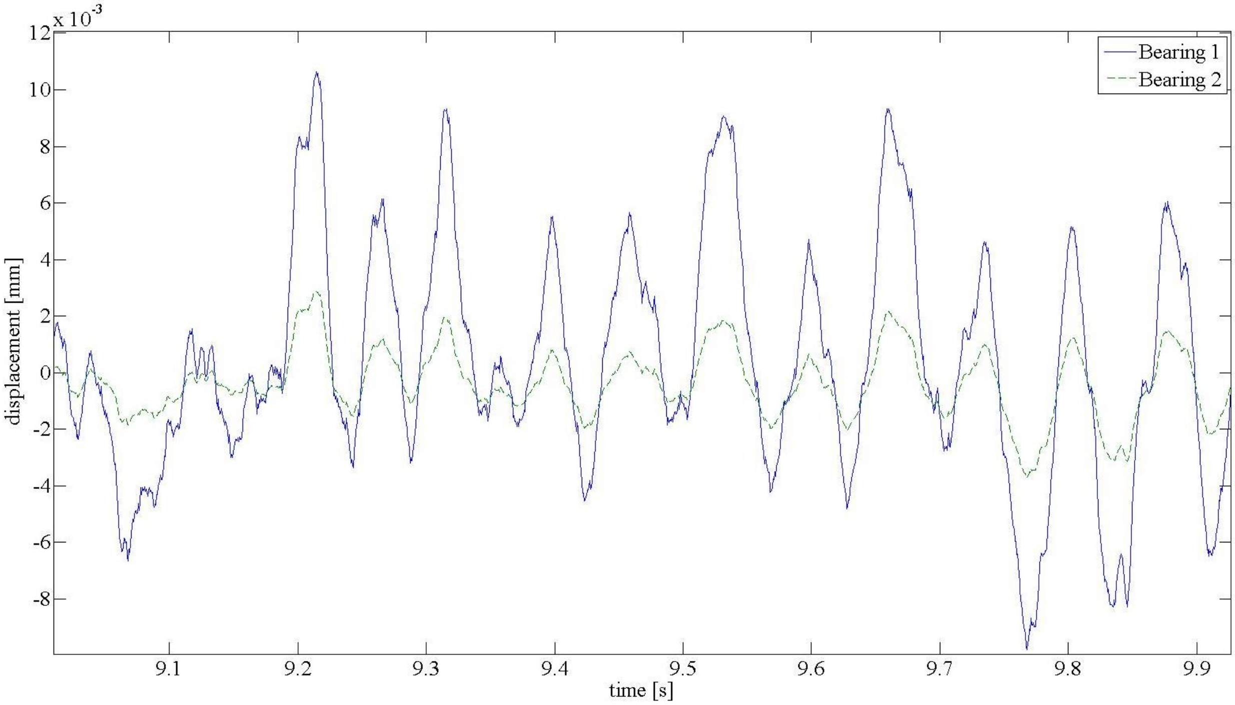


Figure 11. Vertical displacements Bearing 1 and Bearing 2

Firstly, if the first bearing in the gearbox can be considered to have a certain angular clearance when a moment is applied, there will be limited resistance from the bearing against the bending moment. The resulting angular displacement, which in beam bending theory is the slope of the bending curve, will be transmitted into the first gear stage of the gearbox. An oscillating movement of the kind detected in this experiment is likely to cause a damaging effect over a longer period of time in the gearbox.

Secondly, if the bearing does not allow for any movement due to the moment from the main shaft, then the bearing, which is generally made from a harder material than the gearbox wall, would be expected to follow the main shafts angular movement. This will effectively bend and distort the gearbox wall, having a similar effect on the first gear stage. Again, a long term destructive effect would be expected. In this project, further experimental investigations are planned to be able to prove the link between the aerodynamic effects and main shaft motion. Simulation methods will also be used for comparison and validation.

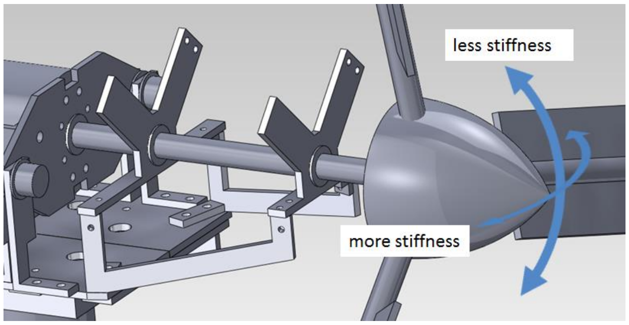


Figure 12. Main shaft framework stiffnesses

Another investigation can be performed with the experimentally obtained data when considering the theory of fatigue in regards to repetitive loading of components or system members. As described earlier, the horizontal movements are rather small compared to the vertical oscillations so that for a simplified approach only the vertical displacements are considered. It is obvious that the peaks of the displacements group in different ranges with different numbers of reoccurrences and a statistical approach can be taken. Since the two bearings move somewhat synchronously only the data for Bearing 1 is considered here. A data peak histogram was created using data similar to that shown in Figure 11, from a time record covering 20 seconds. The number of peaks in the range between 0 mm and 0.0025 mm was 2373, the number of peaks in the range between 0.0025 mm and 0.0075 mm was 2480, the number of peaks in the range between 0.0075 mm and 0.0125 mm was 264, and between 0.0125 mm and 0.0175 mm was 6.

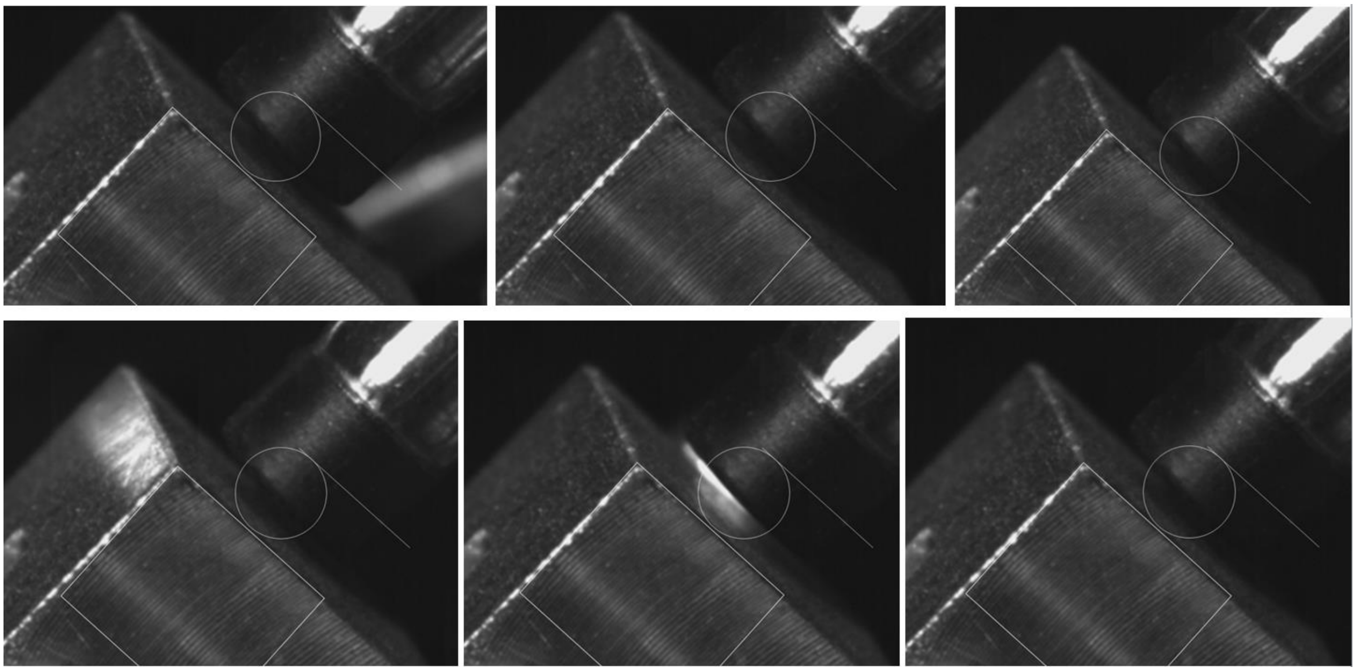


Figure 13. High speed camera sequence of proximity probe tip

Over the given timeframe of the experiment, with a simple extrapolation, it could be calculated that these numbers reach 10^6 cycles after 8428s or 2.34h for the 0 mm to 0.0025 mm range, 8064s or 2.3h for the 0.0025 mm to 0.0075 mm range, 75757s or 21h for the 0.0075 mm to 0.0125 mm range and 3333333s or 39 days for the 0.0175 mm to 0.0125 range.

The 10^6 cycle number has a special meaning in fatigue theory, being the critical number of cycles for members made of steel at which failure and component lifetime can be measured. In a so called S-N curve, a member loaded repeatedly above the limit specified in the S-N curve will fail before reaching 10^6 cycles, whereas the member loaded below this critical limit, will survive, theoretically forever. Given the typical lifetime until failure of five years for wind turbine gearboxes, it could be hypothesized that the simple approach of this repetitive bending theory for this application in this experiment from a fatigue point of view would not primarily contribute to a failure of a gearbox part. Further research is however planned to be conducted to investigate, if such behaviour of the main shaft also exists in larger scale wind turbines with similar main shaft and bearing design, and if the motion pattern, displacements and cycle numbers are similar.

To optically obtain an impression of the movement of the bearings, and hence the main shaft as well as the hub of the rotor, footage has been recorded with a high speed camera.

Figure **13** shows a sequence of pictures of the motion of one front bearing block from its maximum point of distance to its minimum point of distance to the front tip of a proximity probe. The single line, visible in the picture is fixed onto a distinctive dot on the probe front. The circle has been inserted as an object being fixed at the single line and reaching close to the surface of the block that is connected with the moving bearing holder. The radius of the circle is constant throughout the sequence. The square has been aligned with features of the block and is shifting with the movement relative to the circle. Since the displacements are rather small, it is hard, if not impossible to make out any change in the distance between the circle and the square, here in this picture, but if the sequence is switched through in rapid succession on a computer screen, the movement is obvious.

5. CONCLUSION

An introduction has been given into research investigating the cause for premature wind turbine gearbox failures. At the current stage an experimental wind turbine test rig set up exists in a laboratory. The wind flow environment, created by a fan has been quantified by measurements and roughly defined. Turbulences in the flow, which are assumed to be the cause for the gearbox damage have been considered by assuming unequal aerodynamic forces on rotor blades, which create a resulting fluctuating moment at the rotor hub. It is intended to artificially increase the intensity of turbulences by objects being placed in the wind flow and record and analyse the effect on the test rig.

Until now, measurements have been conducted with proximity probes to measure the displacement of the main shaft at bearing positions due to the resulting moment at the hub. It has been found that a dominant repetitive bending movement in the vertical plane exists. Two hypotheses have been stated as to what effect this bending could have on the first bearing of the gearbox and the gearbox wall. In both cases, the effect of the motion of the main shaft is suspected to have a long term damaging effect inside the gearbox. In further investigations it is planned to simulate a gearbox with suitable software and use the real life data, obtained in these experiments, as simulated input motion at the gearbox entrance. Also experimental measurements of displacements, bending angles and stresses at the interface between the main shaft and gearbox wall on the test rig are anticipated.

A further investigation has revealed by using simple fatigue theory, that the repetitive bending for certain ranges of displacements reaches critical numbers long before the gearbox lifetime of five years. This does not exclude that this motion is not contributing to the failure but a direct link to a critical number close to five years has not been established.

In the experiment a high speed camera was used to optically capture the motion of the main shaft. This device is intended to be used further in the project to verify test data as far as it is possible and to investigate and make visible any displacements that occur. In particular when the intensity of the turbulent flow is increased it is hoped to see more obvious effects on the components.

As this paper is only focusing on the main shaft movement which was the first goal of this research, there are more additions of sensors planned in the experimental set up. For example, strain gauges have already been installed on all three blade roots and two encoder wheels with optical switches have been installed on the main shaft which provide further concurrent experimental data to give a broader picture of the effects that take place during operation. It is also intended to install more sophisticated blades as compared to the crude ones which are currently fitted to the hub.

As for the wind environment, it is planned to obtain a more precise picture of the flow pattern downstream of the wind source and straightener device as well as a more sophisticated means of quantifying turbulent flow. Where suitable, flow simulation software will be used to aid in confirming the experimental values gathered.

Multiple variations of factors such as mentioned above are planned with the aim to better understand causes of wind turbine gearbox failures. At the moment the project looks at the aspects of aerodynamics, vibration and gear and bearing wear from a more distant vintage point and rather investigates how these three areas have a combined effect onto the system. Where deemed suitable, a closer look into certain components and under certain operating conditions will be performed in future experiments and subprojects.

REFERENCES

Musial, W., and Butterfield, S., McNiff, B. (2007). *Improving Wind Turbine Gearbox Reliability Analysis of Electronic Prognostics.* European Wind Energy Conference Milan, Italy, May 7-10, 2007. http://www.citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.8227&rep=rep1&type=pdf (accessed June 28, 2010)

Heller, A., (2010), Science & Technology Review *Extracting More Power from the Wind*. April/May 2010. https://str.llnl.gov/AprMay10/ tmirocha.hml (accessed October 6, 2010)

Van Rensselar, J. (2010), Tribology & Lubrication Technology. *The Elephant*. . http://www.stle.org/assets/news/document/Cover_Story_06-10.pdf (accessed October, 2010)

Moser, B., (2010). *Turbine Failure: Fine-tuning turbine gearbox performance.* http://social.windenergyupdate.com/industry-insight/turbine-failure-fine-tuning-turbine-gearbox-performance (accessed October 7, 2010)

Pope, A., Harper, J. J. (1966). *Low-Speed Wind Tunnel Testing*. NJ: John Wiley & Sons, Inc

Rexroth Bosch Group (2010). *Innovative Gearboxes for Wind Turbines*. http://www.mywindpowersystem.com/marketplace/wp-content/uploads/2010/04/Bosch-Rexroth-2_5mW-Gearbox-Techinical-Information-pdf.pdf (accessed December 12, 2010)

Juvinall, R. C., Marshek, K. M. (2006). *Fundamentals of MACHINE COMPONENT DESIGN*. John Wiley & Sons, Inc

Eck, B. (1973). *Fans*. Oxford, Pergamon Press Ltd.

Estimation of APU Failure Parameters Employing Linear Regression and Neural Networks

Renata M. Pascoal[1], Wlamir O. L. Vianna[2], João P. P. Gomes[3] and Roberto K. H. Galvão[4]

[1,2,3]*EMBRAER, São José dos Campos, São Paulo,12227-901, Brazil*

renata.pascoal@embraer.com.br
wlamir.vianna@embraer.com.br
joao.pordeus@embraer.com.br

[4]*ITA – Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo,12228-900, Brazil*

kawakami@ita.br

ABSTRACT

This study is concerned with the building of an appropriate model to estimate failure parameters of an Auxiliary Power Unit (APU). Linear and nonlinear models were used in order to evaluate which model is more suitable for this application. Data for model building and testing were obtained by simulating a nonlinear dynamic model of APU in Matlab/Simulink for various operating conditions to which it may be subjected to and with different levels of failure parameter degradation. Linear models were obtained by least-squares regression, whereas nonlinear models were obtained by training neural networks. The results obtained with these two models were compared. As a result, the neural network models were found to provide a better estimate of the APU failure parameters.

1. INTRODUCTION

The aviation business has grown rapidly in the last decade and the competition between operators becomes increasingly fierce. The development of new technologies to reduce costs and maximize operating profit has become the goal of the manufacturers in order to produce aircraft with competitive advantage. For this purpose, the increase in aircraft availability by means of improved maintenance techniques has become a key issue.

Nowadays aircraft maintenance is no longer a procedure merely reactive (conducted after the occurrence of a fault). Instead, it includes preventive actions (taken to avoid the occurrence of faults and based on statistics of mean time to failure of components) and tends to include more and more predictive actions (Vieira, 2008). In this last case, parameters are used to indicate the condition or state in which a system is close to the end of its useful life.

Hence concerns about systems Prognostics and Health Management (PHM) have increased among aircraft manufacturers. PHM covers the use of various techniques to evaluate the degradation state of a system through operational data analysis. Health data analysis enables optimization of maintenance activities, which reduces aircraft operational and maintenance costs and increases aircraft availability, therefore increases the operating profit of the airline.

In order to implement PHM in a system it is useful to have reasonable and representative amount of measured parameters of this system or other systems that are affected by it. However, the addition of new sensors could result in increasing costs for the manufacturer and add aircraft weight. It could also increase maintenance costs, since the number of components that might fail and require replacement would be higher.

As the aircraft operate under varying conditions of temperature, pressure and load, it is important that the PHM of an aviation system takes into account different operating conditions to which the system is subjected in order to avoid that the effects of variations in operating conditions are interpreted as system degradation.

This paper aims to determine an appropriate method to estimate values of failure parameters introduced in a nonlinear dynamic model of APU. Due to the simplicity of implementation, the linear approach was tried first. Since the results obtained from the use of linear regression did not get an acceptable accuracy, neural networks implementation was chosen in an attempt to get better results. Several levels of degradation of failure parameters are considered, as well as various operating conditions to which an APU is subjected.

Renata M. Pascoal et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The remainder of the text is organized as follows. Section 2 contains a brief description of an APU system, as well as the adopted model and the values of operating conditions and failure parameters that were used in this study. Section 3 presents the methods used to estimate failure parameters and the methodology adopted. Section 4 discusses the estimation results. Concluding remarks are presented in Section 5.

2. AUXILIARY POWER UNIT

The aircraft APU is a gas turbine whose main function is to assist with starting of the engines. In addition it is capable to provide pneumatic power to other systems, as well as electrical power through the activation of an electrical generator. In principle the APU is intended to operate on the ground, but it can be used in an emergency to run the generators in flight.

A typical APU is composed of three main elements which are compressor, combustor and turbine. It also has auxiliary components: fuel system, bleed system that controls the amount of extracted pneumatic power, gearbox and electrical generator. Control laws of the APU are performed by the Full Authority Digital Engine Control (FADEC) (Vianna *et al.*, 2011). An APU schematic diagram is presented in Figure 1.

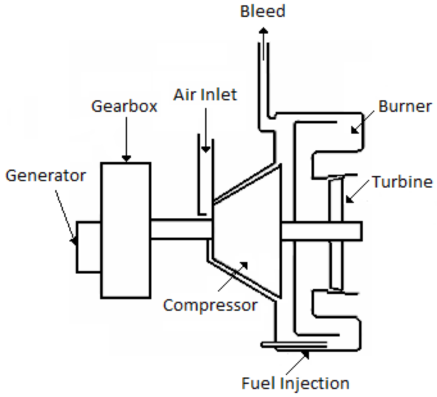


Figure 1. APU schematic diagram.

In order to provide information to FADEC, the APU has several sensors that measure shaft speed, exhaust gas temperature (EGT), bleed pressure and fuel flow.

2.1. APU Model description

The thermodynamic model of APU used in this work was developed in MATLAB/Simulink. The model consists of blocks that model the behavior of each physical component of a real APU. A schematic representation of this model incorporating blocks of three major components, compressor, combustor and turbine, plus two others that model the dynamic of the shaft and the control system is shown in Figure 2. The APU model used in this work is owned by Embraer and content rights are owned by the supplier. Then details about equations and methods related to model construction and faults modeling cannot be shown in this paper.

The three model outputs (EGT, bleed pressure and fuel flow) correspond to sensed values of a real system. The model has inputs for environmental conditions, temperature and pressure, which influence the system behavior. It also has an input for shaft power representing the power extracted by the electrical generator.

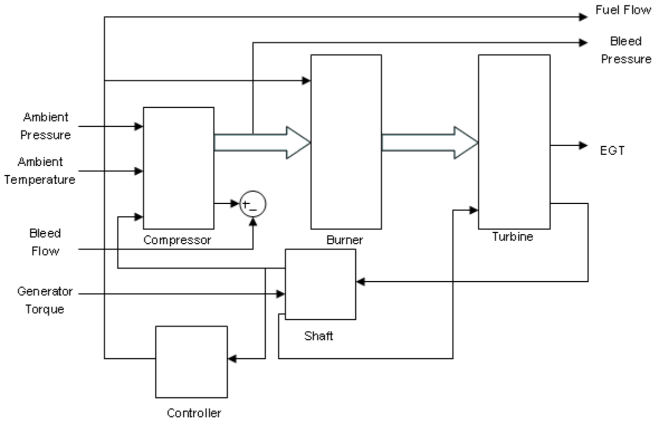


Figure 2. APU model block diagram.

In a real APU, the compressor is the unit that provides compressed air to the combustor. Its performance is defined by parameters such as pressure ratio (ratio between the output pressure and inlet pressure), air flow rate and total adiabatic efficiency, which represents the degree of deviation of the actual compression process in the compressor from a reversible adiabatic compression process.

The compressor block of APU model has functions implemented using maps obtained from charts similar to those shown in Figure 3. These functions take input parameters such as ambient pressure, ambient temperature and shaft speed to provide torque for the compressor, air flow, pressure and temperature at the compressor outlet as outputs.

The functions implemented in the burner block receive the values of pressure, temperature and flow rate of the input air from the compressor block and the fuel flow rate from the controller, and calculate the combustor outlet pressure, temperature and flow rate. The fuel-air ratio (FAR) is also an output of this block.

The turbine block has functions based on maps obtained from charts similar to those shown in Figure 4. These functions take as inputs shaft speed and the corrected value of the air flow, and provide as outputs the pressure ratio, the torque of the turbine and the exhaust gas temperature.

The block that models the control system consists of a PID (proportional-integral-derivative) controller which controls the shaft speed by measuring the APU fuel flow. The block modeling shaft dynamic calculates the resulting shaft acceleration. The value of this acceleration, which depends on the value of inertia of the shaft, is integrated over time to obtain the shaft speed that is the output of this block.

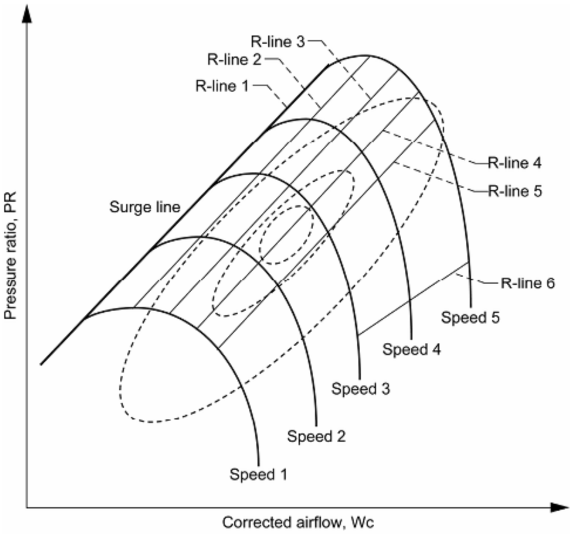


Figure 3. Notional Pressure Ratio Compressor Map (Jones, 2003).

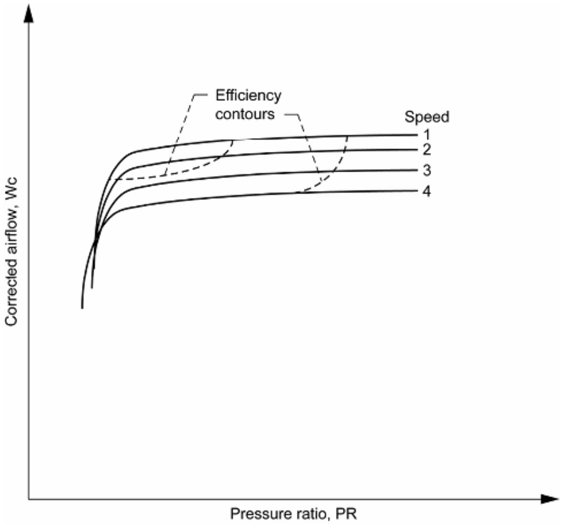


Figure 4. Notional Pressure Ratio Turbine Map (Jones, 2003).

