Introduction to Prognostics

Kai Goebel, Abhinav Saxena, Matt Daigle, Jose Celaya, Indranil Roychoudhury
NASA Ames Research Center, *Prognostics Center of Excellence*
Scott Clements
Lockheed Martin

kai.goebel@nasa.gov

Outline

- Prognostics Overview
  - What is Prognostics?
  - How does Prognostics fit into “PHM”?
  - Types of Prognostic algorithms
- Trends, Remaining Useful Life, & Uncertainty
  - What does a prognostic algorithm tell you?
  - How do you manage thresholds?
  - How does uncertainty screw everything up?
- Prognostics Methods
  - Data-Driven Methods
  - Physics-Based Methods
    - Models
    - Algorithms
  - Hybrid Approaches
  - Example
- Other Considerations
  - Metrics
  - Requirements
- Current Challenges in Prognostics
- Q&A

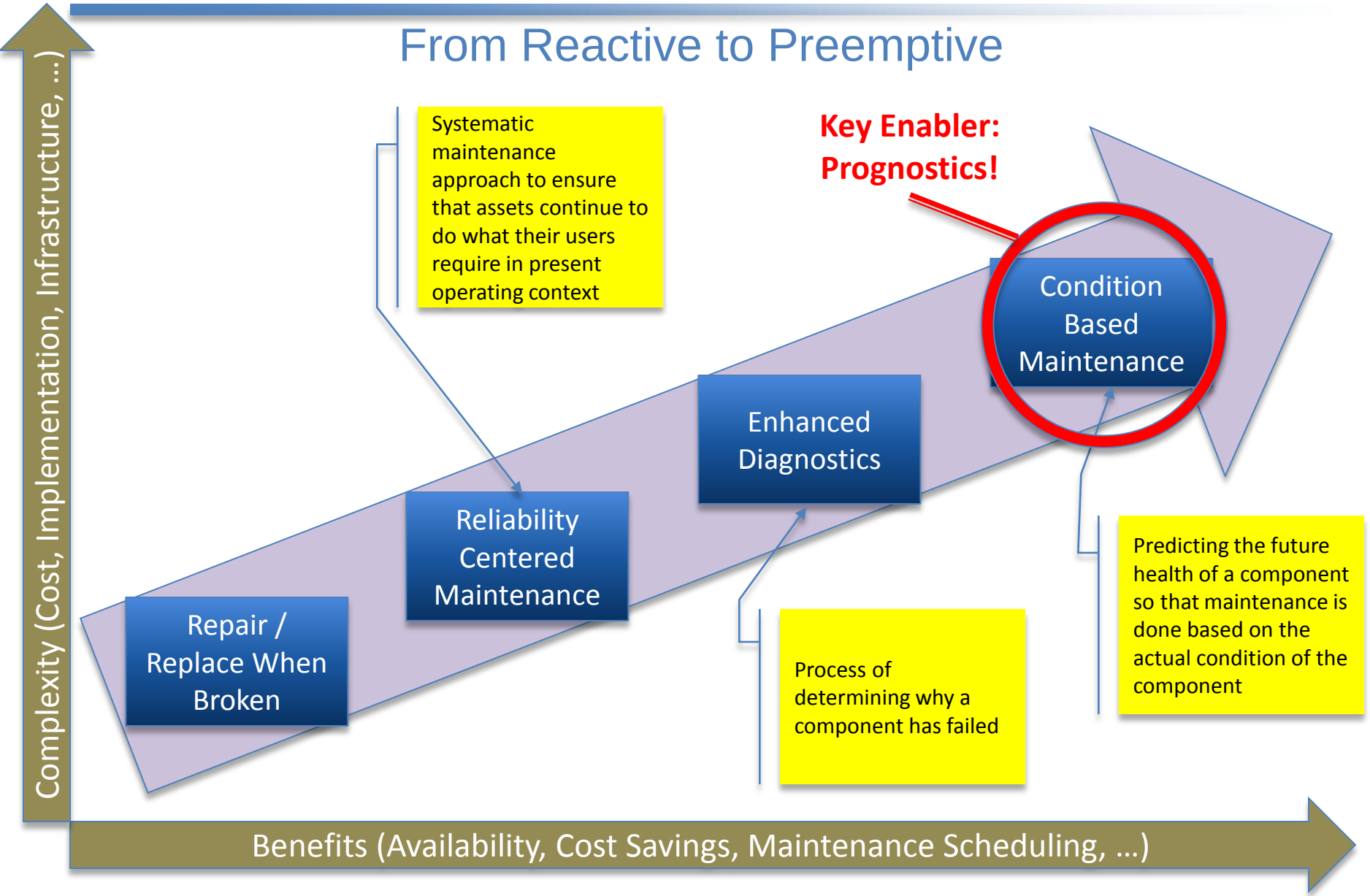
Prognostics Overview

"It's tough to make predictions, especially about the future."