2.2. APU failure parameters

Since an APU is a complex machine made up of several components, various failure modes can occur throughout its life cycle. This work intends to estimate the following three APU failure parameters:

1. Excessive bleed;
2. Compressor efficiency loss;
3. Turbine efficiency loss.

The choice of these failure parameters was based on the fact that their occurrences are commonplace in real APU systems.

2.3. APU Model data acquisition

Operating conditions have a direct influence on the outputs of the APU model. So they must be taken into account in the identification of fault conditions for a satisfactory APU health monitoring. Thus, data for implementing the methods to estimate the failure parameters were obtained through model simulations considering variations in operating conditions and in the degradation of failure parameters with the values specified in Table 1 and Table 2. These values are based on authors' field experience and they cover typical ranges.

Table 1. Values of operating conditions.

Operating Condition	Values
Ambient temperature (ºC)	0, 20 and 40
Ambient pressure (kPa) at sea level, 5000ft and 10000ft	101.35, 85.81 and 70.26
Shaft power required (kJ/s)	0, 22.37, 44.74 and 67.11
Bleed flow extracted (kg/s)	0, 0.189, 0.378 and 0.605

Table 2. Values of failure parameters degradation.

Failure Parameter	Values
Excessive Bleed (kg/s)	0, 0.189, 0.378 and 0.605
Compressor efficiency loss (%)	0, 1.4, 2.8, 4.2, 5.6, 7, 8.4, 9.8, 11.2, 12.6 and 14
Turbine efficiency loss (%)	0, 0.6, 1.2, 1.8, 2.4, 3, 3.6, 4.2, 4.8, 5.4 and 6

To acquire data for estimation model building and testing, three situations were considered. In all situations the operating conditions assume all the possible values from Table 1. These are the situations:

1. The only introduced failure parameter is compressor efficiency loss, which assumes all the possible values of Table 2. Other failure parameters are zero;

2. The only introduced failure parameter is turbine efficiency loss, which assumes all the possible values of Table 2. Other failure parameters are zero;
3. The only introduced failure parameter is excessive bleed, which assumes all the possible values of Table 2. Other failure parameters are zero.

In order to validate the methods, the operating condition values were chosen randomly and limited by their minimum and maximum values on Table 1. Failure parameters were also chosen randomly inside the range described above. For example, in situation 1, compressor efficiency loss values were chosen randomly between 0 and 14%, and the other failure parameters were zero.

3. Methodology of failure parameters estimation

Sections 3.1 and 3.2 describe the methods employed in this work to build linear and nonlinear models, respectively. Section 3.3 summarizes the methodology for failure parameter estimation.

3.1. Multivariable Linear Regression

Let n be the total number of observations, m the number of explanatory variables (ambient temperature, ambient pressure, shaft power required, bleed flow extracted, EGT, bleed pressure and fuel flow in this work) and z the number of dependent variables (compressor efficiency loss, turbine efficiency loss and excessive bleed in this work) in the regression procedure. The matrix of explanatory variables x_{ij} $(i = 1, 2, \dots, n, j = 1, 2, \dots, m$; with an extra column of unit values to account for the offset term in the regression) is denoted by $\boldsymbol{X}$, the matrix of dependent variables y_{ik} $(i = 1, 2, \dots, n,\ k = 1, 2, \dots, z)$ is denoted by $\boldsymbol{Y}$, the matrix of linear regression parameters to be estimated b_{lk} $(l = 1, 2, \dots, m,\ k = 1, 2, \dots, z)$ is denoted by $\boldsymbol{B}$ and the matrix of values estimated by the method $\hat{y}_{ik}$ $(i = 1, 2, \dots, n,\ k = 1, 2, \dots, z)$ is denoted by $\widehat{\boldsymbol{Y}}$. These matrices can be arranged in the following format for use in multivariable linear regression:

$$\boldsymbol{Y} = \begin{bmatrix} \hat{y}_{11} & \hat{y}_{12} & \cdots & \hat{y}_{1z} \\ \hat{y}_{21} & \hat{y}_{22} & \cdots & \hat{y}_{2z} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{n1} & \hat{y}_{n2} & \cdots & \hat{y}_{nz} \end{bmatrix} \boldsymbol{B} = \begin{bmatrix} b_{01} & b_{02} & \cdots & b_{0z} \\ b_{11} & b_{12} & \cdots & b_{1z} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mz} \end{bmatrix}$$
$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1m} \\ 1 & x_{21} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{bmatrix} \widehat{\boldsymbol{Y}} = \begin{bmatrix} \hat{y}_{11} & \hat{y}_{12} & \cdots & \hat{y}_{1z} \\ \hat{y}_{21} & \hat{y}_{22} & \cdots & \hat{y}_{2z} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{n1} & \hat{y}_{n2} & \cdots & \hat{y}_{nz} \end{bmatrix} \quad (1)$$

By using least-squares, the matrix of linear regression parameters to be estimated, $\boldsymbol{B}$, and the matrix of values estimated by the method, $\widehat{\boldsymbol{Y}}$, are obtained from

$$\boldsymbol{B} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \quad (2)$$

$$\widehat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{B} \quad (3)$$

In this work one set of the data simulated by the model as described on section 2.3 will be used to obtain B (training set) and the other set will be used to estimate failure parameters $\widehat{\mathrm{Y}}$ (test set). As a measure of estimation performance, the mean square error (MSE) between estimated and true values will be calculated for the pth failure parameter as

$$MSE(p) = \frac{\sum_{i=1}^{n}\left(y(i,p) - \boldsymbol{Y}(i,p)\right)^2}{n}, p = 1, \dots, z \quad (4)$$

3.2. Neural Networks

Artificial neural networks (ANNs) have been widely investigated for use in fault diagnosis. ANNs are trained to map inputs to outputs via nonlinear relationships in an architecture which resembles the process performed in the brain. Generally, the neural network operates in two phases: one learning phase and one operation phase. The purpose of the learning phase is to adjust the parameters of the neural network which will allow the neural network to function properly during the operation phase (Marinai, 2004).

In this work, Multi Layer Perceptron (MLP) networks with sigmoidal activation function are employed (Marinai, 2004). The adopted MLP architecture comprises three layers: input layer, hidden layer and output layer. Training is accomplished by using the well-known backpropagation algorithm, as implemented in the Neural Network Toolbox of MATLAB (version R2010a). The network inputs and outputs were defined as in the linear regression case.

3.3. Methodology for failure parameters estimation

The flowchart in Figure 5 summarizes the methodology employed in this study. It is worth noting that the data resulting from the APU simulations were divided into two separate sets for model building and validation purposes.

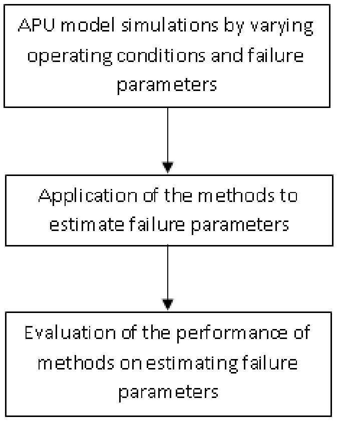

APU model simulations by varying operating conditions and failure parameters
Application of the methods to estimate failure parameters
Evaluation of the performance of methods on estimating failure parameters

Figure 5. Methodology for evaluating the performance of the estimation methods.

4. Results

4.1. Linear Regression

4.1.1. Situation 1

The MSE between the values of compressor efficiency loss estimated by linear regression and the values that were indeed used on simulation was 2.9×10^{-4}. The estimation results are depicted in the plot of estimated *versus* true values presented in Figure 6. The line in the plot represents the optimal result that would be obtained if estimated value was equal to true value.

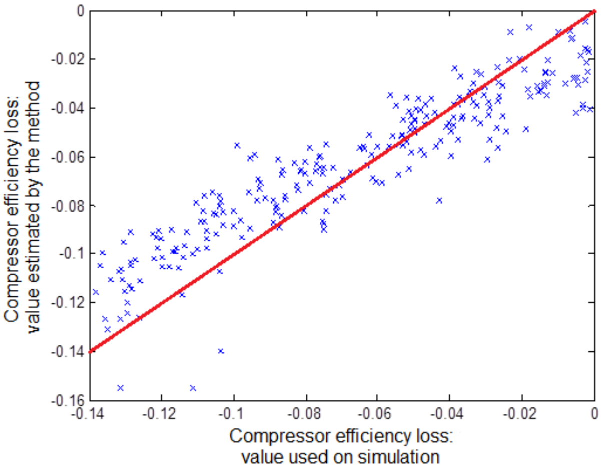


Figure 6. Compressor efficiency loss values from simulation versus values estimated by the linear regression model.

4.1.2. Situation 2

The MSE for estimation of turbine efficiency loss by the Least Squares Method was 6.9×10^{-5}. Figure 7 shows the plot of the values of simulation against the estimated values.

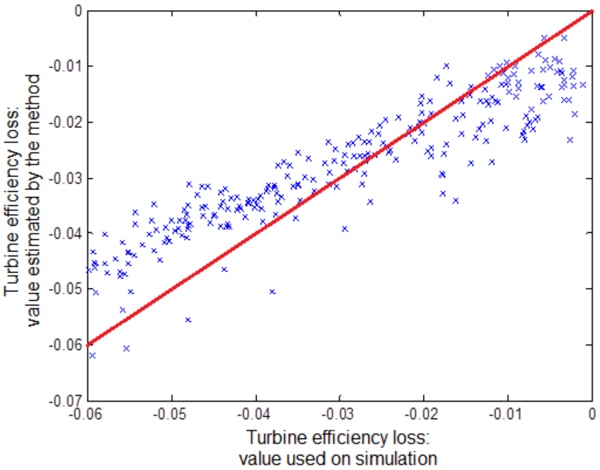


Figure 7. Turbine efficiency loss values from simulation versus values estimated by the linear regression model.

4.1.3. Situation 3

The MSE for estimation of excessive bleed by the Least Squares Method was 136. Figure 8 shows the plot of the values of simulation against the estimated values.

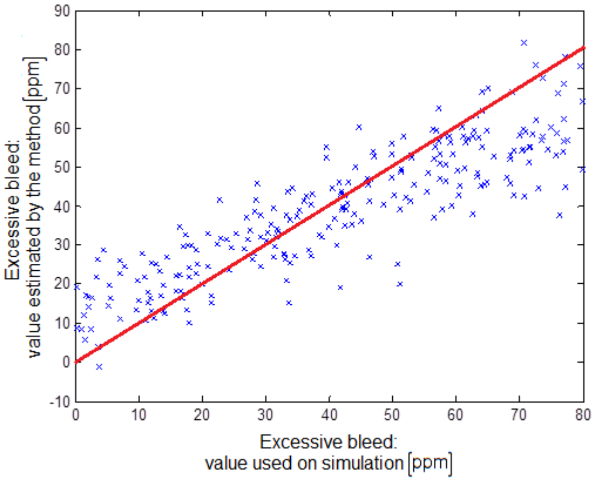


Figure 8. Excessive bleed values from simulation versus values estimated by the linear regression model.

4.2. Neural Networks

4.2.1. Situation 1

The MSE calculated for estimation of compressor efficiency loss by the neural network was 1.6×10^{-5}. Figure 9 shows the plot of the values of simulation against the estimated values.

By comparing Figure 6 with Figure 9 and the MSE values obtained by linear regression and the neural network, it is possible to notice that the neural network presented better performance on estimating compressor efficiency loss values.

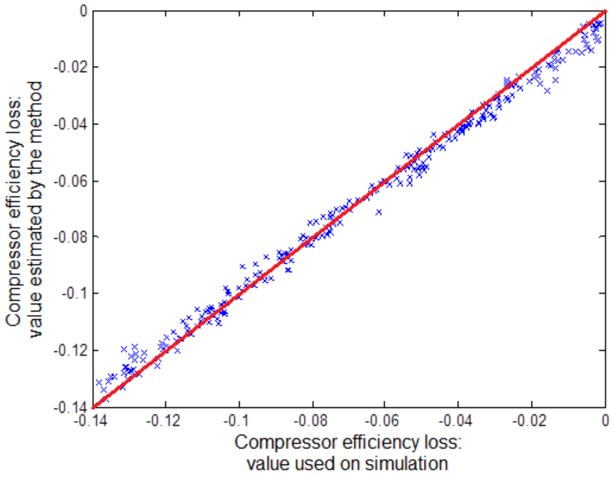


Figure 9. Compressor efficiency loss values from simulation versus values estimated by the neural network model.

4.2.2. Situation 2

The MSE for estimation of turbine efficiency loss by the Neural Network is 1.0×10^{-7}. Figure 10 shows the plot of the values of simulation against the estimated values.

By comparing Figure 7 with Figure 10 and the MSE values obtained by linear regression and the neural network, it is possible to notice that the neural network also presented better performance on estimating turbine efficiency loss values.

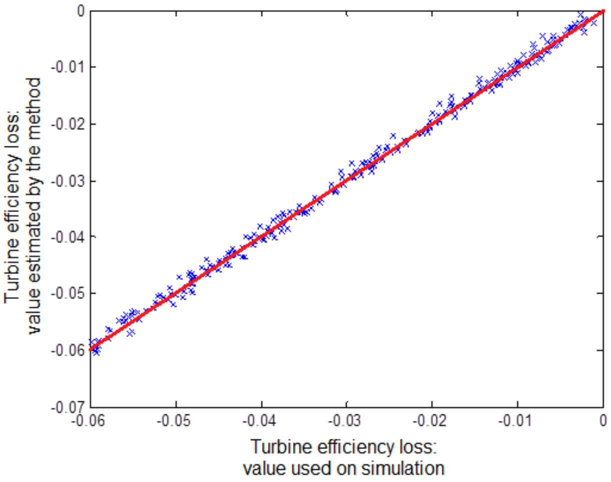


Figure 10. Turbine efficiency loss values from simulation versus values estimated by the neural network model.

4.2.3. Situation 3

The MSE for estimation of excessive bleed by the Neural Network was 1.4. Figure 11 shows the plot of the values of simulation against the estimated values.

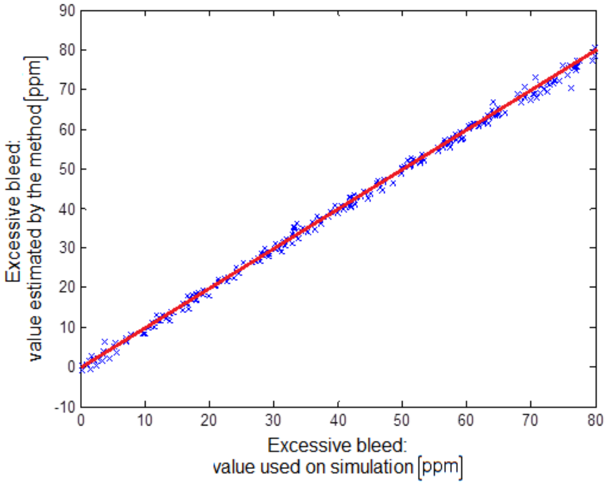


Figure 11. Excessive bleed values from simulation versus values estimated by the neural network model.

By comparing Figure 8 with Figure 11 and the MSE values obtained by linear regression and the neural network, it is possible to notice that the neural network presented better performance on estimating excessive bleed values.

4.3. Results Summary

The MSE for both methods are summarized on the table below:

Table 3. MSE summary.

	Linear Regression	Neural Network
Situation 1	2.9×10^{-4}	1.6×10^{-5}
Situation 2	6.9×10^{-5}	1.0×10^{-7}
Situation 3	136	1.4

5. Conclusion

This paper presented the results of an investigation involving the use of linear regression and neural networks for the estimation of APU failure parameters from operating conditions and measurements of EGT, bleed pressure and fuel flow. In all cases, the neural network models provided considerably better estimation results which indicates that there are nonlinearities in the relation among the monitored variables that cannot be neglected.

Future works could be concerned with extensions of this investigation to encompass the use of field data.

Acknowledgments

This work was supported by FAPESP (grant 2011/17610-0) and CNPq.

References

Beale, M. H., Hagan, M. T. & Demuth, H. B. (2013). *Neural Network Toolbox^{TM} User's Guide*. Natick: The MathWorks, Inc.

Draper, N. R. & Smith, H. (Ed. 3). (1998). *Applied Regression Analysis*. Danvers: Wiley.

Haykin, S. S. (1994). *Neural networks:A comprehensive foundation*. London: Macmillan.

Jones S. M. (2007). An Introduction to Thermodynamic Performance Analysis of Aircraft Gas Turbine Engine Cycles Using the Numerical Propulsion System Simulation Code. *NASA Glenn Research Center*, Cleveland, Ohio NASA/TM—2007-214690.

Júnior, C. L. N. & Yoneyama, T. (2000). *Inteligência Artifcial*. São Paulo: Edgar Blücher.

Marinai, L., Probert, D. & Singh, R. (2004). Prospects for Aero Gas Turbine Diagnostics: A Review. *Applied*

Energy, vol. 79, pp. 109-126. doi:10.1016/j.apenergy.2003.10.005

Vianna, W. O. L., Gomes, J. P. P., Galvão, R. K. H., Yoneyama, T., & Matsuura, J. P. (2011). Health Monitoring of an Auxiliary Power Unit Using a Classification Tree. *Proceedings of Annual Conference of the Prognostics and Health Management Society*, Montreal.

Vieira, F. M. & Bizarria C. O. (2009). Health Monitoring using Support Vector Classification on an Auxiliary Power Unit*, in Proceedings of IEEE Aerospace Conference*, Big Sky, MO.

BIOGRAPHIES

Renata Monteiro Pascoal holds a bachelor's degree on Electrical Engineering (2008) from Universidade Estadual Paulista (UNESP), Brazil, and Master Degree on Systems and Control (2010) from Instituto Tecnológico de Aeronáutica (ITA), Brazil. She is currently pursuing her Master Degree on Aeronautical Engineering from ITA. She is with Empresa Brasileira de Aeronáutica S.A (EMBRAER), São José dos Campos, SP, Brazil, since 2011. She works as a Development Engineer of Flight Control Systems at EMBRAER focused on Fly-by-Wire technology

Wlamir Olivares Loesch Vianna holds a bachelor's degree on Mechanical Engineering (2005) from Universidade de São Paulo (USP), Brazil, and Master Degree on Aeronautical Engineering (2007) from Instituto Tecnológico de Aeronáutica (ITA), Brazil. He is with Empresa Brasileira de Aeronáutica S.A (EMBRAER), São José dos Campos, SP, Brazil, since 2007. He works as a Development Engineer of a R&T group at EMBRAER focused on PHM technology applications in aeronautical systems

João Paulo Pordeus Gomes holds a bachelor's degree on Electrical Engineering (2004) from Universidade Federal do Ceará (UFC), Brazil, Master Degree on Aeronautical Engineering (2006) and doctorate on Electronic Engineering (2011) from Instituto Tecnológico de Aeronáutica (ITA), Brazil. He is with Empresa Brasileira de Aeronáutica S.A (EMBRAER), São José dos Campos, SP, Brazil, since 2006. He works as a Development Engineer of a R&T group at EMBRAER focused on PHM technology applications on aeronautical systems

Roberto Kawakami Harrop Galvão is an Associate Professor of Systems and Control at the Electronic Engineering Department of ITA. He holds a bachelor's degree in Electronic Engineering (Summa cum Laude, 1995) from Instituto Tecnológico de Aeronáutica (ITA), Brazil. He also obtained the master's (1997) and doctorate (1999) degrees in Systems and Control from the same institution. He is a Senior Member of the IEEE and an Associate Member of the Brazilian Academy of Sciences. He has published more than 150 papers in peer-reviewed journals and conference proceedings. His main areas of interest are fault diagnosis and prognosis, wavelet theory and applications, and model predictive control.

Web Based Prognostics and 24/7 Monitoring

Miryam Strautkalns[1] and Peter Robinson[2]

[1] *SGT Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA*
Miryam.Strautkalns@nasa.gov

[2] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
Peter.I.Robinson@nasa.gov

ABSTRACT

We created a general framework for analysts to store and view data in a way that removes the boundaries created by operating systems, programming languages, and proximity. With the advent of HTML5 and CSS3 with JavaScript the distribution of information is limited to only those who lack a browser. We created a framework based on the methodology: one server, one web based application. Additional benefits are increased opportunities for collaboration. Today the idea of a group in a single room is antiquated. Groups will communicate and collaborate with others from other universities, organizations, as well as other continents across times zones. There are many varieties of data gathering and condition-monitoring software available as well as companies who specialize in customizing software to individual applications. One single group will depend on multiple languages, environments, and computers to oversee recording and collaborating with one another in a single lab. The heterogeneous nature of the system creates challenges for seamless exchange of data and ideas between members. To address these limitations we designed a framework to allow users seamless accessibility to their data. Our framework was deployed using the data feed on the NASA Ames' planetary rover testbed. Our paper demonstrates the process and implementation we followed on the rover.

1. INTRODUCTION

Through observing the vast number of stove-piped data plotting tools used at NASA as well as the current practices for prognostics, diagnostics, and the concerns of their analysts, the framework for Web Based Prognostics and 24/7 Monitoring (WBPM) was created. Most projects use hard-drive based applications for data gathered during testing (e.g. Labview, ICE, C#, Java, and Matlab). Some will have options for delivering data to servers for storage. The dilemma is that the compatibility for most analysts is limited to a single platform. While most software packages provide multiple releases intended for different platforms, the known issues in the program would vary depending on the user's chosen operation system (OS). This inconsistency forces analysts to collaborate on uniform devices and software packages. The limitations change yet again depending on the length of project. Both short-term and long-term projects present their own unique needs and limitations for data acquisition and analysis. These range from reliability of storing data locally to human factors associated with long-term projects. Many analysts will plan a visit with researchers at National Aeronautics and Space Administration (NASA) Ames Research Center and other facilities to have the ability to contribute with research being conducted. Once their time has elapsed there are limited ways for them to have input and access to current data. The need to remove boundaries is of great importance in order to support increased collaboration and innovation at NASA and other institutions.

For the majority of analysts, the expectation of Hyper Text Markup Language 5 (HTML5), Cascading Style Sheets Level 3 (CSS3), JavaScript (JS), and PHP: Hypertext Preprocessor (PHP) are for the creation of static websites. In the past, this may have been the case; PHP once was an acronym for Personal Home Page. The latest updates to web-based computer languages have given power to accomplish tasks that were once only executable with languages traditionally OS based. The case study using a data feed from the NASA Ames' planetary rover testbed encompasses the majority of tasks WBPM is designed to address. The rover gathers data with the aid of one researcher overseeing the controls, it stores the information on a laptop directly on the rover, users can subscribe to the outputs, and it is considered to be conducting short term (ST) tests. These tests happen frequently providing many opportunities to test with the WBPM framework. Adjustments in rover mission profile based upon feedback

Miryam Strautkalns et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

from the data take time. The WBPM will accept data and the modifications would make the data received relevant.

The contribution of the paper is a framework that allows analysts the ability to monitor their data on any smart mobile device, computer, or laptop. Platform independent storage and retrieval of data is possible.

The paper is organized as follows. Section (2) Motivations, explains the reasons that drove this framework to be created. Section (3) The Flow of Data, explains the languages used and how the data is stored. Section (4) WBPM Framework, describes the data being monitored for the rover test bed. Section (5) Case study with NASA planetary rover test bed, overviews the observations and use of the rover test bed simulator to test the framework. Section (6) Framework Abilities, Benefits, and Future, explains the capabilities and benefits of using a web environment for data acquisition. Section (7) Conclusion, summarizes the findings made.

2. MOTIVATIONS

During most research projects, it was observed that certain restrictions were problematic to the overall quality of projects. The solution was more involvement and time on-site by analysts. More information was gathered about the common ST and long term (LT) research practices and it was realized that there was an extraordinarily large amount of physical energy expended to remain present and ensure the quality of one's project, such as remaining in close proximity to the data (e.g. sleeping onsite or at a lodge).

Concerns changed depending on the parameters of the project, though the overall concern was maintaining the results from the time spent with testing. With projects whose data acquisition was maintained on a single computer, the concern was maintaining the system. There were experiences where a day's work was lost due to the failure of the system, resulting in a loss of data collected. Live data storage on a server would have prevented any data loss experienced from relying on the computer. Many other scenarios would end on the same concern and the same solution was available to all projects: data acquisition that did not rely on the computer alone to gather and store project data.

One server, one web based application. The solution was available because of the strength web based languages had gained with the latest updates made. Once the framework as a solution was established the previous ideas of a project's process changed, and the options available had increased. The environments for collaboration had broadened; the ability to store and present live data during experiments was available with more customization. With every test more features that benefit the research process present themselves.

3. CORE TECHNOLOGIES USED

The WBPM framework is built using HTML5, CSS3, JS, PHP, and MySQL with rover test bed simulator. The following subsections explain and summarize the languages, technologies, and content used.

3.1. HTML5

To create a framework that would be malleable and easily customizable, HTML5, JS, CSS3, and PHP were used. HTML5 comes with many updates that continue what HTML4 started with its updates and contributions in changing HTML from a document-based language to a dynamic and interactive user experience. Canvas is a powerful visual tool that is available in HTML5, and with JS, the visualizations can be used to create graphing systems, charts, and visualizations that update live. Prior to HTML5, the canvas was usually defined using a Java plugin or a Flash plugin. Form validation offers a security to the quality of information collected; HTML5 validation is available within the browser.

3.2. CSS3

CSS3 is a way to organize and represent information through the browser environment in a useful and relevant structure for the content and audience. This includes the style sheet parameters required by the Canvas object. CSS3 introduces 2d and 3d transformations, transitions, and animations. These features organize the visualization of information in an effective manner for the user. Studies have shown that different categories of information are more susceptible to understanding if presented accordingly. In addition, CSS3 styles can be dynamically modified through JavaScript operating on the Document Object Model (DOM). CSS3, like its predecessors, tailor the presentation of the data depending on whether it is on a computer, smart phone, or tablet.

3.3. JS

JS is a dynamic object-oriented language used to control and moderate actions and behaviors of webpages and web-based applications. Whether it is a computer, smart phone, or tablet, JS is the language used to aid activities ranging from game play to live information feeds. HTML5's tools in combination with JS create environments ideal to handle the viewing and controlling of live content. Because of its popularity and power, JS is commonly referred to as the language of the web. JS in this work provides a way to facilitate both the visualizations of the content viewer and a portal to initiate the transmission of data. Developer groups are still performing extensive work to increase the performance of JS.

3.4. MySQL

My Structured Query Language (MySQL) is a database management system that with commands can create different layouts for information storage. While MySQL alone does not contain an interface to make these modifications, there are many user interfaces in many different languages that are created for the task. MySQL is the world's most widely used database system and can be used on any platform. A standard interface to MySQL is through PHP. Modern interfaces can also be through languages such as Python.

3.5. PHP

A server-side language, PHP is a lightweight language used for accessing and altering databases and their content. In addition to being used to work with databases, PHP can be embedded and used to create webpages with HTML. PHP executes the chosen connector for the framework data, stores the data in the MySQL database and passes the information forward as a response back to the JS and HTML in an array.

3.6. Modern Browsers

Web browser software development is far greater than all other software development combined. As of June 2012, there are approximately 2.4 billion Internet users and most of those users have access to browsers via smart phones and tablets outside of the traditional browser facilitated by a desktop or laptop computer. Since HTML5's updates, the browser environment now has the power to take on tasks thought to be limited to computer based software packages and stand alone applications. Current browsers are capable of accomplishing tasks ranging from new transitions and movement, 3d Canvas drawing, and reading motion sensor data for accelerometers, gyroscopes, and magnetometers. Development for the web helps alleviate concerns that most analysts face in keeping their software current so their data is readable and usable as hardware and software platforms evolve. This is due to the fact that the cost and person power used to fix existing bugs, create new functions, and extend the software will not be wasted on a software platform that may not be around in the future.

4. WBPM Framework - Current Rover Test Bed Data Acquisition

The NASA planetary rover currently monitors 84 parameters/second, including the statuses of twenty-four batteries, the voltage output and temperature of each, the currents, commanded speed, and actual speed for each of the four wheels, the global positioning system for latitude, longitude, and elevation, and the accelerometer, gyroscope, magnetometer readings for x, y, and z values. The output of data comes in the form of an array including the timestamp and a new updated array of values comes every second for the duration of a test. In order to observe the live tests performed while the rover is driven, the users must install Windows based communication software used to facilitate subscription.

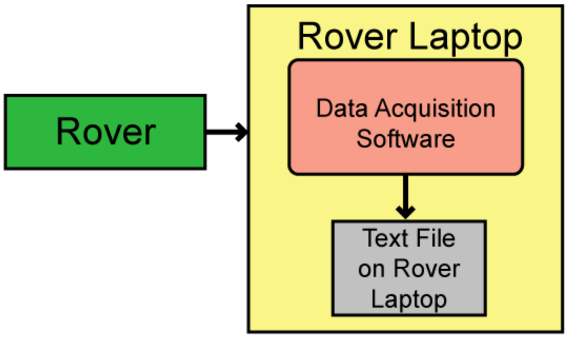


Figure 1. Current Flow chart of Rover Data.

Figure 1 shows the current sequence for Rover Data storage:

- Rover begins transmitting data.
- Local Data Acquisition Software monitors data being transmitted and sorts data into an array
- Array is saved to the rover laptop's hard drive in a text document file.

Once the text file containing rover data for the test session is saved, a researcher will upload the file to be stored on their server. Here people are able to download the file, and, using a corresponding Matlab file, can distinguish between the different values listed.

4.1. WBPM Framework

Depending on the project environment, the initialization of how the framework begins will change to accommodate different levels of security.

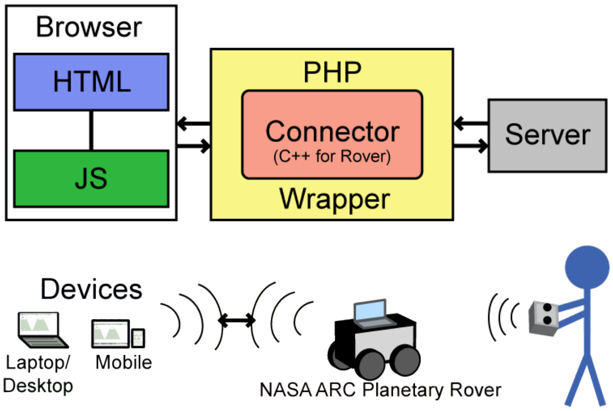


Figure 2. Framework Chart

Figure 2 shows the steps for implementing the WBPM Framework on the rover test bed, with the arrows signifying data being transmitted between languages and devices:

- Browser is initialized; the JS creates an Asynchronous JavaScript and Extensible Markup Language (AJAX) request.
- The AJAX request is created, establishes a successful connection and the PHP is opened.
- The PHP executes the connector program used to communicate data being transmitted from the project.
- The data string is received by the PHP and is split into its individual parameters and separated into an array.
- The array's data is stored by the PHP into the server's MySQL database along with other parameters including originator and access list (for security).
- When specific parameters in a time frame are requested, the information is then sorted into new arrays and sent to the JS as a response to the initial query.
- The AJAX response receives a new value for each area of the rover being monitored. Each value is added to the corresponding array.
- The array values in JS are sent to populate the HTML5 Canvas drawn graphs.
- User will be able to use common thumb gestures to scroll the data.

In real-time mode this process repeats with a set time interval. The rover receives new data every second and the AJAX is set to request data in accordance.

5. Case Study

Initial steps were to incorporate the standard methods users have been following to observe the data. After installation of the communications software there were compatibility problems encountered. The connectors and programming created for users to observe the rover's testing were written in a platform specific manner around many Windows functions – confirming one of the rationale for developing the one server, one web-based application approach. The computer being used in this case is an Apple MacBook Pro. The result was that observing the data with the same research environment, software that allow analysts to observe the rover test bed during a live experiment, was not platform-independent. This situation was a real life motivation for the WBPM framework.

The installation of the platform-dependent communication software was not an option and the solution was to recode the simulation available to be compatible. Understanding the goal is to store and retrieve data the flexibility of framework allows analysts choices on how to present and store it.

New data from the rover is transmitted every second and that dictates the cycle for AJAX requests sent. There is an empty array created within the JS and each AJAX response from the rover is added per second. The AJAX requests initiate the communication between the PHP and the Matlab rover simulation. The Matlab rover simulation provides simulated rover data. The rover data is transmitted through a Java connector embedded in Matlab in a string of values lead by a timestamp, this contains the values for each sensor on the rover for that time. The Java connector and PHP have several tasks to perform.

Java connector:

1. Initiate a connection to the MySQL database.
2. Connect with the Matlab Rover simulation to retrieve the data string.
3. Separate the string into arrays for organization.
4. Store the data into the MySQL table.

PHP Query:

1. Opened by an AJAX request.
2. Initiate a connection to the MySQL database.
3. Send a MySQL query to gather data from the database.
4. Send sorted data back as a response to the AJAX request.

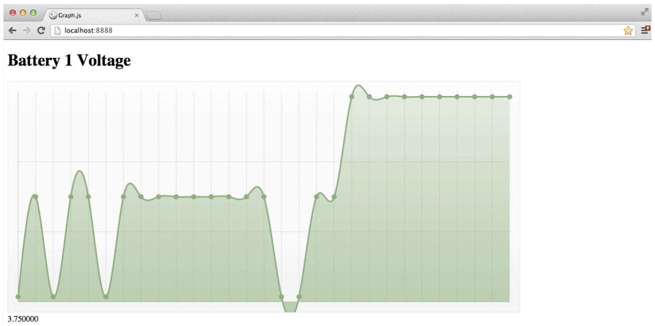


Figure 3. Entire set of rover data: 4 min 3 sec test - Battery 1 Voltage

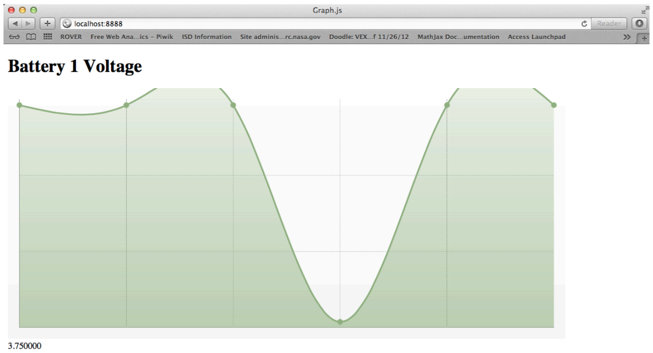


Figure 4. Six latest rover data values - Battery 1 Voltage

The Canvas graph has been programmed to show two views when selected, one for the values over the entire length of the rover test, Figure 3, or the six most recent plots of data, Figure 4.

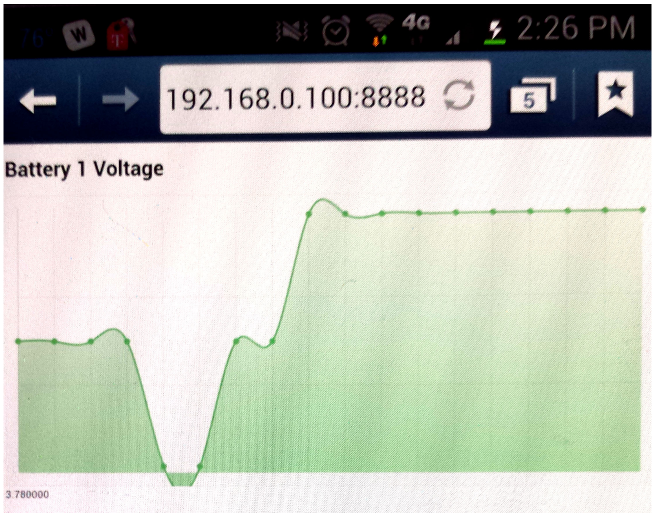


Figure 5. Smartphone test.

The WBPM framework then required mobile device testing to review future possibilities for project monitoring. A connection to be established between the simulation on localhost and a smartphone meant creating an open port for communication over a wireless connection. Figure 5 shows the mobile WBPM framework monitoring the rover's battery 1 voltage, running on a smartphone.

6. WBPM Abilities and Benefits

WBPM is providing a flexible base that can provide secure and reliable monitoring. The resources, browsers, and programming languages used have an extremely vast network providing regular updates and support. All because of the fact that the online world changes so rapidly, when engineering for the web you have to create for the future. Programming for the present in some cases is already too late. Despite the fast paced speed of innovation the support maintained for older web standards is a part of planning web applications. There are common programming practices to ensure that users with older browsers can use applications. This in combination with the browser's built-in cross OS features create an environment where analysts do not need to worry about whether collaborators have the required software installed on their computer, much less the correct OS installed on their system to support the software itself. Developing for the web helps prevent software development in directions which are not upgradable.

6.1. WBPM – Future

Having a WBPM system will only increase in relevance and strength. The capabilities to monitor in real time the performance of a project from a smartphone exists, receiving text messages when errors occur, auto generated emails upon completion of a project cycle – these are all valuable options for analysts. Not only providing a new definition for a project's environment scope, but the opportunity for an analyst to make greater use of their time outside of the scope of in person monitoring. There is no need to physically observe a task during the duration of its test. (Except for parameters for which there are no methods of measuring). The timeliness of data monitoring is not confined to a computer's storage to be viewed at a later date. With secure connections and encryptions it is a safe and cost effective solution to have instantaneous access to the data and to share that data in worldwide collaborations. Not only can you share data, but remotely adjust the settings to the project itself.

Much development of the plotting capabilities needs to be addressed. The plotting of multiple parameters at once is a challenging issue. The ability to zoom in and focus analysis on different portions of the data is also a challenge. In addition methods to perform remote commanding will also require addressing new TBD solutions.

In addition, though the central notion is one server, one web-based application, each data acquisition source will require the development of additional code to push the data to the server.

7. Conclusion

The WBPM framework creates an environment that allows remote viewing of rover data on all OSs and mobile devices. WBPM is the beginning of a cloud based monitoring and fault diagnosis application. Projects rely on computer based software packages; with the framework analysts can incorporate stability for data acquisition. The next steps for effective prognostics will be to have monitoring applications that are server based with a connector receiving data from the project source.

The WBPM resources and languages used create an environment that is ever changing and updating. By using the browser to monitor live data analysts have new options to how they conduct projects and collaborations and monitor PHM systems. The rover case study shows the benefits of data monitoring and storage when comparing the current method of stove-piped applications to the WBPM framework's method. Future extensions of the framework will only expand the number of systems that can benefit from the framework. The interactivity will increase the number of ways analysts remotely manage their data.

ACKNOWLEDGEMENT

I would like to acknowledge Adam Sweet (NASA Ames Research Center) and Sriram Narasimham (UCSC), they took the time to make their rover test bed data available for my case study. Matthew Daigle (NASA Ames Research Center) for his counsel during the creation of the WBPM framework. Peter Berg (SGT Inc. NASA Ames Research Center) for initiating the course of events that led up to this.

NOMENCLATURE

NASA	National Aeronautical and Space Administration
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
MySQL	My Structured Query Language
JS	JavaScript
OS	Operating System
LT	Long Term
ST	Short Term

REFERENCES

Flanagan, David. *JavaScript: The Definitive Guide*. O'Reilly Media, Inc., 2013. Print.

Tatroe, Kevin. Programming PHP. O'Reilly Media, Inc., 2013. Print.

E. Balaban, S. Narasimhan, M. Daigle, I. Roychoudhury, A. Sweet, C. Bond, and G. Gorospe, "Development of a Mobile Robot Test Platform and Methods for Validation of Prognostics-Enabled Decision Making Algorithms," *International Journal of Prognostics and Health Management*, vol. 4, no. 1, May 2013

Predictive Modeling of High-Bypass Turbofan Engine Deterioration

Christina Brasco, Neil Eklund, Mohak Shah, and Daniel Marthaler

General Electric Global Research, Software Center of Excellence, San Ramon, CA, 94583, USA
{brasco, eklund, shahm, marthaler}@ge.com

ABSTRACT

The deterioration of high-bypass turbofan aircraft engines is an area of study that has the potential to provide valuable information to both engine manufacturers and users. Differences in deterioration between engines corresponding to different airlines, climates or flight patterns offer insight into ideal maintenance patterns and fine-tuned estimates on engine lifetime for airlines that operate over a wide range of conditions. In this paper, a model of high-bypass turbofan aircraft engine deterioration – based on cycle frequency, air quality, relative passenger mass and climate – and its possible application as a predictor of engine health and lifetime is described. Because the quantity of interest was long-term changes in engine health, the data set was mid-flight snapshot data, grouped as a set of time-series corresponding to different engines. Ultimately, a simple model was derived which can be used to predict how long a high-bypass turbofan engine will last under given conditions. Since all of the engines used in this study were the same configuration and model, the numeric results will be most valid when predicting health of engines of that variety. However, the approach outlined here could be used for any type of engine with enough available data. The results will allow manufacturers to provide better maintenance recommendations to owners of the assets.

1. INTRODUCTION

As profit-motivated organizations, manufacturers and users of high-bypass turbofan engines should strive to use and take care of their engines in the most cost and time-efficient way possible. However, with the variation in flight patterns, and environmental conditions across airlines, continents and even aircraft, it is clear that a one-size-fits-all maintenance program will not be the best solution for all airlines using the same type of engine. Because of this, there is a need for information that allows for tailoring maintenance programs to fit the usage profile of a given airline.

Many studies of turbine engine deterioration have been performed in recent years. Some, such as the damage propagation modeling study by Saxena, Goebel, Simon and Eklund (2008), use simulated models of turbine engines to predict how they will react to different conditions. These studies are immensely helpful in determining the general character of engine deterioration. Others consider the effectiveness of different strategies for the detection of deterioration patterns or faults (Krok & Ashby, 2002; Changzheng & Yong, 2006, Weizhong & Feng, 2008).

Some of the approaches used in past studies inspired the one used here. Like Saxena et al. (2008), we took into consideration the effects of maintenance events on the deterioration pattern. However, instead of incorporating maintenance events as process noise, we attempted to identify them and use their locations as starting and stopping points in analysis.

The analysis performed here differs from these past studies in a few key ways. In using real snapshot data from engines belonging to several different airlines, we are able to consider the average effect of certain environmental conditions on a group of engines

The remainder of the paper is separated into three sections as follows. In Section 2, we outline the experimental strategy that was used to create a model of deterioration for one type of engine. This was based on the use of a trained neural network to predict Exhaust Gas Temperature (EGT), an indicator of engine health, and the analysis of changes in EGT over time for several different engines. We also comment on the assumptions made in the process of performing this analysis and the motivation behind them. In Section 3, we first describe the general trend that was observed in the data as a means of characterizing the deterioration of high-bypass turbofan engines. Then, we discuss the observed relationships between flight conditions – cycle frequency, environment, passenger load and air quality – and consider two different sets of airlines – grouped by climate – as case studies. Finally, we summarize the main results in Section 4 and outline several possible uses of this information for engine manufacturers and airlines along with the shortfalls of this experiment and

Christina Brasco et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ways in which it could be improved with the addition of more data.

2. Experimental Setup

2.1. Data Set

For the purposes of this analysis, the data set of interest was snapshot flight data from one type and configuration of high-bypass turbofan engines recorded over three years. Before training or testing the neural network, the data was preprocessed in two different steps. First, data points with one or more missing values were removed from the set. Next, the distribution of each variable was considered separately and, as each resembled a normal distribution, the tails of these distributions (mostly extreme outliers) were discarded to reduce the variance of the entire data set. As an example, the distribution of ambient temperature from the primed data set is shown in Figure 1. Finally, the testing and training data sets were assembled in different ways from the primed data set.

2.2. Neural Network Setup

In this analysis, a trained neural network was used to predict EGT given five different inputs at the time that the snapshot was recorded – Bleed Ratio, Mach number (ratio of airplane speed to the speed of sound through air), ambient temperature, N1 (the percentage of maximum fan speed of the engine, directly related to throttle setting) and altitude. An artificial neural network like this one is modeled after a biological neural network: with several hidden elements (called nodes) and weights assigned to the connections between input, hidden and output nodes. Each hidden and output node has an activation function associated with it, through which an appropriately weighted sum is passed to determine the output of the node. Because of the complex interior structure of a neural network, it has the ability to be trained to accurately predict an output given a series of inputs for arbitrarily complex functions (Jain, Mao and Mohiuddin, 1996). This quality makes a neural network an ideal choice for approximating our unknown function of EGT based on several inputs.

The training of the neural network takes place in two steps. The first step, feed-forward, involves sending the inputs for a given data point through the activation functions at the various levels. Then, in back-propagation, the different sets of weights are adjusted based on the derivative of the activation function, values of the weights and error in the output for the given data point (Jain et al., 1996). The neural network used here contained one hidden layer with five nodes. There was a sigmoid activation function from the input layer to the hidden layer and from the hidden to the output layer with adjustable weights at each step.

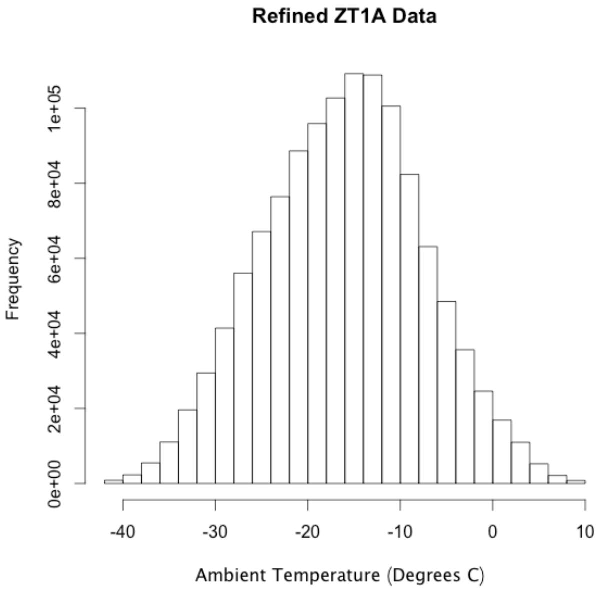


Figure 1: Histogram of ambient temperature (degrees C) of primed data set

The training data for the neural network was created by averaging subsets of points in the snapshot flight data. This was done to create a training data set that was completely separate from the testing set and to reduce the variance in the training data set. To do so, limits of between five and fourteen bins were set for each input variable such that each bin contained a non-negligible number of points. Then, the entire data set was divided into 5-dimensional hypercubes bounded on each side by a bin from one input parameter. All of the points contained in one such hypercube were averaged to create a single point in the training data set. Only points from hypercubes containing one hundred or more original data points were kept.

In deciding how the neural network should be tested and how the output should be viewed, it was necessary to consider how the deterioration of high-bypass turbofan engines would appear. For these engines, EGT is considered to be an indicator of the engine's health. The EGT margin is defined as the amount that the EGT is below the allowable limit for a given stage in the flight. When an engine is new, its EGT margin is at its highest. Over time, it shrinks until the engine must be retired.

Because none of the five input variables used were time or health-dependent, it was inferred that the neural network would not be sensitive to changes in an engine's health. Thus, residual EGT – the difference between the predicted and actual EGT – for a particular engine should change with time as the engine deteriorates or has maintenance performed on it. Based on this information, the network was tested for one engine at a time and residual EGT was recorded for each data point. Additionally, it was determined that an increase in residual EGT would be equivalent to a decrease in EGT margin. Therefore, the speed at which residual EGT changes for a given engine should indicate how quickly the health of the engine deteriorates as a whole.

2.3. Regression Analysis

After collecting time-dependent residuals for a given engine, these residuals needed to be analyzed in order to pinpoint the differences between engines. The approach for such an analysis was determined by observing similarities between several graphs of residuals vs. time.

As demonstrated in one such graph in Figure 2, residual EGT tends to increase with time, as expected, until there are sudden shifts in the graph. At these points, residual EGT decreases by a few degrees Celsius before continuing to follow the same upward trend as before. These jumps downward indicated maintenance had been performed on the engine.

We decided that a regression analysis should stop at these points – not go through them – because they interrupt the trend. Once this decision was made, it remained to devise a method of finding these jumps. In the absence of maintenance data, two different strategies were used. First, groundings for an extended period of time – more than five days – were assumed to be maintenance events. Since each data point corresponds to one flight, this was a simple matter of finding all pairs of points separated by five days or more. This length of time allowed for planes to be grounded for weather or other non-maintenance reasons such as the temporary closure of an airport. Next, downward jumps in the data were detected using criteria similar to that used in visual identification. That is, we identified times when a local max was closely followed by a local min at both the top and bottom of the band of residual EGT, giving the appearance of a downward vertical shift like those shown in Figure 2.

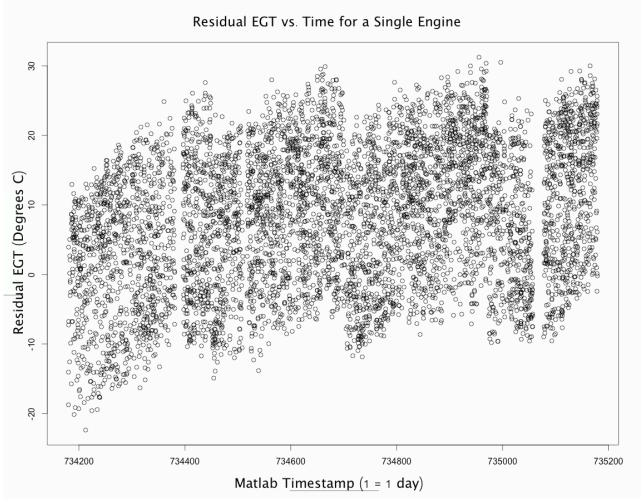


Figure 2: Residual EGT vs. Time (days, in MATLAB format) for one engine

Figure 3 shows several examples of the algorithm's success in identifying maintenance-like events. Without ground truth for maintenance event timing, the success of the jump-finding algorithm could only be judged by evaluating how often it correctly identified maintenance events compared with identification with the naked eye. Testing this method on several different graphs of residuals vs. time, we found that this method correctly identified 80 to 90 percent of the vertical discrepancies that were perceived by the naked eye to probably indicate a maintenance event. In addition, very few false positive events were identified.

Once the boundaries of the jumps were found, it remained to determine how quickly residuals changed between those boundaries. The first step in this endeavor was to cluster the data using the built-in k-means clustering algorithm (MacQueen, 1967). K-means clustering partitions a set of observations into k clusters such that the sum of the errors (distance between the cluster center and points contained in the cluster) is minimized. This is done by choosing k points, assigning each data point to the closest of those k points, and calculating the new average of each of the k sets of points. This is repeated until the centers of the clusters no longer move (MacQueen, 1967). There are many different methods that can be used to find an ideal number of clusters, k, although it has been noted that there is not necessarily a unique best value (Sugar & James, 2003) In light of this, we chose the number of clusters by performing k-means on several different time-series of residual EGT and noting how many centers would effectively cut down the noise in the data – likely due to differences in variables for which we did not account – while still demonstrating the moving trend. We found that, for this data set, approximately one center per 150 data points provides a good compromise.

An example of the effects of the k-means clustering algorithm is shown in Figure 4, which contains a plot of the original residual EGT output for a single engine, alongside the points obtained by the clustering algorithm run over the output data set. In both plots, vertical lines mark maintenance. Figure 4 shows that the resulting set of clustered points does indeed serve as a good approximation of the original data set while making performing regressions simpler. Between each set of maintenances jumps, the EGT changes in a predominantly linear fashion and the net trend is similar to those in the original data.

Next, the data was smoothed using an exponential smoothing algorithm with a small smoothing coefficient (Ostertagova & Ostertag, 2012). This technique was employed to bring potentially noisy data points just slightly closer to a perceived trend line, again to improve the

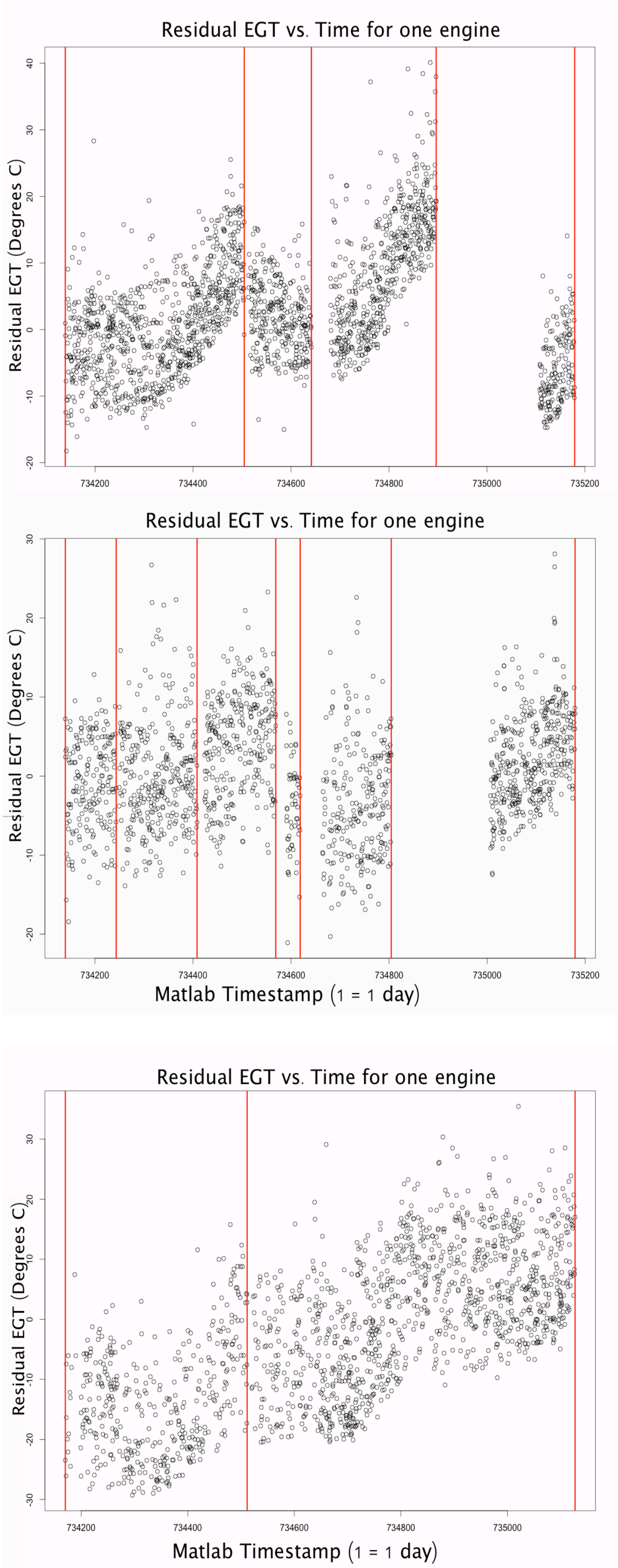


Figure 3: Samples of jumps found (marked with vertical lines) using identification criteria on graphs with slightly different shapes

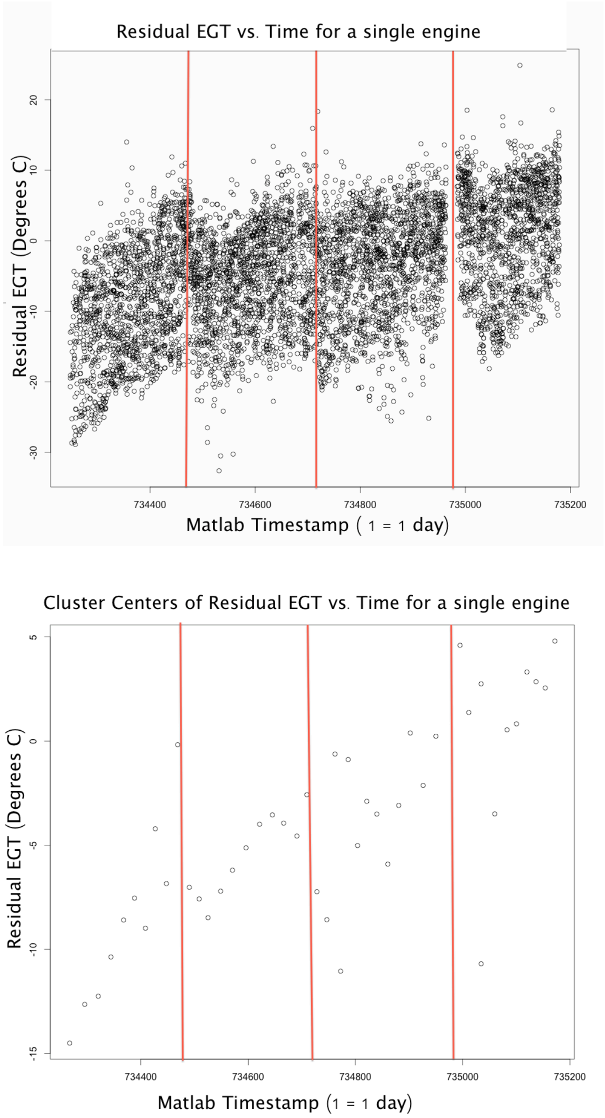


Figure 4: Residual EGT (degrees C) vs. time (days, in MATLAB format) for a single engine before (top) and after (bottom) clustering

accuracy of regression. Finally, three fits were made of the data: exponential (ln(residual) vs. time), quadratic (square root of residual vs. time) and linear. All of the appropriate equation shifts and coefficients were recorded along with the correlation coefficients. Later, this information was used to determine the best type of model for deterioration as a function of time.

3. Results

3.1. Characterizing Deterioration

Bearing in mind the ultimate goal of quantifying deterioration and engine usable life as they differ based on environmental factors and flight characteristics, the type of deterioration must first be characterized. When the groups of clustered points were analyzed, the average Pearson's r values for the three types of regressions were nearly identical over the entire set of engines, likely due to the small number of clustered points and fairly slow rate of change between maintenances. Based on this criterion, no single equation type was clearly superior. Previous work demonstrates that EGT margin deterioration rates stabilize after a period of fast initial loss and remain fairly constant until the engine needs to be removed (Ackert, 2011), it was decided that the general form of deterioration between maintenances would be

$$residual\,EGT = \alpha t + \beta \tag{1}$$

In Eq. (1), t is the number of days since the first flight in the recorded data set, and α and β are coefficients determined by a linear regression. We see from this equation that the speed of engine deterioration is determined by α, indicating that this will be the quantity of interest for this study. Further, β is understood to be the initial deterioration of the engine at time $t = 0$ Going forward, α will be referred to as the deterioration coefficient.

When increase in EGT margin through maintenance events is taken into account, Eq. (1) is not a complete description of the progression of residual EGT as a function of time. If we consider several different types of maintenances, numbered $\{1, \ldots , k\}$,which can be performed throughout the engine's lifetime, the fully general expression for residual EGT (or, equivalently, decrease in EGT margin) is

$$residual\,EGT = \alpha t + \beta - \sum_{i=1}^{k} n_i \delta_i \tag{2}$$

In Eq. (2), δ_i is the increase in EGT margin attributed to maintenance type i and n_i is the number of times that maintenance type i has been performed between the beginning of the engine's lifetime and time t.

3.2. Differences in Deterioration Coefficient

Having determined the most likely function for EGT margin deterioration as a function of time, it remains to quantify how this depends on environmental factors and cycle frequency. It must be noted that the character of deterioration may be different and less linear towards the beginning or end of an engine's life. However, the data on initial installation dates is currently unavailable and the possible time-dependency will be ignored for the purposes of this analysis as we consider a strictly linear model of deterioration.

In determining the appropriate equation for deterioration coefficient as a function of cycle frequency (*f*), the density of particles in the atmosphere near takeoff (PM10 in $\mu g/m^3$, denoted p, taken from a database of experimental PM10 values), and effective passenger mass (denoted *m* and calculated based on the number of first/business/economy class passengers on a flight), observed mathematical relationships and one physical constraint were taken into account. The limit we placed on our equation was that if *f=0*, α*=0*. That is, if an engine is never in flight, it will experience negligible or zero deterioration. As a consequence of this assumption and the observation that there was a very strong positive linear correlation between deterioration coefficient and cycle frequency, we concluded that the general form of the equation for deterioration coefficient would be

$$\alpha(f, p, m) = g(p, m)(Af) \tag{3}$$

Where $g\,(p, m)$ is an unknown function of p and m. Ultimately, we found that $g(p, m)$ was well approximated by the general form

$$g(p, m) = (Bp + C)(Dm + F) \tag{4}$$

Therefore, the complete equation for α will be

$$\alpha(f, p, m) = (Bp + C)(Dm + F)(Af) \tag{5}$$

In Eq. (5), the units of α are degrees Celsius per day. So this equation can be used to predict the lifetime of an engine in days or years assuming a constant cycle frequency. If we wish to predict the lifetime of an engine simply in the number of cycles, we may define the quantity $\gamma = \frac{\alpha}{f}$ as the deterioration coefficient in units of degrees Celsius per cycle. Then, Eq. (5) can be equivalently written as

$$\gamma(p, m) = A(Bp + C)(Dm + F) \tag{6}$$

where $B, C, D,$ and F are constants which can be determined for airlines operating in different climates.

We must note that the relevant data points here are airlines, not engines. The average deterioration coefficient and cycle frequency were found for all of the engines with a common central hub operating under the same airline. Values of p and m were taken from the PM10 data at the most common hub city and seating configuration for each airline, respectively. In determining airplane load, we only considered passenger mass because it was assumed that the cargo bay would be filled equally between planes and that the differences in overall load would come from varying numbers of passengers on the plane. This choice was motivated by the fact that there is a good deal of variance in the distributions of deterioration coefficients for an airline. However, as shown in Figure 5, these distributions have well defined peak values. Therefore, this information is the most meaningful as it applies to groups of engines with common characteristics. In this case, those belonging to a single airline.

3.3. Deterioration Coefficients in Different Climates

Having derived an appropriate equation for deterioration coefficient, it remains to show how the unknown constants vary with the climate of the main hub of these airlines. To begin, the main hub city for each airline was designated as

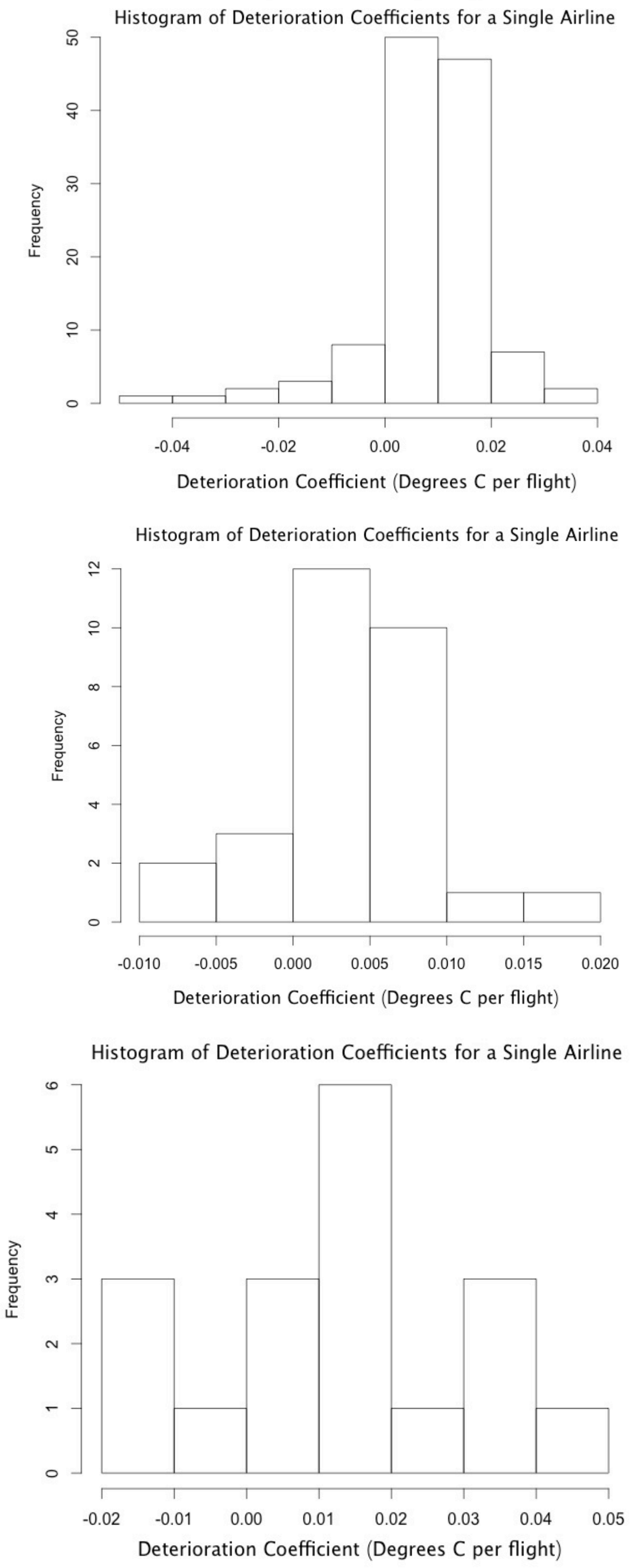


Figure 5: Distribution of deterioration coefficients for three airlines with different numbers of engines

one of five climate types - Tropical/Equatorial, Dry (arid/semiarid), Mild temperate, Continental/Microthermal or Polar – based on the Köppen-Geiger climate classification system, the most frequently-used set of climate classification criteria (Kottek, Grieser, Beck, Rudolf and Rubel, 2006). The motivation for such a classification comes from the fact that several factors – air composition, average precipitation, seasonal humidity variations – that may effect engine performance but for which in-flight data is not available, differ greatly between different locations around the globe. In the Köppen-Geiger classification system, these are accounted for and geographic locations are grouped according to the typical ranges of values exhibited for these characteristics. Grouping airlines in this way allows us to potentially reduce some of the error due to conditions we cannot quantitatively account for. Kottek et al. (2006) provide a detailed description of the criteria considered for these classifications.

Then, the coefficients for Eq. (6) were determined for engines operating in both Arid/Semiarid and Equatorial climates. The lines of the equations derived for Arid and Equatorial climates are shown in Figure 6, graphed in the form $\frac{\gamma(p,m)}{Bp+C} = A(Dm + F)$ along with the points for airlines corresponding to those climates.

For Arid climates, it was found that the equation for γ would be

$$\gamma(p,m) = (1.31 \cdot 10^{-5}p + 0.0033)(4.84 \cdot 10^{-5}m - 4.44) \tag{7}$$

And similarly, in Equatorial climates,

$$\gamma(p,m) = (-5.81 \cdot 10^{-7}p + 0.0012)(1.25 \cdot 10^{-6}m - .10) \tag{8}$$

Based on Eq. (7) and Eq. (8), we see that the dependency of γ on the different input parameters varies based on climate. In arid climates, where flight conditions are generally harsher, we see that

$$\frac{\partial\gamma}{\partial p} = (1.31 \cdot 10^{-5})(4.84 \cdot 10^{-5}m - 4.44) \tag{9}$$

$$\frac{\partial\gamma}{\partial m} = (1.31 \cdot 10^{-5}p + 0.0033)(4.84 \cdot 10^{-5}) \tag{10}$$

With values for p on the order of 10^2 and m on the order of 10^5, the values of the partial derivatives of γ with respect to p and m, respectively, are on the order of 10^{-6} and 10^{-7}.

On the other hand, in Equatorial climates

$$\frac{\partial\gamma}{\partial p} = (-5.81 \cdot 10^{-7})(1.25 \cdot 10^{-6}m - 0.10) \tag{11}$$

$$\frac{\partial\gamma}{\partial m} = (-5.81 \cdot 10^{-7}p + 0.0012)(1.25 \cdot 10^{-6}) \tag{12}$$

Using the same estimates for p and m, we see that the values of the partial derivatives of γ with respect to p and m, respectively, are on the order of 10^{-8} and 10^{-9}, almost negligible compared with those in Arid climates. So we see that in less harsh climates, deterioration coefficient is much less sensitive to changes in flight conditions.

4. Discussion and Conclusion

We see in Figure 5 that although this model of deterioration coefficient is fairly accurate, it is not perfect. Here, we ignored several possible parameters that could have affected deterioration coefficient, possibly bringing the points in Figure 5 closer to the trend line and creating a slightly better model. This was motivated both by a desire to keep the model from becoming too complicated to be useful and an absence of reliable data. A few such parameters would have been runway length or fuel efficiency - indicators of how the plane is flown differently between airlines. However, we also note that these factors may be difficult to define before the engine is put into service, making an accurate prediction of deterioration coefficient with a refined model difficult.

Despite the possibility that this model is not a perfect description of deterioration coefficient, we are now equipped with a tool that can be used to help manufacturers and users of high-bypass turbofan engines with reasonable accuracy.

First and foremost, our model allows us to make estimates on the relative lifetimes of engines under different conditions. Using the specific value of α or, equivalently, γ for a given airline, along with a pre-specified maintenance plan and initial EGT margin, we can use Eq. (2) to predict either the number of days or cycles that an engine will last on the wing of a plane. Both airlines and engine manufacturers can use this information to determine how often an engine needs to be maintained for it to reach a desired number of cycles or years of use.

Once lifetime estimates and maintenance patterns are determined for a specific engine, this information can be used in financial considerations for producers and consumers. Companies that produce or maintenance engines, knowing what the maintenance frequency will likely be, can use this information to determine how much maintenance events should cost to appropriately offset the price of producing the engine. Airlines can use this model and the resulting recommended maintenance patterns in a similar way. Knowing how much an airline will need to spend on an engine (or a set of engines in a fleet) during its usable life will allow ticket prices to be adjusted accordingly.

We see here that this model has the potential to help save both time and resources. The major shortfall of this study is that it only included a few airlines per climate type, some of which did not have data on very many engines, and that only two different climate types were considered. At the time of the study, all of the available data was used. However, with more flight data from a wider array of engine models, configurations, locations and airlines, the analysis performed here could be expanded, making it more accurate and robust.

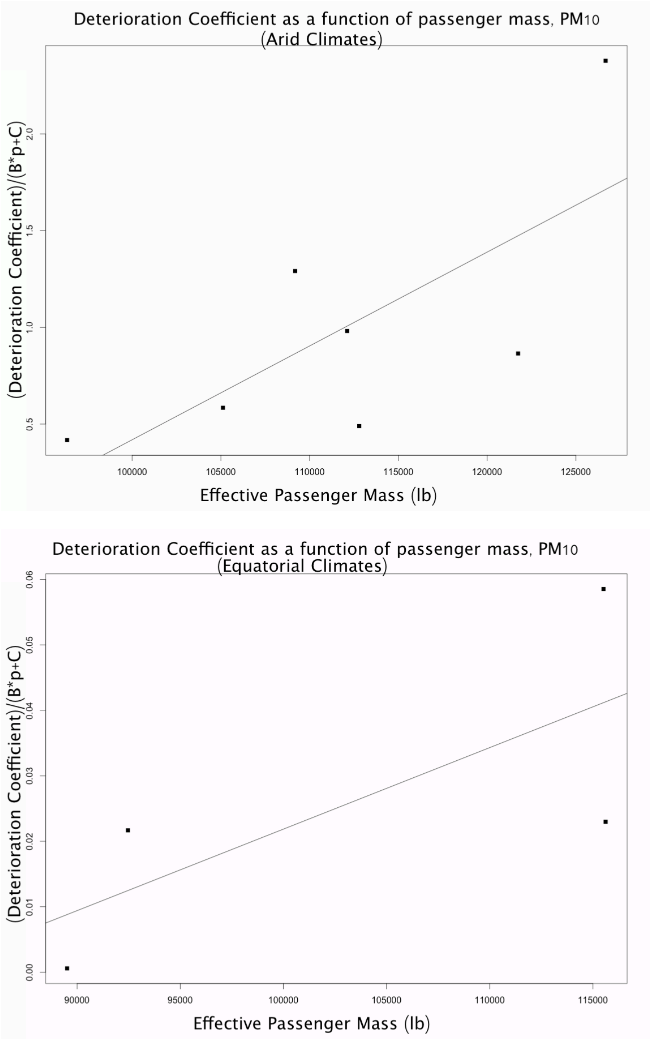


Figure 6: Functions of deterioration coefficient graphed with points for different airlines operating in Equatorial (top) and Arid (bottom) climates

References

Ackert, S. (2011). Engine Maintenance Concepts for Financiers: Elements of Turbofan Shop Maintenance Costs. *Aircraft Monitor*. http://www.aircraftmonitor.com/uploads/1/5/9/9/15993320/engine_mx_concepts_for_financiers___v2.pdf

Changzheng, L., Yong, L. (2006). Fault Diagnosis for an Airgraft Engine Based on Information Fusion. *IEEE 3rd International Conference on Mechatronics* (199-202)

Jain, A.K., Mao, J., Mohiuddin, K.M. (1996) Artificial

Neural Networks: A Tutorial. IEEE Computer 29, 31-44. http://www4.rgu.ac.uk/files/chapter3%20-%20bp.pdf

Kottek, M., Grieser, J., Beck, C., Rudolf, B., Rubel, F. (2006). World Map of the Köppen-Geiger Climate Classifiation Updated. *Meteorologische Zeitschrift.* 15 (6), 259-263.

Krok, M.J., Ashby, M.J. (2002). Condition-based, Diagnostic Gas Path Reasoning for Gas Turbine Engines. *2002 IEEE lnternational Conference on Control Applications.* September 18-20. Glasgow, Scotland,U.K.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability,* 281-297. Berkeley, CA, USA

Ostertagova, E.,Ostertag, O. (2012). Forecasting using simple exponential smoothing method. *Acta Electrotechnica et Informatica* 12 (3), 62-66. DOI: 10.2478/v10198-012-0034-2

Saxena, A., Goebel, K., Simon, D., Eklund, N., (2008). Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation. *International Conference on Prognostics and Health Management, 2008.* pp. 1–9. IEEE.

Sugar, C.A., James, G.M. (2003). Finding the number of clusters in a data set: An information theoretic approach. *Journal of the American Statistical Association.* 98 (463). pp. 750-763. DOI: 10.1198/016214503000000666

Weizhong, Y., Feng, X. (2008). Jet Engine Gas Path Fault Diagnosis Using Dynamic Fusion of Multiple Classifiers. *2008 International Joint Conference on Neural Networks (1585-1591).*

BIOGRAPHIES

Christina Brasco is currently an undergraduate at Yale University and will receive her B.S. in Physics and Mathematics in 2014. She worked as a Data Science CoE intern at General Electric's Software Center of Excellence in the summer of 2013. During the summers of 2012 and 2011, she worked as an undergraduate research assistant on the MicroBooNE Collaboration. She won Yale's Benjamin F. Barge Prize for the solution of original problems in mathematics in 2011 and was a member of the 2010 U.S. Physics Team.

Neil H. W. Eklund, Ph.D. had been a research scientist in the Machine Learning Laboratory of General Electric Global Research since 2002. He has worked on a wide variety of research projects related to remote monitoring, diagnostics, and prognostics for military and commercial aircraft engines, medical devices, wind turbines, manufacturing systems, ground-based gas turbines, and other platforms. He holds eight patents, with another eight pending, and has authored over 70 technical publications. Neil is also an Adjunct Professor of Electrical Engineering and Computer Science at Union Graduate College in Schenectady, NY since 2005. He was one of the co-founders of the PHM Society, the first Editor-in-Chief of the International Journal of Prognostics and Health Management (ijPHM) from 2009-2011, and is active in the organization of the annual PHM Society conference. He also works with the ISO to develop standards related to diagnostics and prognostics.

Mohak Shah, Ph.D. is a research scientist at GE Global Research. He is the co-author of "Evaluating Learning Algorithms: A Classification Perspective" published by Cambridge University Press. His research interests span machine learning, statistical learning theory and their application to a variety of domains. More details can be found at: http://www.mohakshah.com

Daniel E. Marthaler, Ph.D. has been a Senior research scientist in the Industrial Internet Analytics Laboratory of General Electric's Software Center of Excellence since May 2012. He holds a PhD in Mathematics from Arizona State University and has been working on problems in the realm of Artificial Intelligence and applied Machine Learning for over a decade. He has over 25 peer reviewed publications and has authored the Gaussian Process in Machine Learning (GPML) toolkit in Python.

Analysis of data quality issues in real-world industrial data

Thomas Hubauer[1], Steffen Lamparter[2], Mikhail Roshchin[3], Nina Solomakhina[4], and Stuart Watson[5]

[1,2,3]*Siemens AG, Munich, 81739, Germany*
fname.lname@siemens.com

[4]*Siemens AG, Munich, 81739, Germany*
nina.solomakhina.ext@siemens.com

[5]*Siemens Industrial Turbomachinery Ltd, Lincoln, LN5 7FD, United Kingdom*
watson.stuart@siemens.com

ABSTRACT

In large industries usage of advanced technological methods and modern equipment comes with the problem of storing, interpreting and analyzing huge amount of information. Handling information becomes more complicated and important at the same time. So, data quality is one of major challenges considering a rapid growth of information, fragmentation of information systems, incorrect data formatting and other issues. The aim of this paper is to describe industrial data processing and analytics on the real-world use case. The most crucial data quality issues are described, examined and classified in terms of Data Quality Dimensions. Factual industrial information supports and illustrates each encountered data deficiency. In addition, we describe methods for elimination data quality issues and data analysis techniques, which are applied after cleaning data procedure. In addition, an approach to address data quality problems in large-scale industrial datasets is proposed. This techniques and methods comprise several well-known techniques, which come from both worlds of mathematical logic and also statistics, improving data quality procedure and cleaning results.

1. INTRODUCTION

Caused by decreasing software cost and technological improvements, the amount of data produced, processed and stored by companies grows continuously. This data contains information regarding work process, equipment, staff involved and even more. Based on this data decisions are made, long-term plans are drawn up and statistics are compiled. Therefore, even small amounts of poor quality data may cause problems and costly consequences. Examples for complications caused by dirty information include wrong decisions, inadequate prognoses based on imperfect statistics, troublesome handling and analysis of data. Hence, data cleansing is one of the most important tasks in information technologies, especially in knowledge-based systems. The current work examines data analysis and cleaning using an example from the Siemens Energy Sector, in particular its subdivision Oil and Gas solutions. Part of its operational data is analyzed for possible data quality problems and a number of approaches to their solution are considered.

This paper is structured as follows. Firstly, an introduction to the topic of data quality and a description of related work is provided. The third section describes an industrial use case and particular data schemes. The forth section examines primary data characteristics describing its quality conditions. First subsection here comes with the list of generally defined data quality dimensions. Next, they are discussed in conjunction with our industrial scenario and illustrated with factual examples. In the fifth section there are proposed techniques and methods, which help to overcome difficulties of low data quality and make use of such information. In addition, data analysis techniques are described. The paper concludes with a summary of findings and statements of further requirements and needs for future development in data quality assessment and data cleaning for industrial data-related procedures.

2. RELATED WORK

Unsatisfactory data quality affects each field of action in both IT-related procedures and business-related tasks. Many companies elaborate their approaches to data quality assessment with respect to their own data purposes and types. Huge amounts of data, including names, addresses, numerical and categorical values have to be stored and manipulated. Towards to improvement of information quality assessment there is a number of research works

Thomas Hubauer et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

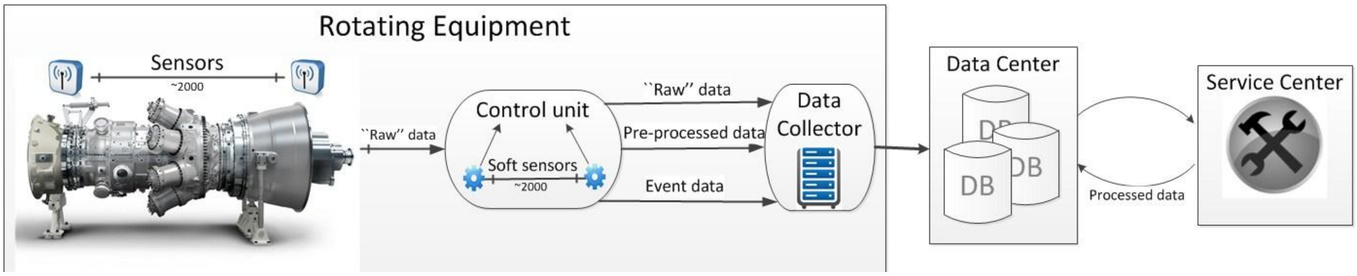


Figure 1. Appliance structure and data flow

conducted by post, insurance (Corporation & Consulting, 2011) and product trading (Pipino, Lee, & Wang, 2002) companies, criminal-record governmental system (Laudon, 1986) and many others (Wang, Strong, & Guarascio, 1996). In industry and research fields the challenge of complex data access is relevant as well: there exist a number of research works from different branches, such as industrial ecology (Weidema, B. P. & Wesnæs, 1996), healthcare industry (Safran et al., 1998; Gendron & D'Onofrio, 2001), meteorology (Foken et al., 2005), sensor networks (Wallis et al., 2007). Moreover, recently there have been launched a project "Optique" intended to improve data quality and to provide a quick end-user access to Big Data. It is conducted jointly by several European universities and two big industrial companies: Siemens AG and Statoil USA. The goals of the project are (Optique, 2012):

- to provide a semantic end-to-end connection between users and data sources;
- to enable users to rapidly formulate intuitive queries using familiar vocabularies and conceptualizations;
- to integrate data spread across multiple distributed data sources, including streaming sources;
- to exploit massive parallelism for scalability far beyond traditional RDBMSs and thus to reduce the turnaround time for information requests to minutes rather than days.

3. Case study: Industrial Data

This paper relates to data quality at Siemens Energy Sector. Data handling and processing in energy domain is becoming a big challenge, while power generation is getting more and more important in the course of time.

Siemens Energy Services maintains thousands of power generation facilities, specifically, the major core components: gas and steam turbines, called in the latter "appliances" or "rotating equipment". Operational support is provided through a global network of more than 50 service centers. These centers are in turn linked to a common database center, which stores the information coming from the appliances in several thousands databases. Further in this chapter data organization, processing and data types used in the tables are presented (see also Figure 1).

Each appliance comprises several industrial computers, which operate based upon information from sensors and serve the functions of (i) control unit and (ii) data collector. Overall approximately 2000 sensors are used to monitor the functioning of a single appliance.

The control unit serves the following functions: receiving sensor measurements, real-time monitoring of the appliance and communication of all information to a data collector. To conduct monitoring, it processes received sensor data in several ways and generates corresponding short messages ("events"), that describe the status of a unit and its functioning. There are three levels of data processing offered by a control unit:

- no processing applied at all, data remains as it was generated by hardware sensors ("raw" data);
- soft sensors: small chunks of code, which use predefined rules (i.e., thresholds, trends) in order to generate events for condition monitoring; there are usually approx. 2000 soft sensors and mostly each soft sensor is assigned to one or several hard sensors;
- simple analysis: information preprocessing, based on hard sensors' measurements and soft sensors' calculations (e.g., Fast Fourier Transformation).

The main function of a data collector is to accumulate the information, passed by control unit and to send it regularly to the central database. Below are described different types of tables stored in databases:

- **Serial numbers, identification codes and all general characteristics of unit components.** Below is shown exemplary Table 1 with such information.

  In the addition, location of the appliance, weather conditions, history of operation and conducted maintenance, performance indices are provided. Generally, information of that kind is polytypic: it contains strings, numerical data and other types.

Table 1. Main characteristics of an appliance

ID	Engine Type	Power Output	Frequency	…
T1	TurbineType1	12.90MW(e)	50/60 Hz	…
T2	TurbineType2	19.10MW(e)	50/60 Hz	…

Table 2. Measurement data

SensorID	Timestamp	Value1	Value2	…
TMP23-1	2010/07/23 23:11:55	44	49	…

- **Measurements of sensors and monitoring devices.** Table 2 depicts the schema of such data, which is often called as "raw" data, referring to the fact, that it represents unprocessed data incoming from machinery itself. Tables of that category contain mostly numerical data and have extremely large size.
- **Pre-processed data and events.** Typically data preprocessed by control unit and soft sensors is stored in different tables. Though these tables are distinguished, they have the same structure, showed below in the Table 3. Tables of this category have a huge size as well and consist mostly of text and date/time data.
- **Processed data.** In that category databases store results of analyses, conducted previously by service centre for a particular appliance. All diagnostics results based on data from central database, store in the database as well and might be used for further diagnostics.

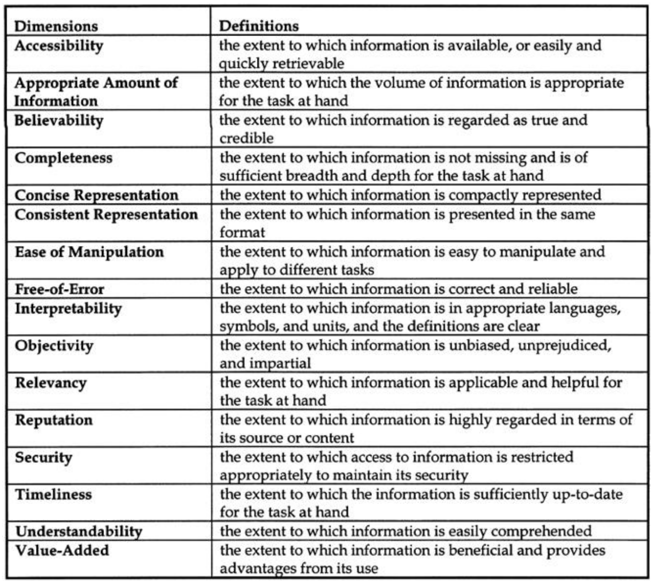
Each table has up to 20 attributes and contains various data formats, including scaled (nominal, ordinal, interval, ratio types), separated (with comma, tabulation), binary, floating-point (single, double) data types. Per a single appliance overall amount of tables exceeds 150. In sum, tabulations with sensor and event data result in 100 TB of timestamped data. Moreover, sensors continuously produce measurements at a rate between 1 and 1000 Hz and about 30 GB of a new sensor and event data are generated per day. Due to numerous causes, such as different vendors of devices or historical reasons, for a database scheme there exist more than 10 various logical schemes.

Thus, in described situation there arises a number of challenges that complicate access to information and its processing. Their overcoming requires great amount of time and resources. Further in this paper these challenges and approaches to them are discussed more thoroughly.

4. Data Quality Dimensions

In order to point out and classify the defects of data, special data characteristics have been defined. Deficient condition of any one of them has an impact on effective analysis and processing of the information. They are called *Data Quality Dimensions.*

The first part of this section lists general data characteristics used to describe data of any purpose. The second part explores dimensions of industrial data and provides some explanatory factual examples.



Dimensions	Definitions
Accessibility	the extent to which information is available, or easily and quickly retrievable
Appropriate Amount of Information	the extent to which the volume of information is appropriate for the task at hand
Believability	the extent to which information is regarded as true and credible
Completeness	the extent to which information is not missing and is of sufficient breadth and depth for the task at hand
Concise Representation	the extent to which information is compactly represented
Consistent Representation	the extent to which information is presented in the same format
Ease of Manipulation	the extent to which information is easy to manipulate and apply to different tasks
Free-of-Error	the extent to which information is correct and reliable
Interpretability	the extent to which information is in appropriate languages, symbols, and units, and the definitions are clear
Objectivity	the extent to which information is unbiased, unprejudiced, and impartial
Relevancy	the extent to which information is applicable and helpful for the task at hand
Reputation	the extent to which information is highly regarded in terms of its source or content
Security	the extent to which access to information is restricted appropriately to maintain its security
Timeliness	the extent to which the information is sufficiently up-to-date for the task at hand
Understandability	the extent to which information is easily comprehended
Value-Added	the extent to which information is beneficial and provides advantages from its use

Figure 2. Data Quality Dimensions

4.1. Main characteristics of a data

Overall there are 16 typical data quality dimensions describing data features (Kahn, Strong, & Wang, 2002) as listed in Figure 2.

Typically, classification of dimensions slightly differs depending on the purpose of information and used data types. From time to time some dimensions are omitted and others are split up to several more concrete attributes. The reason is that in various fields of actions some particular characteristics are more important and more attention is paid to them. For instance, for military government information security is a major feature, whereas for postal services complete and free-of-errors address database is more of a priority. For easier prioritizing and handling data quality issues, dimensions can be clustered in three *hyperdimensions* (Karr, Sanil, & Banks, 2006):

- **Process:** characteristics related to a maintenance of data, such as Ease of Manipulation, Value-Added, Security.
- **Data:** characteristics of the information itself, such as Believability, Completeness, Free of Error, Objectivity, Relevancy.
- **User:** characteristics related to usage and interaction with users, such as Appropriate Amount of Information, Accessibility, Timeliness, Understandability.

Nevertheless, all above-listed data attributes are important for databases of any purpose and there exist different techniques and methods to estimate them and correct existing data to improve its attributes.

Table 3. Processed data

ApplianceID	Time	Class	ErrorCode	Downtime	…
XX476	2010/07/23 21:10:35	Warning	OilTemperatureHigh	00:00:05	…

To obtain acceptable data quality however often requires a lot of time and resources and at times even manual correction to ensure cleanliness of data.

From now on we focus on industrial data and its significant data quality attributes, in particular in the domain of energy solutions.

4.2. Data Quality Dimensions in industry

In this section we analyze the quality of real industrial data based on (a relevant subset of) the data quality dimensions defined previously. One of the tools used during this project for analyzing data in the Siemens database and exploring its quality is the "Diagnostics of rotating equipment" software. Its main features include:

- loading from a database sensor and event data corresponding to one particular or several appliances, components or devices during a certain time period;
- visualization of data using tables and graphs;
- analyzing sensor signals by means of statistical methods;
- identifying patterns in event data i.e., revealing regularities preceding occurrences of a particular event.

In the following we give concrete examples of Data Quality Dimensions presented in Section 4.1. In order to illustrate relevancy of data quality problems there are used thermocouples measurements monitoring functioning of a gas turbine.

Completeness, accessibility

The fullness of information i.e., the fact that data is not missing and sufficiently detailed, is the most important characteristic of a data. Nevertheless, data loss is not uncommon in industry for several reasons. These reasons include the inability to access the required data: the appliance might be located in a remote region and due to a bad (or absent) connection between the data collector in the unit and the main database, the information may be unavailable. Another reason is device faults. Depending on causes, there might be absent only one type of data tables: "raw" or event data, and in that case it is still possible to make use of available information in order to conduct an analysis. More severely is the case that no data for a particular period is available at all. Figure 3 depicts loss of sensor measurements whereas Figure 4 shows absence of event pre-processed data for a week between 20th and 26th of September.

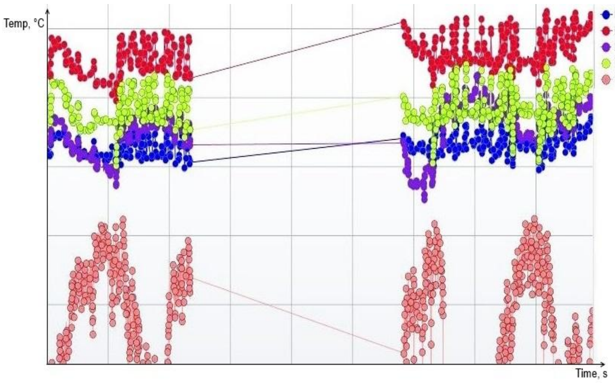
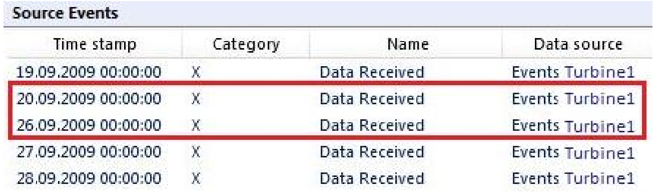


Figure 3. Signal data loss


Source Events

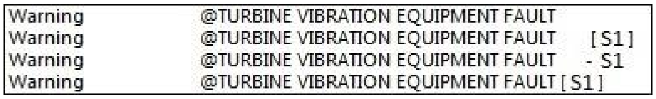
Time stamp	Category	Name	Data source
19.09.2009 00:00:00	X	Data Received	Events Turbine1
20.09.2009 00:00:00	X	Data Received	Events Turbine1
26.09.2009 00:00:00	X	Data Received	Events Turbine1
27.09.2009 00:00:00	X	Data Received	Events Turbine1
28.09.2009 00:00:00	X	Data Received	Events Turbine1

Figure 4. Event data loss

Consistent Representation

When information comes from multiple sources, it is essential to have data represented in the same format. In the current use case there exist a number of contraventions:

- various recordings of timestamps, as date and time can be written in several ways. For instance, devices of one kind write timestamps as *DD/MM/YYYY~hh:mm:ss* while another have a format *YYYY-MM-DD~hh:mm:ss* and many more of other types of devices having other date and time formats.
- data types of some information sources and monitoring devices require conversion from one format to another e.g., from *String* to *Float* or from *String* to *Integer.*
- different monitoring systems and control units indicate the same event in different ways. That happens due to diverse reasons such as various device vendors, different software versions or even location. Therefore a lack of standardization might occur and the same entities and events might be denoted differently. As shown in Figure 5, when a device S1 measurements show the failure of vibration devices, the corresponding event is denoted in several ways: different quantity of spaces between



```
Warning        @TURBINE VIBRATION EQUIPMENT FAULT
Warning        @TURBINE VIBRATION EQUIPMENT FAULT    [S1]
Warning        @TURBINE VIBRATION EQUIPMENT FAULT    - S1
Warning        @TURBINE VIBRATION EQUIPMENT FAULT [S1]
```

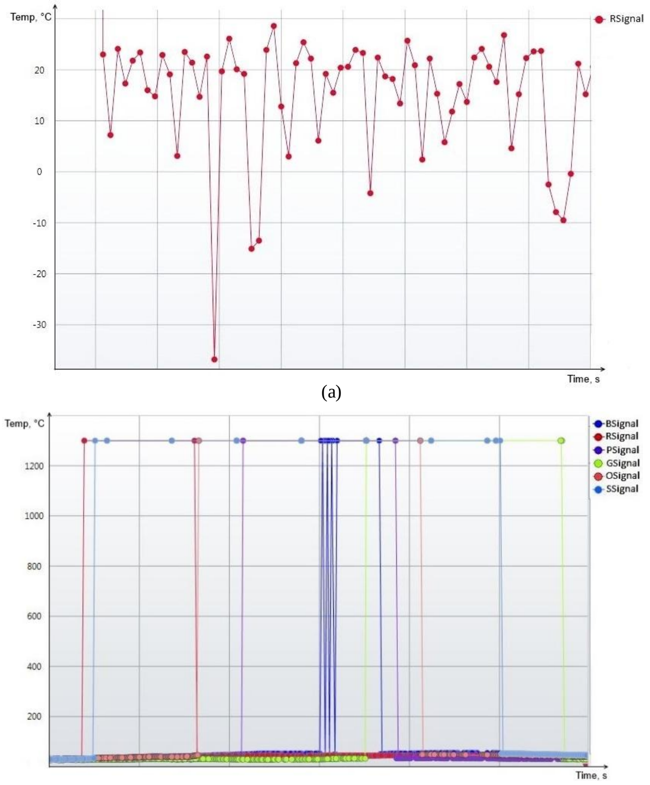
Figure 5. Different denotations of the same event

event message and sensor ID, sensor ID in parentheses etc, which badly affects analysis and statistics.

Free of Errors, Believability, Accuracy

In order to rely on results of analysis, data should be correct, precise and relevant. The possible causes of occurrence of erroneous and inaccurate data are very diverse: (i) one or several devices of the appliance faulted and gave inaccurate or wrong measurements; (ii) control unit failure occurred and there was an error during data preprocessing; (iii) there are three data transfer segments - from sensors to control unit, from control unit to data collector and between the appliance and data warehouse, for each frequencies of data transfer and speeds of data flow differ. It might happen that poor connection distorted information on one of these segments. Below are listed a few examples of discrediting data or insufficient data accuracy.

- Time Synchronization - timestamps of events and measurements incoming from several different devices might slightly differ due to such reasons as (1) time settings of a particular devices; (2) frequency and duration of data transfers between components, control unit and data collector.
- Range of values. Figure 6a shows an example where thermocouple sensor measure values are out of domain, namely minus temperatures. Additionally, occasionally outliers occur – spikes or sudden changes of value within the domain. They should be treated properly during the analysis. Figure 6b depicts an example of outliers - all sensors show alternately range maximum and minimum.
- Oscillations and noise. Figure 7a shows heavy oscillations of all signals. Figure 7b depicts the case, when signal measurements contain too much noisy data.
- Vast difference in measures. If there are several sensing elements, which duplicate each other, and they measure completely different values, then it is problematically to rely on these measurements. On a Figure 7c RSignal measurements differ from all other measurements for more than 100 degrees. In the case shown on Figure 7d, duplicating sensors measure the similar values, but as soon as temperature drops or rises, sensors measurements change with the different amplitude, as it is marked inside of black rectangles.



(a)

(b)

Figure 6. (a) Values out of range: minus temperatures. (b) Outliers: measurements of range minimum, maximum.

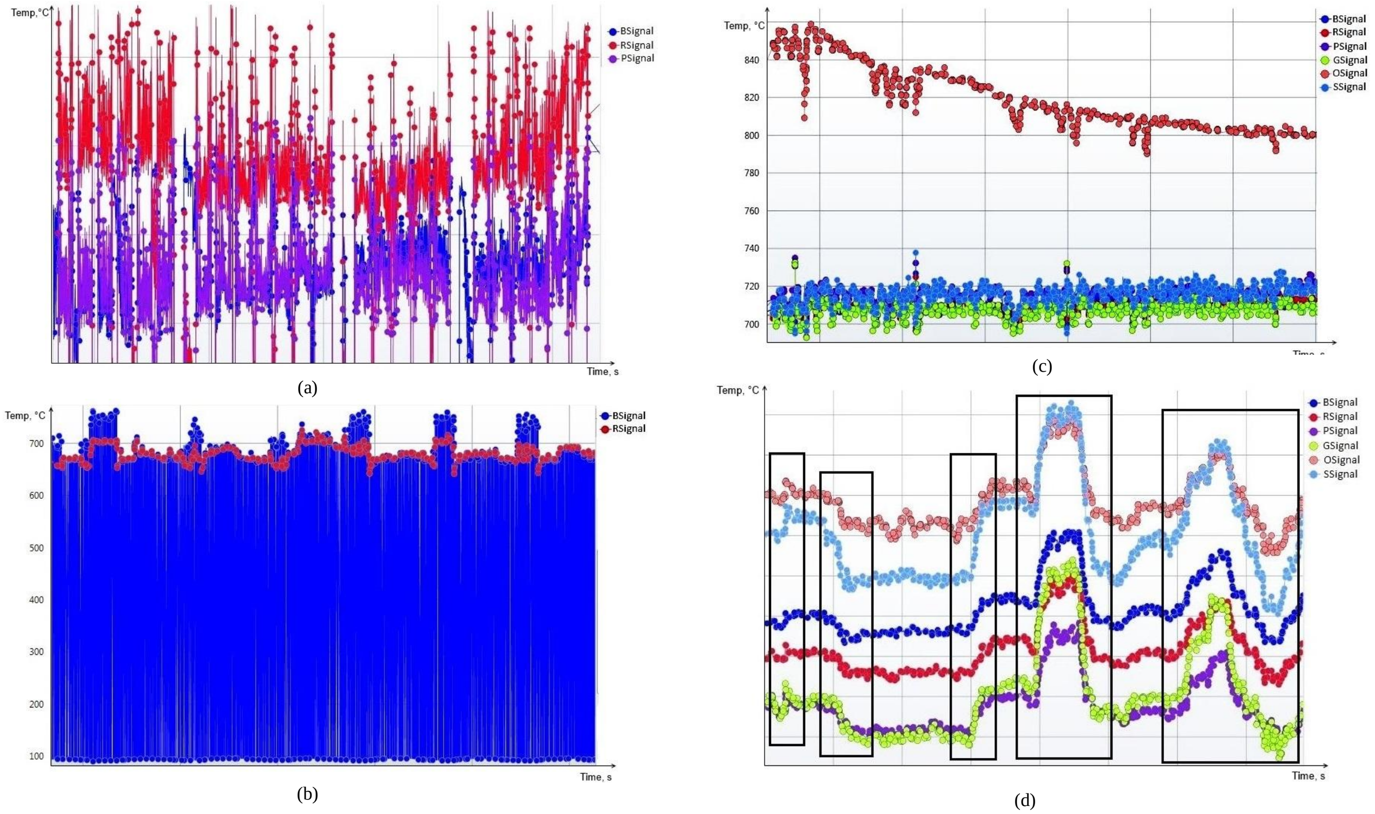
- Signal alternation. On the Figure 8 is shown the case, when two signal at some moment alternated each other and swapped their measurements, as it is also marked with the black rectangles.

Ease of Manipulation, Data Schemes

Data schemes and structures are highly heterogeneous, depended upon which technique was used to create it, which unit it belongs to, from where it comes historically. Moreover, not all foreign keys between databases are present. If information on the same entity is distributed among several sources, for instance, if information concerning a particular malfunction of an appliance should be extracted from tables “Incident Summary”, “Daily Event Log”, “Burner tip temperature” and others, the problem of missing foreign keys do not allow for easy merging of data.

Timeliness, Appropriate Amount of Information

For a thorough analysis it is critical to have all data available and updated. Though for each diagnostics case the considered time period always differs: it might be sufficient to consider only the last hour in order to identify a cause of an event, but in other cases one needs to analyze the last several years, for example to detect a deterioration of a particular component.



(a)

(c)

(b)

(d)

Figure 7. (a) Oscillatory signals. (b) Noisy data - BSignal measurements look like a white noise. (c) RSignal shows divergent values. (d) SSignal and GSignal have rise/drop amplitudes differing from other duplicating sensors.

Thus, usually data typically does not expire and become irrelevant in several years but on the other hand, has to be stored for decades.

In conclusion, for successful information analysis it is crucial to determine how reliable data is and to bring it to the representation convenient for required purposes. In the following section we discuss methods and techniques developed to achieve this goal.

5. Data Analysis

In this chapter there are examined techniques which help to get use of low quality data. Firstly, there are described data cleaning methods and in addition, a proposal to improve them is made. Data analysis techniques are described in the second part of the chapter.

5.1. Quality assessment and cleaning

There are several directions in data cleaning and existing techniques aimed at particular problems (Rahm & Do, 2000): duplicate identification and elimination, data transformations, schema matching, data mining approaches and others. Moreover there are also unified techniques. The main scientific approaches include statistical, machine learning and knowledge-based approaches. But in general any data cleaning technique should satisfy several requirements (Rahm & Do, 2000):

- should detect and remove all major errors and inconsistencies both in individual data sources and when integrating multiple sources.
- should be supported by tools to limit manual inspection and programming effort and be extensible to easily cover additional sources.
- should not be performed in isolation but together with schema-related data transformations based on comprehensive metadata.

Statistical methods are used to: (i) visualize the data; (ii) summarize and describe existing data by means of univariate and multivariate analysis; (iii) offer hypotheses and decisions with the aid of statistical tests; (iv) interpret data employing sampling techniques.

One of the most widely used statistical tools for data quality assessment is called quality indicator (Bergdahl et al., 2007). It is a measure of how well provided information meets criteria and requirements for an output quality. Also there exist a number of statistical/probabilistic techniques and its modifications (Winkler, 1999), 1-1 matching methods and bridging file technique (Winkler, 2004).

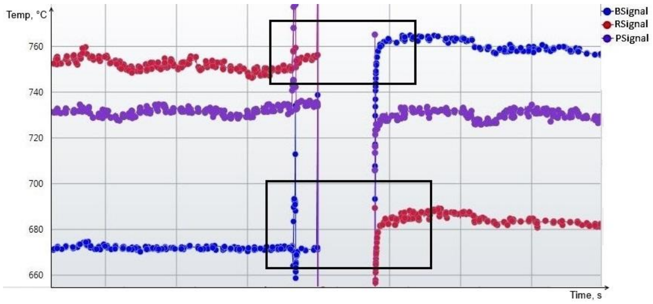


Figure 8. BSignal and RSignal traded places.

In the current use case statistical approach is widely used for detecting faults in sensor readings. For large-scale databases that enlarge day by day with new portions of sensor measurements it is highly essential to use fast and robust techniques detecting changes in signal behavior. The main approach is time-series analysis, cross- and autocorrelation, spectrum and Fourier analyses in particular. In addition, there are simple indicators and distribution tests are exploited in order to detect quickly changes in statistical parameters of sensor readings.

There exist a number of effective machine-learning algorithms. The most widely used are artificial neural networks, clustering algorithms, support vector machines, similarity learning. For a faulty sensor readings detection machine learning approach is successfully used for analysis of several sensor signals at once in order to establish confidence level for each device and thus to identify malfunctioning sensors straight away.

Another application of machine learning algorithms is duplicate elimination. For this task usually clustering and neural networks are exploited. One more technique is sorted neighbourhood method and its modifications (Bertolazzi, De Santis, & Scannapieco, 2003; Yan et al., 2007). All these methods are used in large-scale databases as well (Hernandez & Stolfo, 1995) and in the current use-case can be exploited to get rid of duplicates in event data.

For a knowledge-based approach the application domain can be represented (Batini & Scannapieca, 2006):

- procedurally in form of program code, or implicitly as patterns of activation in a neural network;
- as an explicit and declarative representation, in terms of a knowledge base, consisting of logical formulas or rules expressed in a representation language.

Typically the most general approach to perform data transformations are extensions of standard query language SQL (Rahm & Do, 2000), which allows flexible transformation step definitions, their easy reuse and supports query processing tasks.

Additionally, there are several systems developed which improve the quality of data by means of rules extracted from domain knowledge and domain-independent transformations (Batini & Scannapieca, 2006), e.g. the Intelliclean system (Lup, Lee, & Wang, 2001) aimed at efficient duplicate elimination, the Atlas technique (Tejada, Knoblock, & Minton, 2001) which allows to obtain new rules through a learning process and Clue-Based method for record matching (Buechi et al., 2003).

Table 4. Measurement data for an exemplary sensor

Sensor ID	Timestamp	Value
TMPS1	2010/08/28 13:21:55	597.2
TMPS1	2010/08/28 13:22:00	598.5
TMPS1	2010/08/28 13:22:05	599.6
TMPS1	2010/08/28 13:22:10	600.3

$$\text{TempSensor} \sqsubseteq \forall \text{hasValue.xsd:int}[\geq 0, \leq 600]$$

Figure 9. Temperature sensor measurements range represented in a model

As a proposition for a further work, we propose to combine existing techniques in order to increase productivity and effectiveness of the data cleaning process. The dataset introduced here can serve as a test. As a motivating example, consider measurements of a temperature sensor presented in a Table 4 both in a semantic model and as a statistical value.

In the model-based representation of a sensor data, such as indicated on Figure 9, after processing a measurements presented in a Table 4 the system would detect an outlier at a time 13:22:10.

On the contrary, analyzing data with statistical methods, there would be a trend detected. Thus, having available results both by model-based reasoning and statistical techniques would prevent a false alarm.

Likewise, it is useful to combine multivariate statistical analysis and machine learning algorithms such as clustering and neural networks for establishing a quality of several sensors measurements.

Therefore, that joint approach would help to improve the following weak points in managing low-quality data:

- efficient detection of data deficiency, such as (i) false positive errors and (ii) false negative errors;
- detecting correlations between particular sequences of events and their consequences and between measurements using numerous solutions, such as pattern-matching algorithms, independency tests and others; and

- model-driven correction of a model in case of changes in system structure.

5.2. Analysis and diagnostics

In this subsection there are shortly explained, how the cleaned data is studied and processed further in the current industry case. The main use is continuous diagnosing of the condition of the appliance in order to predict and prevent future faults of the machinery and to react instantly as anomalies or faults in operating are detected. Two main approaches for that are: (i) data-driven and (ii) knowledge-based techniques. Data-driven approaches includes pattern recognition, neural networks, numerical approaches; knowledge-based techniques include case descriptions, faults and correct behavior modeling. The following factors determine the choice of the appropriate diagnosing method in a particular case (ISO 13379-1, 2009):

- application and initial design of the equipment;
- availability of data to be analyzed and its complexity; and
- required qualifications of a resulting computations and models.

Brief summary of above-mentioned diagnosing techniques, presented in (ISO 13379-1, 2009):

Data-driven approach methods classify different functioning states of an appliance: normal, fault one, fault two etc. In order to achieve this, firstly the model is trained with the historical data from each condition and after that launched with the new data, which has to be classified.

The great advantage of data-driven approach is that it does not have need for a thorough knowledge of the system to be diagnosed. The other strong advantage is absence of constraints on the data type of independent variables. As a disadvantage it is worth to mention, that it might be computationally difficult to train a model, as it requires comparatively large amount of prescribed fault and non-fault states to construct a model. In addition, modelling by this approach does not result with an explanatory diagnosis.

The list of the most common data-driven techniques:

- Statistical data analysis, case-based reasoning;
- Neural networks;
- Classification trees;
- Random forests;
- Logistic regression;
- Support vector machines.

Knowledge-based approaches are used to represent knowledge using various knowledge representation techniques and reason over it to infer new knowledge. Their biggest advantage is the possibility of thorough diagnostics. There are two fundamental knowledge-based methods used by engineers:

- Fault/symptom diagnostic approach;
- Causal tree diagnostic approach.

In special situations several approaches may be combined for better results, but still both approaches are not disjoint, i.e. there are methods which might be referred to both types. However, each approach has its advantages as well as drawbacks and an engineer chooses appropriate diagnostic technique based on the type of an appliance, complexity of modeling, availability of necessary data and other factors.

6. Conclusion

For current industry use-case, data is employed to conduct calculations necessary for emergency diagnostics, prognosis of efficiency and further analysis. However, due to imperfect, incomplete or defective information and data schemes these tasks have become rather difficult to realize: wrong, missing or incorrectly formatted data may result in erroneous computations and false decisions, which can be quite disastrous for an industry processes, especially for large-scale industries.

The current paper studies data quality and different approaches to its assessment. We summarized and illustrated the most common defectiveness of a large-scaled industrial database by the example of Siemens Energy Domain and its equipment measurements. We also reviewed existing techniques that are used to overcome errors in data and proposed an approach to address data quality problems. And as shown in the examples, there is no doubt that data requires continuous control and quality improvement, although the design of a convenient technological solution to that challenge is far from trivial.

References

ISO 13379-1, I. D. (2009). *Condition monitoring and diagnostics of machines data interpretation and diagnostics techniques part 1: General guidelines.* ISO, Geneva, Switzerland.

Batini, C., & Scannapieca, M. (2006). *Data quality: concepts, methodologies and techniques.* Springer.

Bergdahl, M., Ehling, M., Elvers, E., Földesi, E., Körner, T., Kron, A., and others (2007). *Handbook on data quality assessment methods and tools*, 9–10.

Buechi, M., Borthwick, A., Winkel, A., & Goldberg, A. (2003) *ClueMaker: A Language for Approximate Record Matching*. IQ, 207-223.

Corporation, A., & Consulting, W. M. (2011). Data quality in the insurance market.

Foken, T., Göockede, M., Mauder, M., Mahrt, L., Amiro, B., & Munger, W. (2005) *Post-field data quality* *control*. Handbook of micrometeorology, Springer, 181-208.

Gendron, M. S., & D'Onofrio, M. J. (2001). Data quality in healthcare industry. *Data Quality*, 7(1), 23–31.

Kahn, B. K., Strong, D. M., & Wang, R. Y. (2002). Information quality benchmarks: product and service performance. *Communications of the ACM*, 45(4), 184–192.

Laudon, K. C. (1986). Data quality and due process in large interorganizational record systems. *Communications of the ACM*, 29(1), 4–11.

Lup Low, W., Li Lee, M., & Wang Ling, T. (2001). A knowledge-based approach for duplicate elimination in data cleaning. *Information Systems*, 26(8), 585–606.

Optique. (2012). *Optique: project description.* Retrieved November, 2012, from CVS: "http://www.optique-project.eu/ about-optique/about-optique/".

Pipino, L. L., Lee, Y.W., & Wang, R. Y. (2002). Data quality assessment. *Communications of the ACM*, 45(4), 211–218.

Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4), 3–13.

Safran, D. G., Kosinski, M., Tarlov, A. R., Rogers, W. H., Taira, D. A., Lieberman, N., & Ware, J. E. (1998). The primary care assessment survey: tests of data quality and measurement performance. *Medical care*, 36(5), 728–739.

Strong, D. M., Lee, Y. W., & Wang, R. Y. (1997). Data quality in context. *Communications of the ACM*, 40(5), 103–110.

Tejada, S., Knoblock, C. A., & Minton, S. (2001). Learning object identification rules for information integration. *Information Systems*, 26(8), 607–633.

Wand, Y., & Wang, R. Y. (1996). Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11), 86–95.

Wang, R. Y., Strong, D. M., & Guarascio, L. M. (1996). Beyond accuracy: What data quality means to data consumers. *J. of Management Information Syste*ms, 12(4), 5–33.

Winkler, W. E. (1999) *The state of record linkage and current research problems.* Statistical Research Division, US Census Bureau.

Winkler, W. E. (2004). Methods for evaluating and creating data quality. *Information Systems, Elsevier, 29*, 531-550.

Yan, S., Lee, D., Kan, M.-Y., & Giles, L. C. (2007) Adaptive sorted neighborhood methods for efficient record linkage. *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, 185-194

Learning Diagnoser and Supervision Pattern in Discrete Event System: Application to Crisis Management

Moussa Traore [1], Moamar Sayed-Mouchaweh [2], and Patrice Billaudel [3]

[1,3] *University of Reims, Champagne-Ardenne, CReSTIC, Reims, BP 1039, 51687, France*
moussa.traore@univ-reims.fr
patrice.billaudel@univ-reims.fr

[2] *University of Lille 1, F-59000 Lille, IA, Mines de Douai, 59500, France*
moamar.sayed-mouchaweh@mines-douai.fr

Abstract

The increase of natural, industrial disasters and diverse crisis has stimulated more research interest in the world. A crisis can be industrial accident, train accident, earthquake, and etc. However, the crisis management is currently an important challenge for medical service and research, to develop new technical of decision support system to guide the decision makers. Crisis management is a special type of collaboration, therefore several aspects must be considered. The more important aspect or problem in a crisis management, is the coordination (and communication) between different actors and groups involved in the management. In this paper the focus is how to handle the coordination and interaction between these different actors and groups involved in crisis management by using a finite state automaton. The representation of the crisis management as a set of couple of states and events allows to optimize the crisis management by having real time the evolution of the situation and the prediction of their evolution at their earliest.

1. INTRODUCTION

Nowadays, there has been a lot of interest in crisis management. Because, in the last years, we assist to a growing number of disasters and diverse crisis, such as the Indian Ocean tsunami 2004, the Japanese earthquake and tsunami 2011, and ect (Reuter, Heger, & Pipek, 2013). Therefore, the response to these disasters and crisis (natural or man-made) have to be fast and effective. A fast and affective response in a crisis situation allows to reduce the disaster consequences on people and the damages in nearby areas. However, the response to a crisis situation requires the collaboration between different numerous people and groups, for example police, the personnel working in the site, first aid agents, Doctors, government delegates, position of victims. The figure 1 depict a given scenario for crisis management implying various actors and groups. When a disaster occurs, the people on the accident site send information to the communication center and the latter sends rescuer teams to the accident site or affected zone as shown in figure 1 (Benkhelifa, Moussaoui, & N-Taboudjemat, 2013). In crisis management several aspects must be considered. The more important aspect in crisis management, is the coordination between the actors and groups involved in the management. The coordination between different actors involved in crisis management is fundamental to reduce the disaster consequence on victims and nearest areas.

The new challenge of crisis situation is the representation of the crisis management as a set of couple of states and events to guide decision makers. In this paper, the coordination between different actors and groups involved in crisis management is viewed as discrete model (event). Discrete Event System (*DES*) are dynamic system whose the behavior is governed by occurrence of physical events that cause abrupt changes in the state of the system (Sayed-Mouchaweh & Billaudel, 2012). Most of the last approaches of *DES* is represented by Automaton (Yunxia, 2003; Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995; Kwong & Yonge-Mallo, 2011) and Petri Net (Cabasino, Giua, & Seatzu, 2010). The figure 2 illustrates an example of modeling of a crisis management as Finite State Automaton (FSA). The purpose of modeling the crisis management as FSA is to optimize the crisis management by having real time the evolution of the situation and the prediction of their evolution at their earliest. The FSA of the crisis management is used to generate languages (or sequence of events) for diagnosis purpose. The diagnosis in crisis management is not to detect failures, but to detect the critical situations. The critical situation in figure 2 is going twice in the state "x_5" during the crisis management. The state "x_5" corresponds to the waste of time of

Moussa Traore et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the intervention team to access to the crisis site, and in crisis management the waste time is unacceptable.

Recently, the prediction of *DES* based on the sequence of events (or equivalently, a trajectory) has increasingly interest of many researchers in the last years. The prediction, in general, is the process of making a decision about a future result or evolution of a situation. In literature, a lot of prediction approaches of *DES* are presented. The authors in (Genc & Lafortune, 2009) are presented a prediction method of a possibly unobservable event in the system behavior, based on the language containing the observable events. In (Takai & Kumar, 2012), the local prognosers of *DES* exchange their observations for the sake of arriving at the prognosis decision. The prediction problem in (Xi-Rien, 1989) is a special type of projection between two languages. We present in this paper, a new approach for the prediction of *DES* and adaptive detector of supervision pattern using *FSA*, which is based on the discrete model corresponding to the crisis management.

This paper is organize as follows. In section 2, we briefly review same notation and definition of the Automaton model of *DES*. In section 3, we describe the discrete model of crisis management. The standard diagnoser for the dynamic model is shown in section 5. A general definition of discrete event dynamic system is presented in section 4. We present in section 6, the prediction of *DES*. Finally, the learning diagnoser and adaptive supervision pattern are presented in section 7.

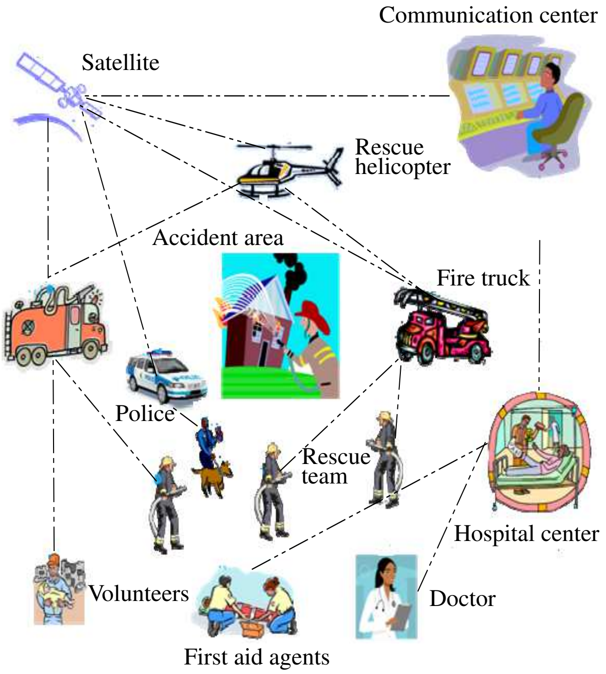


Figure 1. Emergency response scenario, (*I. Benkhelifa et al*).

2. Finite State Automaton

A Finite State Automaton can be defined as a six-tuple $M = (X, \Sigma, Y, \delta, x_0, F)$, where

- X is the set of states, Σ is the set of input events,
- Y is the nonempty finite set of outputs,
- $\delta : X \times \Sigma \longrightarrow X$ is the transition function,
- $x_0 \in X$ is the start (initial) state,
- $F \subseteq X$ is the (possibly empty) set of accepting or terminal states.

The finite set of events Σ can be partitioned in two subset, such that $\Sigma = \Sigma_o \cup \Sigma_{uo}$, where Σ_o is the observable events and Σ_{uo} is the unobservable events. A string is a finite-length sequence of events in Σ. The set of all strings formed by events in Σ is denoted by Σ^*. The set Σ^* is also called the Kleene-closure of Σ.

Further, we extend the transition function δ to $\hat{\delta}$ to accept words over Σ as following $\hat{\delta} : X \times \Sigma^* \longrightarrow X$. A state $x' \in X$ is reachable from the state x it there exists a sequence $T_M \in \Sigma^*$ such that $x' = \hat{\delta}(x, T_M)$ and we write $x \longmapsto x'$, and $x' = \hat{\delta}(x, T_M)$ is a path ξ in M if $x = x_0 \in X$. The state x_0 is called beginning of ξ and x_n is called the end of ξ. In the following, we call the strings $T_M = t_1 \cdots t_n$, with $t_1, \cdots, t_n \in \Sigma$ a trajectory in the system M. Thus, the path ξ is defined as $\xi = x_0 \xrightarrow{t_1} x_1 \cdots x_{n-1} \xrightarrow{t_n} x_n = x_0 \xrightarrow{T_M} x_n$.

Let T_M be a trajectory in Σ. For each trajectory $T_M \in \Sigma^*$, $|T_M|$ denoted its length. We say, the trajectory $T_M \in \Sigma^*$ is accepted by M if and only if there exists a path $\xi = x_0 \xrightarrow{T_M} x_n$, labeled by T_M, in the state diagram of M leading from start state x_0 to terminal state $x_n \in F$.

Any subset of Σ^* is called a language over Σ. The generated language of M, denoted by $\mathscr{L}(M)$ is defined as

$$\mathscr{L}(M) = \{T_M \in \Sigma^* \mid \hat{\delta}(x_0, T_M) \in X\}.$$

The language accepted by the system M is the set of all and only those trajectories over Σ that are accepted by M. The marked language accepted by M is defined by

$$\mathscr{L}_m(M) = \{T_M \in \Sigma^* \mid \hat{\delta}(x_0, T_M) \in F\}.$$

The language accepted by a deterministic FSA $\mathscr{L}_m(M)$ is called a regular language. A FSA of M is deterministic, if any given path in M labeled by trajectory $T_M \in \Sigma^*$ has a unique run, otherwise, FSA of M is non-deterministic.

The projection of strings from $\mathscr{L}(M) \longrightarrow \Sigma_o^*$ is denoted by $P : \mathscr{L}(M) \longrightarrow \Sigma_o^*$. Given a strings $T_M \in \mathscr{L}(M)$, P is obtained by removing all elements of Σ_{uo} in string T_M.

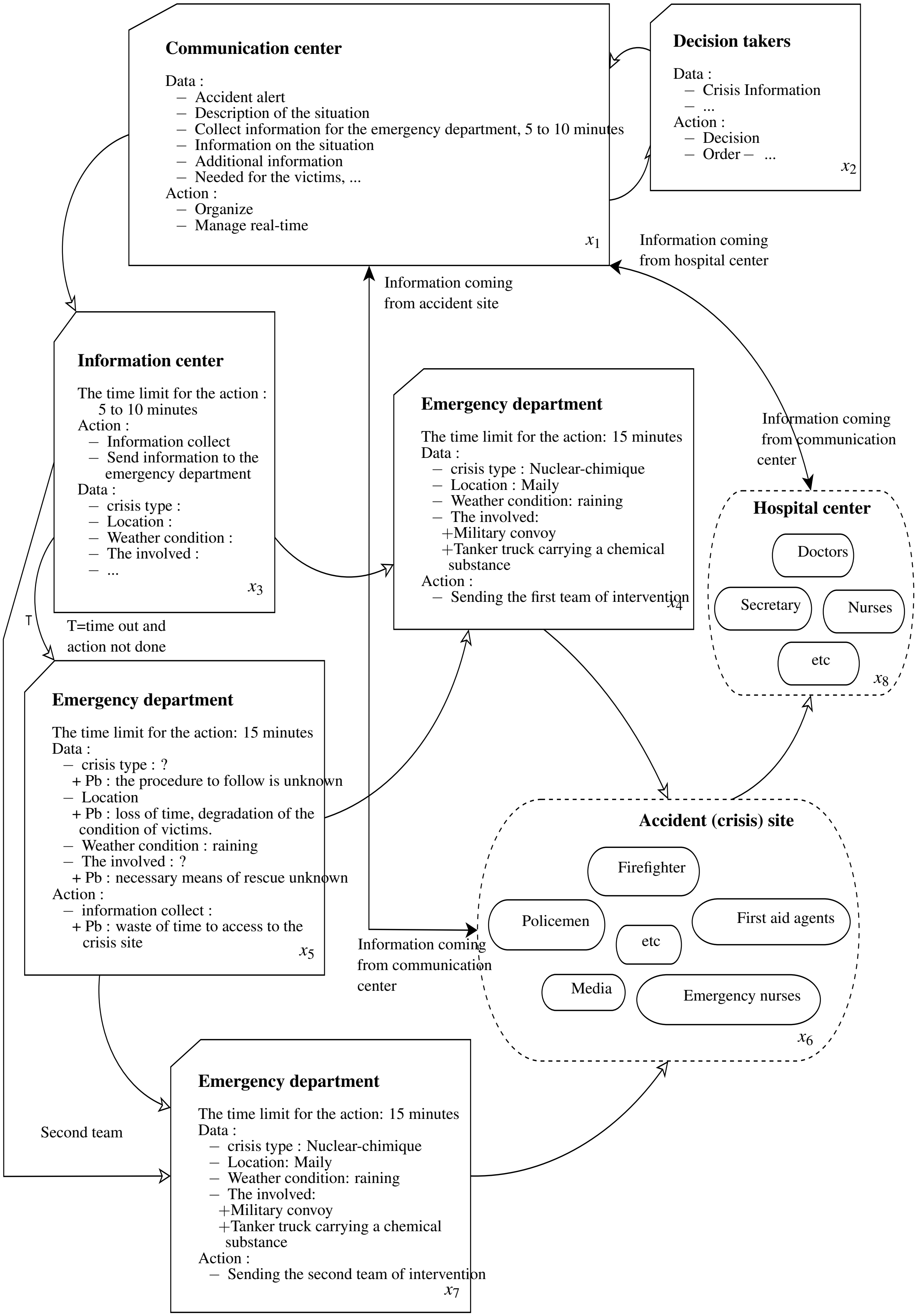

Communication center
Data :
– Accident alert
– Description of the situation
– Collect information for the emergency department, 5 to 10 minutes
– Information on the situation
– Additional information
– Needed for the victims, ...
Action :
– Organize
– Manage real-time
x_1
Decision takers
Data :
– Crisis Information
– ...
Action :
– Decision
– Order – ...
x_2
Information coming from hospital center
Information coming from accident site
Information center
The time limit for the action : 5 to 10 minutes
Action :
– Information collect
– Send information to the emergency department
Data :
– crisis type :
– Location :
– Weather condition :
– The involved :
– ...
x_3
Emergency department
The time limit for the action: 15 minutes
Data :
– crisis type : Nuclear-chimique
– Location : Maily
– Weather condition: raining
– The involved:
+Military convoy
+Tanker truck carrying a chemical substance
Action :
– Sending the first team of intervention
x_4
Information coming from communication center
Hospital center
Doctors
Secretary
Nurses
etc
x_8
T
T=time out and action not done
Emergency department
The time limit for the action: 15 minutes
Data :
– crisis type : ?
+ Pb : the procedure to follow is unknown
– Location
+ Pb : loss of time, degradation of the condition of victims.
– Weather condition : raining
– The involved : ?
+ Pb : necessary means of rescue unknown
Action :
– information collect :
+ Pb : waste of time to access to the crisis site
x_5
Accident (crisis) site
Firefighter
Policemen
etc
First aid agents
Media
Emergency nurses
x_6
Information coming from communication center
Second team
Emergency department
The time limit for the action: 15 minutes
Data :
– crisis type : Nuclear-chimique
– Location: Maily
– Weather condition: raining
– The involved:
+Military convoy
+Tanker truck carrying a chemical substance
Action :
– Sending the second team of intervention
x_7

Figure 2. Interaction between different actors involved in the crisis management.

3. Discrete model of crisis management

In this paper, the interaction between different actors and groups involved in crisis management is viewed as discrete model. The discrete model corresponding to the crisis management is modeled as a FSA. This is represented as a quintuple structure,

$$G = (X, \Sigma, Y, \varphi, x_0),$$

with φ is the transition relation, φ is the extension of δ of the system M, the relation φ has type $X \times \Sigma \rightarrow X \times Y$. For instance, $z' = (x', y') \in \varphi(x, t_i)$, with x, $x' \in X$, $y' \in Y$ and $t_i \in \Sigma$.

Example 1: *The Figure 3 shows the FSA of the crisis management corresponding to the figure2, with $x_0 = x_1$. In this example,*

- $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$,
- $\Sigma = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}\}$,
- $Y = \{y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8\}$,

The figure 3 shows a discrete event model and outputting y_i for all $i \in \{1, \cdots, 8\}$, when the system is in state x_i for $i \in \{1, \cdots, 8\}$, respectively.

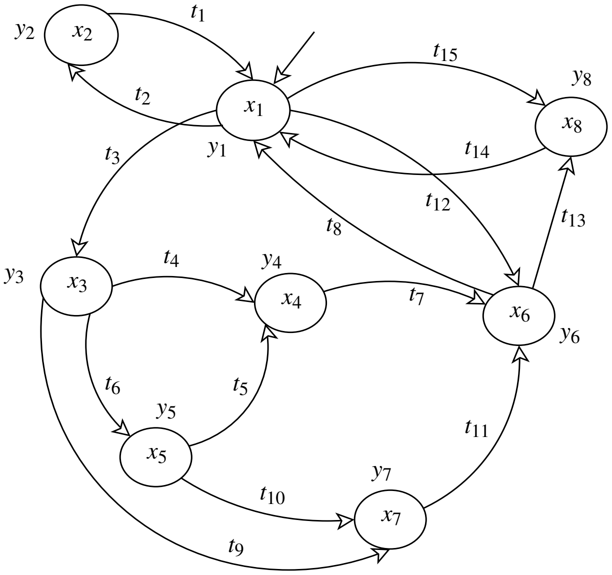


Figure 3. FSA model of the crisis management.

The FSA model of the crisis management shown in figure 3, allows one hand to monitor the communication and information between various groups involved in crisis management, and also to supervise same specific behaviors (or pattern) that can be a critical situation in the management. The notion of pattern means to define a language associated with a path of system G that we are interested in for the purpose of diagnosis. Other word, the pattern is defined as the recognition problem of the path whose intention is to answer the question whether trajectories corresponding to observed path are accepted or not by the model of the pattern. In (Ye & Dague, 2012), a pattern is define as a FSA. The language may be associated with the occurrence of single or multiple critical situation.

The transition function φ of $G = (X, \Sigma, Y, \varphi, x_1)$ can be extended to take input sequence. For example in figure 3, $\varphi(x_1, t_3) = \{x_3, y_3\}$ and $\varphi(x_1, t_3t_9) = \{x_7, y_3y_7\}$.

The equation $\varphi(x_1, t_3) = \{x_3, y_3\}$, means when the system G is in state x_1 and the communication event t_3 is emitted, the system G moves in the state x_3 and sends a communicate message y_3 (output).

Define two projections φ_1 and φ_2 of φ such that φ_1 gives the states reached from a state and an input given. The projection φ_2 defines the input/output pairs from state. These projections are defined as

$$\begin{cases} \varphi_1(x, t_i) = \{x' \in X \mid \exists\, y' \in Y \ \text{ such that } \ (x', y') \in \varphi(x, t_i)\}, \\ \varphi_2(x, t_i) = \{y' \in Y \mid \exists\, x' \in X \ \text{ such that } \ (x', y') \in \varphi(x, t_i)\}, \end{cases}$$

The projections φ_1 and φ_2 of φ may be extended as well to take input sequences. By applying φ_1 and φ_2 on the diagram represented in figure 3, we get

$$\varphi_1(x_1, t_3t_9) = \{x_7\} \ \text{ and } \ \varphi_2(x_1, t_3t_7) = \{y_3y_7\}.$$

Let $L(G)$ be the language defined by the FSA G containing the input sequence allowed by G. Formally

$$L(G) = \{T_G \mid T_G \in \Sigma^* \ \text{ and } \ \Delta_G \in \varphi_2(x_1, T_G)\},$$

with x_1 start state and Δ_G output corresponding to the input T_G. The state $x \in X$ of G has an associated language

$$L_G(x) = \{T_G \mid T_G \in \Sigma^* \ \text{ and } \ \Delta_G \in \varphi_2(x, T_G)\},$$

with $\Delta_G = y_1 \cdots y_k$ and $T_G = t_1 \cdots t_k$ such that $y_1, \cdots, y_k \in Y$ and $t_1, \cdots, t_k \in \Sigma$. The language $L_G(x)$ is the set of all trajectory that originate from the state x of the system G. Clearly in figure 3, $L(G) = L_G(x_1)$.

Let $\mathscr{K}(L_G(x_1), t_i)$ be the trajectory that ends with t_i (Genc & Lafortune, 2009). Formally

$$\mathscr{K}(L_G(x_1), t_i) = \{T_G = T_0t_i \mid T_0 \in \Sigma^* \ \text{ and } \ t_i \in \Sigma\}.$$

We recall here, the FSA model of a dynamic system is defined as $G = (X, \Sigma, Y, \varphi, x_1)$, where $\varphi : X \times \Sigma \longrightarrow X \times Y$ is the transition function.

4. Discrete event dynamic system

In the literature, the event set Σ may include failure events $\Sigma_f = \{\Sigma_1, \cdots, \Sigma_p\}$. Indeed, a dynamic system can have p failure modes of critical situations $(F_1, \cdots, F_p)$ that describe the condition of the system. In addition to normal and failure modes the system may have a degraded mode, denoted N_d. Therefore, the state set X can be partitioned according to the condition of the system.

$$X = X_N \cup X_{N_d} \cup X_{F_1} \cup \cdots \cup X_{F_p}.$$

The condition of the system goes in X_{N_d}, when the degradation event occurs and in X_{F_i} when the failure event occurs.

To define the condition map of a dynamic system on a trajectory T_G of G, we introduce the label propagation function $LP : X \times \Omega \times \Sigma^* \to \Omega$. $LP(x, \lambda, T_G)$ propagates the label λ over $T_G \in \Sigma^*$, starting from $x \in X$ and following the dynamics of G, *i.e.* according $L_G(x)$, with $x \in X$, $\lambda \in \Omega$ and $L_G(x) \in \Sigma^*$ such that $T_G = \mathscr{K}(L_G(x), \mathscr{T})$.

$$LP(x,\lambda,T_G) = \begin{cases} N, \text{ if } \exists\, x' \in X \mid x' \in \varphi_1(x,T_G) \text{ and } x' \in X_N \\ N_d, \text{ if } \exists\, x' \in X \mid x' \in \varphi_1(x,T_G) \text{ and } x' \in X_d \\ F_i, \text{ if } \exists\, x' \in X \mid x' \in \varphi_1(x,T_G) \text{ and } x' \in X_{F_i} \end{cases}$$

The definition of the conditions map may be extended to subsets of X.

$$\text{for all } z \subseteq X,\ LP(z,\lambda_z,T_G) = \bigcup_{x_1 \xrightarrow{T_G} x_i \in z} \{LP(x_1,\lambda_i,T_G)\}.$$

Let $x_1, \cdots, x_m \in X$ and $m \in \mathbb{N}$ such that $z = \{(x_1,\lambda_1), \cdots, (x_m,\lambda_m)\}$. The system's condition λ_i is normal if $\lambda_i = N$ for all $1 \le i \le m$, certain if $\lambda_i = F_i$ for all $1 \le i \le m$ and uncertain if there exist $\lambda_j = N$ and $\lambda_i = F_i$ for same $1 \le i, j \le m$. Further detail about notions of certain and uncertain system's condition may be found in (Zad, Kwong, & Wonham, 2003) and (Genc & Lafortune, 2009).

Example 2: *Figure 3 shows a FSA model of a crisis management. We use the input of the system G to supervise the behavior corresponding to the critical situation. The critical situation that we want to detect is outputting twice y_5 during the crisis management. The first appearance of the output y_5 in the output sequence Δ_G brings the system into the set X_{N_d} corresponding to the degraded mode N_d. The second appearance of the output y_5 in the output sequence Δ_G during the crisis management brings the system into the set X_F corresponding to the critical mode. In this example*

$$\begin{aligned} X &= \{x_1,\, x_2,\, x_3,\, x_4,\, x_5,\, x_6,\, x_7,\, x_8\} \\ \Omega &= \{N,\, N_d,\, F\} \\ Y &= \{y_1,\, y_2,\, y_3,\, y_4,\, y_5,\, y_6,\, y_7,\, y_8\} \end{aligned}$$

The necessary and sufficient condition for the pattern of a *DES* is based on the learning diagnoser and prediction of *DES*. The learning diagnoser is obtained from the standard diagnoser.

5. Standard diagnoser

A standard diagnoser denoted D_G must be able to detect and isolates faults and failures (Sampath et al., 1995), or to detect critical situations in crisis management. A fault implies a certain level of degradation of performance and a failure on the other hand denotes a complete operational breakdown of equipment or the process (Yunxia, 2003). A standard diagnoser is a FSA built for to detect and isolates faults and failures of G. Let $G = (X, \Sigma, Y, \varphi, x_0)$ be the discrete event model for the dynamic system that we want supervise. The set Y is the output of system G. The standard diagnoser that we use for discrete event dynamic systems is a FSA that takes the output sequence $\Delta_G = y_1y_2\cdots$ of system G as its input as shown in figure 4, with λ_i the condition functioning of the system.

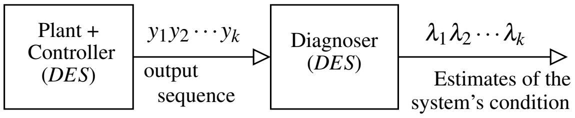


Figure 4. System and Supervision pattern

The standard diagnoser D_G of G is defined as $D_G = (Z, Y, \Omega, \zeta, z_0)$, with Z is the set of standard diagnoser state, Y is the set of standard diagnoser input, Ω is the set of standard diagnoser output, ζ is the standard diagnoser state transition function, the relation ζ has type $Z \times Y \to Z$, $z_0 \in Z$ is the start state of the standard diagnoser.

The diagnoser state space Z is the resulting subset of $2^{X\times\Omega}$ composed of the state of the diagnoser that are reachable from z_0 under ζ. The initial state z_0 of the diagnoser is defined by $z_0 = (x_0, \lambda_0)$. Assume the system G is normal to start, then $\lambda_0 = N$. State $z \in Z$ is given by

$$z = \{(x_1,\lambda_1),(x_2,\lambda_2),\cdots,(x_{n-1},\lambda_{n-1}),(x_n,\lambda_n)\},$$

where $x_i \in X$ and $\lambda_i \in \Omega$, for all $i \in \{1, \cdots, n\}$. In the following, $|z| = 1$.

Basing on the output sequence $\Delta_G = y_1y_2\cdots y_k$ of the system G, a state $z_k = (x_k, \lambda_k) \in Z$ is determined to which x_k may belong at the time that y_k was generated. For the diagnoser, the estimate of the system's condition from x_1 will be $LP(x_1, \lambda_k, \Delta_G)$ such that $(x_1, \lambda_1) \in \zeta(x_k, \lambda_k)$, with $z_1 = (x_1, \lambda_1)$ and $z_k = (x_k, \lambda_k)$.

The diagnoser state transition is defined by
$z_{k+1} = \zeta(x_k, y_{k+1})$ with $z_k = (x_k, \lambda_k)$ and $y_{k+1} \in Y$. In the following, we write the diagnoser state $z_k = (x_k, \lambda_k)$ as $z_k = (x_{z,k}, \lambda_k)$. The standard diagnoser presented further above is shown in figure 5. Here, the pattern that has to supervise is having twice y_5 in the output sequence during the crisis management. *We remind that $y_1, y_2 \cdots \in Y$ are outputs of the system G and inputs for the diagnoser D_G.*

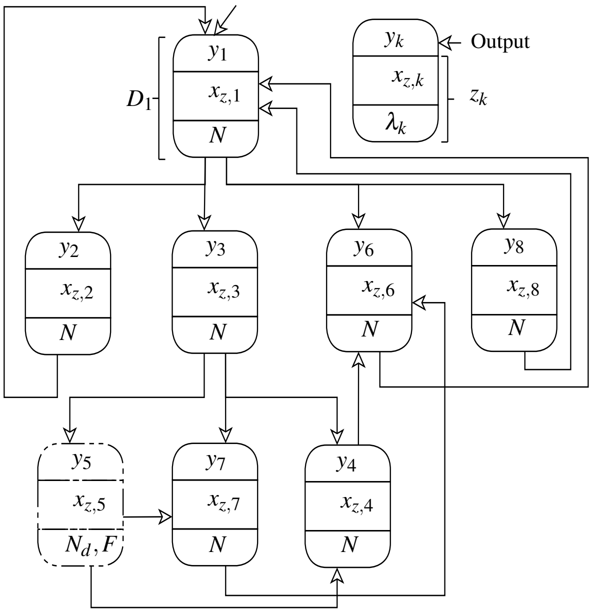


Figure 5. Standard diagnoser of the critical behavior in crisis management.

We address the problem of supervision pattern of a Discrete Event Dynamic System (*DEDS*). Let H be a bounded set of finite-length strings over Σ. The pattern can be define as a bounded H. The definition of the language that should be recognized by the supervision pattern depend the problem studied. In this paper, the critical situation (behavior) during the crisis management is detected if and only if.

$$\begin{cases} \mathscr{L}_{D_G}(x_{z,1}) = \{T_{D_G} \mid T_{D_G} \in Y^* \text{ and } \Delta_{D_G} \in \zeta_2(x_{z,1}, T_{D_G})\}, \\ \text{such that, it exists a language } L \in \Sigma^* \text{ defined by} \\ L = \{T_G \in \Sigma^* \mid T_{D_G} \in \varphi_2(x_1, T_G\} \text{ and } f(L) = true, \\ f(L) \text{ is a condition to define.} \end{cases}$$

In this paper, the condition is given as

$$\begin{cases} f(L) = |P_o(L)| \geq \mathrm{C}, \\ P_o : \Sigma^* \longrightarrow H^*,\ H \subseteq \Sigma, \\ \text{C = Criteria, here C is a positive number,} \end{cases}$$

with P_o is the projection of strings, $T_{D_G} = y_1 y_1 \cdots$, and φ_2 is the extension of φ of G. For the behavior that we want to supervise here, we have $H = \{t_6\}$ and the Criteria $C = 2$. Until now, only the occurrence of t_6 brings G in the state x_5 as shown in figure 3.

The supervision pattern shown in figure 6 recognizes the language $\mathscr{L}_{D_G}(x_{z,1})$ if and only if the condition $|P_o(L)| \geq 2$ is verified.

The trajectory $\mathscr{L}_{D_G}(x_{z,k})$ is used to predict the evolution of the situation during the crisis management. The prediction, in general, is the process of making a decision about a future result or evolution of a situation. In the next section, we introduce the problem of prediction of discrete event system.

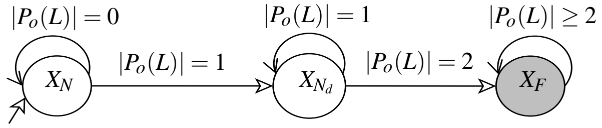


Figure 6. New supervision pattern for the critical situation in crisis management.

6. The prediction of *DEDS*

The prediction of a trajectory (or equivalently, sequence) of a dynamic system behavior is defined in the context of formal language.

Let $\mathscr{L}_{D_G}(x_{z,1})$ denote the set of all trajectory that originate from the start state diagnoser $z_1 = (x_{z,1}, \lambda_1)$, and $\mathscr{K}(\mathscr{L}_{D_G}(x_{z,1}, y_\alpha))$ the trajectory ends with $y_\alpha \in Y$.

$$\mathscr{K}(\mathscr{L}_{D_G}(x_{z,1}, y_\alpha)) = \{\beta \in Y^* \text{ such that } \beta = y_1 \cdots y_n y_\alpha\}.$$

Let ζ_1 and ζ_2 be the two projections of ζ of diagnoser D_G, with ζ_1 is given by

$$\zeta_1(x_{z,k-1}, y_k) = \{x_{z,k} \mid \exists\, \lambda \text{ such that } (x_{z,k}, \lambda) \in \zeta(x_{z,k-1}, y_k)\},$$

with $\lambda = LP(x_{z,1}, \lambda_k, \beta) \in \Omega$ if $y_\alpha = y_k$ and the state $z_k = (x_{z,k}, \lambda) \subseteq Z$ is the state estimate of D_G at time k,
and ζ_2 is defined by

$$\zeta_2(x_{z,k-1}, y_k) = \{\lambda \mid \exists\, x_{z,k} \in z_k \text{ such that } z_k \in \zeta(x_{z,k-1}, y_k)\}.$$

Let $\psi(x)$ be the function giving the state immediately after the state x. This function is defined as

$$\psi(x) = \{x' \mid \exists\, y \in Y \text{ such that } x' \in \zeta_1(x, y)\}.$$

Roughly speaking, a diagnoser state is predictable if it is always possible to detect the future diagnoser state, strictly before to arrive in this state. In this paper, we base only on the output sequence of *DEDS* model of system G to predict the future state or evolution. The prediction of the future diagnoser state at time k, when $x_{z,k}$ is generated, is given by

$$\widehat{x}_{z,k+1} = \psi(x_{z,k}) \cap \zeta_1(x_{z,k}, y_{k+1}),$$

with $y_{k+1} \in Y$ and y_{k+1} is the input of D_G.

The predicted state of the diagnoser D_G is :
$\widehat{z}_{k+1} = (\widehat{x}_{z,k+1}, y_{k+1})$.
Thus, the prediction of the trajectory $\widehat{\mathscr{K}}(\mathscr{L}_{D_G}(x_{z,1}, y_{k+1}))$ is the form: $\widehat{\mathscr{K}}(\mathscr{L}_{D_G}(x_{z,1}, y_{k+1})) = \{\widehat{\beta} = y_1 \cdots y_k y_{k+1}\}$.
The prediction of the system's condition is the propagation of the label λ_{k+1} over $\widehat{\beta}$, defined by $LP(x_{z,1}, \lambda_{k+1}, \widehat{\beta})$. Finally the diagnosis state predicted from $x_{z,1}$ is the form

$\widehat{z}_{k+1} = (\widehat{x}_{k+1}, LP(x_{z,1}, \lambda_{k+1}, \widehat{\beta}), x_k \longmapsto \widehat{x}_{k+1}$.

For instance, suppose at time k the output sequence $\Delta_G = y_1 y_3$ is observed, then the diagnoser state is $x_{z,k} = x_{z,3}$ and the system's condition from $x_{z,1}$ is $LP(x_{z,1}, \lambda_3, y_1 y_3) = N$.

When the system is in the state x_3 in figure 3, and if the next output symbol y_{k+1} is anything other than y_4, y_5, y_7, we get

$$\psi(x_{z,3}) \cap \zeta_1(x_{z,3}, y_{k+1}) = \emptyset,$$

that means the observation generated after y_3 is inconsistent with the model dynamic and the diagnoser cannot proceed. The currant diagnoser state $x_{z,k+1}$ is different to diagnoser state $\widehat{x}_{z,k+1}$ predicted before. Basing in the language $\mathscr{L}_G(x_1) = T_G$, in particular the output sequence Δ_G, we determine the state candidate.

When the output sequence is inconsistent with the model of the system G, then we have to revise the model of G by adding to its new transition that we believe are missing in the nominal model. This situation is not interpreted as a faulty situation, because we did not add new states. All the normal and fault modes are known and we add only the messing transitions. Adding new transitions in Σ of G is called learning diagnoser. In the next section we detail the construction of a learning diagnoser.

7. Learning Diagnoser

A learning diagnoser is a standard diagnosis that tolerant of missing transitions (information) about the system to be diagnosed. The learning diagnosis must be able to learn the true model of the system G, when missing information about the system are presented.

Let t_{new} be a new event detected and not found in Σ of system G. The new set of input events of G is given by $\Sigma_{new} = \Sigma \cup \{t_{new}\}$. A transition $x_d \xrightarrow{t_{new}} x_a$ is ordered pair of state denoting a transition from the state x_d to the state x_a. Let φ' be the extend function transition of φ of the system G such that

$$\varphi_1'(x, t_j) = \begin{cases} x_a & \text{if } x = x_d \text{ and } t_j = t_{new} \\ \varphi_1(x, t_j) & \text{otherwise} \end{cases}$$

Let be a dynamic model G' of G defines as $G' = extend(G, \Pi) = (X, \Sigma \cup \Pi, Y, \varphi', x_0)$. And G' is called the extension of G by Π, with Π is the set containing all the new transitions founded. The set transition Π is empty, if the model G of the system is consistent with the output sequence.

For instance in figure 2, when an accident happen, the information center is going to send data and actions to the emergency department. All the information are sent in 5 minutes, but with a wrong weather condition. For example, if the information center sends a temperature of 30°C in winter (Europa). Then the system is going to detect the temperature 30°C in winter in Europa is wrong. That mean the first team can not go before the correct temperature. During this wait, the first team loses time (waste of time) that mean in the diagram 3, the system is in the state x_5. The transition from the state x_4 to the state x_5 is a new transition for the system G. The resulting diagnoser, including the new transition, is shown in figure 7, then $\psi(x_{z,4}) = \{x_{z,6}, x_{z,5}\}$.

When the model of G is inconsistent with the output sequence, the subset H for the supervision pattern may be updated. In this paper, the critical situation that we want supervise is going twice in the state x_5, then with the new transition t_{new}, we can go to the x_5 (event t_6 or t_{new} occurs). Then the new subset for the supervision pattern is define by $H = H_{update} = \{t_6, t_{new}\}$. The critical pattern of behavior in crisis management is detected if and only if

$$\begin{cases} \mathscr{L}_{D_G}(x_{z,1}) = \{T_{D_G} \mid T_{D_G} \in Y^* \ \& \ \Delta_{D_G} \in \zeta_2(x_{z,1}, T_{D_G})\} \\ \text{such that it exist a language } L \text{ defined by} \\ L = \{T_G \in \Sigma^* \text{ such that } T_{D_G} \in \varphi_2(x_1, T_G)\} \text{ and } |P_o(L)| \geq \mathrm{C} \\ P_o^{new} : \Sigma_{new}^* \longrightarrow H_{update}^*, \ \Pi \supseteq H_{update} = H \cup \{t_{new}\} \\ \text{C = Criteria and in figure 6 } P_o = P_o^{new}. \end{cases}$$

$P_o^{new} : \Sigma_{new}^* \longrightarrow H_{update}^*$ is the new definition of P_o and Criteria=2. The fact to update the bounded set $H \subseteq \Sigma$, we obtain a learning supervision pattern (*see figure 6*).

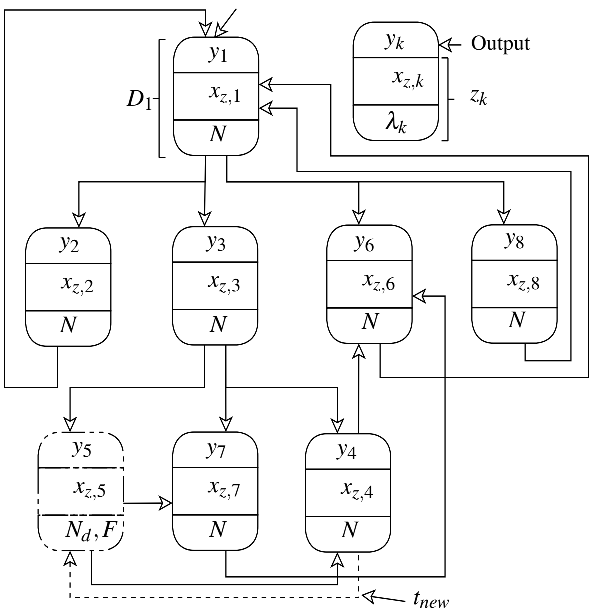


Figure 7. Learning diagnoser of the critical behavior in crisis management.

The diagnoser, as a Finite State Automaton, can be automatically translated into computer code. For example, the algorithm for a part of diagnoser 7, is given by

Algorithm 1 D_1

```
Require: $D_1 : z = (x_{z,k}, N)$ and $y$
  while $y == y_1$ do
    read $y$
    $z_1 \leftarrow (x_{z,k}, N)$
  end while
  if y==y2 then
    go to $D_2$
  else if $y == y_3$ then
    go to $D_3$
  else if $y == y_6$ then
    go to $D_6$
  else if $y == y_8$ then
    go to $D_8$
  else
    go to inconsistency
  end if
```

8. Conclusion and perspectives

In this paper, we proposed a representation of the crisis management as a set of couple of states and events. A learning diagnoser and prediction approaches are proposed and applied onto crisis management. Also, a method of adaptive supervision pattern is proposed in this paper.

Future work will also focus to introduction of the notion of the probability and to integrate time information onto the new transitions detected.

Acknowledgment

This work is supported by the CPER prject AidCrisis, sponsored by the Champagne-Ardenne region and the French ministry of higher education and research.

References

Benkhelifa, I., Moussaoui, S., & N-Taboudjemat, N. (2013). Locating emergency responders using mobile wireless sensor networks. *In proceedings of IEEE, ACM, Proceeding ISCRAM, 10th Internationa Conference on Information Systems for Crisis Response and Management. Baden-Baden, Germany.*

Cabasino, M. P., Giua, A., & Seatzu, C. (2010). Fault detection for discrete event systems using petri nets with unobservable transition. *Automatica*, *Vol. 46, Issue 9*, 1531-1539.

Genc, S., & Lafortune, S. (2009). Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, *Vol. 45, Issue 2*, pp. 301 - 311.

Kwong, R., & Yonge-Mallo, D. (2011). Fault diagnosis in discrete-event systems: Incomplete models and learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *Volume 41 Issue : 1*, pp. 118-130.

Reuter, C., Heger, O., & Pipek, V. (2013). Combining real and virtual volunteers through social media. *In proceedings of IEEE, ACM, Proceeding ISCRAM, 10th Internationa Conference on Information Systems for Crisis Response and Management. Baden-Baden, Germany.*

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete event systems. *IEEE Transaction On Automatic Contol*, *vol. 40, No. 9*, 1555-1575.

Sayed-Mouchaweh, M., & Billaudel, P. (2012). Abrupt and drift-like fault diagnosis of concurent discrete event systems. *Machine Learning and Applications (ICMLA)*, *vol. 2*, 434 - 439.

Takai, S., & Kumar, R. (2012). Distributed failure prognosis of discrete event systems with bounded-delay communications. *IEEE Tansactions On Automatic Control*, *vol. 57, No. 5*, pp. 1259 - 1265.

Xi-Rien, C. (1989). The predictability of discrete event systems. *IEEE Transaction Automatic Contol*, *vol. 34 (11)*, pp. 1168-1171.

Ye, L., & Dague, P. (2012). A general algorithm for pattern diagnosability of distributed discrete event systems. *International Conference on Tools with Artificial Intelligence*.

Yunxia, X. (2003). *Integrated fault diagnosis scheme using finite state automaton* (Unpublished master's thesis). National University of Singapore.

Zad, S. H., Kwong, R. H., & Wonham, W. M. (2003). Fault diagnosis in discrete-event systems: framework and model reduction. *IEEE Transaction, on Automatic Contol*, *vol. 48, No. 7*, pp. 1199-1212.

Biographies

Moussa Traore is a Postdoctoral fellow at the University of Reims, France working under supervision of prof. M. Sayed-Mouchaweh and prof. P. Billaudel. He is working on Fault Diagnosis in Discrete Event System represented by Finite State Automaton. He is working on an AidCrisis project sponsored by Champagne-Ardenne region and the French ministry of higher education and research. He obtained his *Ph.D* degree (Control System and Signal Processing) from University Lille 1, France. He did his diploma in Safety Engineering (Optimization and safety of functioning Systems) from University of Technology of Troyes, France, 2006 and Bachelor's degree in Electrical Engineering from Faculty of Sciences and Technology of Nouakchott, Mauritania, 2004. His research work allowed him to publish 2 journal papers, 6 papers with proceedings at international conferences and 3 papers with proceedings at national (France) and international conferences (in French) in the area of diagnosis, prognosis and predictive maintenance.

Moamar Sayed-Mouchaweh received his Master degree from the University of Technology of Compiegne-France in

Author Index

A

Adams, Douglas E. 430
Agarwal, Vivek 237
Ahmed, Rizwan 100
Ahn, Kwang-Il 100
Alonso, Juan J. 449
Alonso-Gonzalez, Carlos 275, 288
Amyot, Normand 486
An, Dawn 159, 642
Ardsomang, Tutpol 150
Astafiev, Konstantin 372

B

Bailey, Kevin D. 492
Balaban, Edward 449
Banerjee, Avisekh 478
Barber, John P. 114
Bartram, Gregory 167
Battat, Mor 540
Bechhoefer, Eric 18, 33, 580
Belec, Mario 486
Berglin, Eric J. 531
Bertram, Torsten 438
Bickford, Randall 237
Biegert, Lydia 492
Billauder, Patrice 694
Biswas, Gautam 402
Bodden, David S. 419
Bole, Brian 325, 503
Bond, Leonard J. 531
Bond, Raymond M. 430
Bortman, Jacob 540
Boskoski, Pavle 107
Brabant, France 486
Brahmakshatriya, Umang 93
Brasco, Christina 677
Breger, Louis S. 185
Bregon, Anibal 275, 288, 312
Buchegger, Thomas 548

C

Camci, F. 624
Carlson, Chris 637
Carricajo, Tomas 127
Carter, Matthew C. 633
Celaya, Jose R. 220, 402, 503
Cerdeira, Paula B. O. 616
Chanthery, Elodie 300
Chen, Huimin 213
Chi, Quach C. 325, 503
Chiachio, Juan 363
Chiachio, Manuel 363
Choi, Joo-Ho 159, 642
Clements, N. Scott 419
Coble, Jamie B. 531

D

Daigle, Matthew 114, 134, 249, 262, 312, 503
Daoud, Sami 492
Decker, Ed 513

E

Efendic, Hajrudin 548
Eker, O. F. 624
Eklund, Neil 677
Elkjaer, Karl 372

F

Fang, Austin 18
Fejzic, Amer 185
Feldman, Alexander 275
Fink, Wolfgang 566
Frenette, Francois-Xavier 486

G

Galvao, Roberto K. H. 74, 616, 664
Garbos, Ray 27
Garcia, Ephrahim 18
Gardner, Craig 492
Ghedira, Khaled 199
Ghoneim, Youssef A. 57
Goebel, Kai 141, 220, 325, 337, 350, 363, 402, 503
Goel, Nita 478
Gomes, Joao P. P. 74, 664
Gouby, Aurelie 49
Guo, Limeng 598

H

Hammani, Moez 199
Harrison, Robert F. 66
Hazra, Budhaditya 461
He, David 33, 580
Hecke, Brandon Van 33
Henager Jr, Chuck H. 531
Heo, Gyungyoung 100
Hess, Andrew 566
Hines, Wesley J. 150
Hirt, Evelyn H. 531
Hogge, Edward 325, 503
How, Jonathan P. 185
Howard, Ian 656

Howell, Mark 57
Hubauer, Thomas 685
Hudon, Claude 486
Hur, Seop 100

I

Ippolito, Corey 381
Iung, Benoit 521

J

Jacazio, Giovanni 39
Jennions, I. K. 624
Jing, Rong 513
Johnston, Kyle B. 114
Juricic, Dani 107

K

Khamassi, Imen 199
Khorasgani, Hamed 402
Kim, Jung T. 573
Kim, Jung-Taek 100
Kim, Nam H. 159, 642
Kinloch, Gordon 492
Klein, Renata 230, 540
Kodali, Anuradha 412
Kogan, Gideon 540
Koul, Ashok K. 478
Kripper, Felipe 127
Kulkarni, Chetan 249, 402
Kumar, Amar 478
Kushnirsky, Alex 540

L

Lacaille, Jerome 49, 589
Lamparter, Steffen 685
Leger, Jean-Baptiste 521
Levesque, Melanie 486
Libraro, Marco 39
Liu, Datong 598, 608
Liu, Yongming 141
Lughofer, Edwin 548
Lutschinger, Dirk 656
Lybeck, Nancy 237

M

Mah, Robert 1
Mahadevan, Sankaran 167
Malere, Joao P. P. 616
Marthaler, Daniel 677
Martin, Rodney 1
Mavris, Dimitris 1
Mbaya, Timmy 381
McArthur, Stephen D. J. 469
Medina-Oliva, Gabriela 521
Mengshoel, Ole J. 381
Meyer, Ryan M. 531
Miller, Jim 27
Mills, Andrew R. 66
Mitchell, Mark R. 531
Mohon, Sara 83
Monnin, Maxime 521
Mornacchi, Andrea 39
Mosher, Mark 637
Moya, Noemi 288

N

Narasimhan, Sriram 312, 461

O

Oliva, Javier A. 438
Orchard, Marcos E. 127
Oza, Nikunj 1

P

Pang, Jingyue 598, 608
Pantula, Shilpa 461
Paredes, Rodrigo 127
Park, Gee-Yong 573
Park, Soo Y. 100
Pascoal, Renata M. 664
Patterson, Jon 27
Patterson-Hine, Ann 412
Pattipati, Bharath 57
Pattipati, Krishna 57
Peng, Tishun 141
Peng, Yu 598, 608
Peters, Glenn 637
Pham, Binh T. 237
Pichler, Kurt 548
Pichler, Markus 548
Piol, Olivier 49
Pisu, Pierluigi 83
Popov, Alexandre 566
Provan, Gregory 275
Pulido, Belarmino 275, 288, 312

Q

Qu, Yongzhi 580

R

Rabenoro, Tsirizo 589
Ramuhalli, Pradeep 531
Raptis, Ioannis A. 1
Reinbacher, Thomas 381
Ribot, Pauline 300
Ringgard, Erling 372
Robinson, Peter 412, 671
Rodrigues, Leonardo R. 11, 74
Roshchin, Mikhail 685
Roychoudhury, Indranil 220, 312

Rozier, Kristin Y. 381
Rus, Guillermo 363
Rusaw,Richard 237

S

Salman, Mutasim 57
Sankararaman, Shankar 141, 262, 337, 350
Saxena, Abhinav 141, 220, 363
Sayed-Mouchaweh, Moamar 199, 694
Schumann, Johann 381
Sconyers, Christopher 1
Serdio, Francisco 548
Shah, Mohak 677
Sharp, Michael E. 556
Sherbahn, Jim 93
Skaf, Zakwan 66
Solomakhina, Nina 685
Sorli, Massimo 39
Srivastava, Alka 478
Stoeckle, Matthew R. 185
Strautkalns, Miryam 671
Sukyoung, Pak 100

T

Teubert, Christopher 134, 325
Traore, Moussa 694

U

Upadhyaya, Belle R. 150

V

Vachtsevanos, George 1, 325
Vasquez, Sixto 325, 503
Vianna, Wlamir O. L. 664
Villeburn, Michael J. 492
Voisin, Alexandre 521

W

Wallace, Christopher J. 469
Watson, Stuart 685
Weber, Adam 503
Weihrauch, Christoph 438
West, Graeme M. 469
Wootan, David W. 531

X

Xi, Zhimin 513
Xiang, Yibing 141

Y

Yacher, Luis 127
Yan, Weizhong 93
Yang, Xiao G. 513
Yin, Shan 608
Yoneyama, Takashi 11, 74
Yu, Lijie 93

Z

Zabi, Said 300
Zaidan, Martha A. 66
Zawada, Tomasz 372
Zhao, Jun 478
Zhu, Junda 580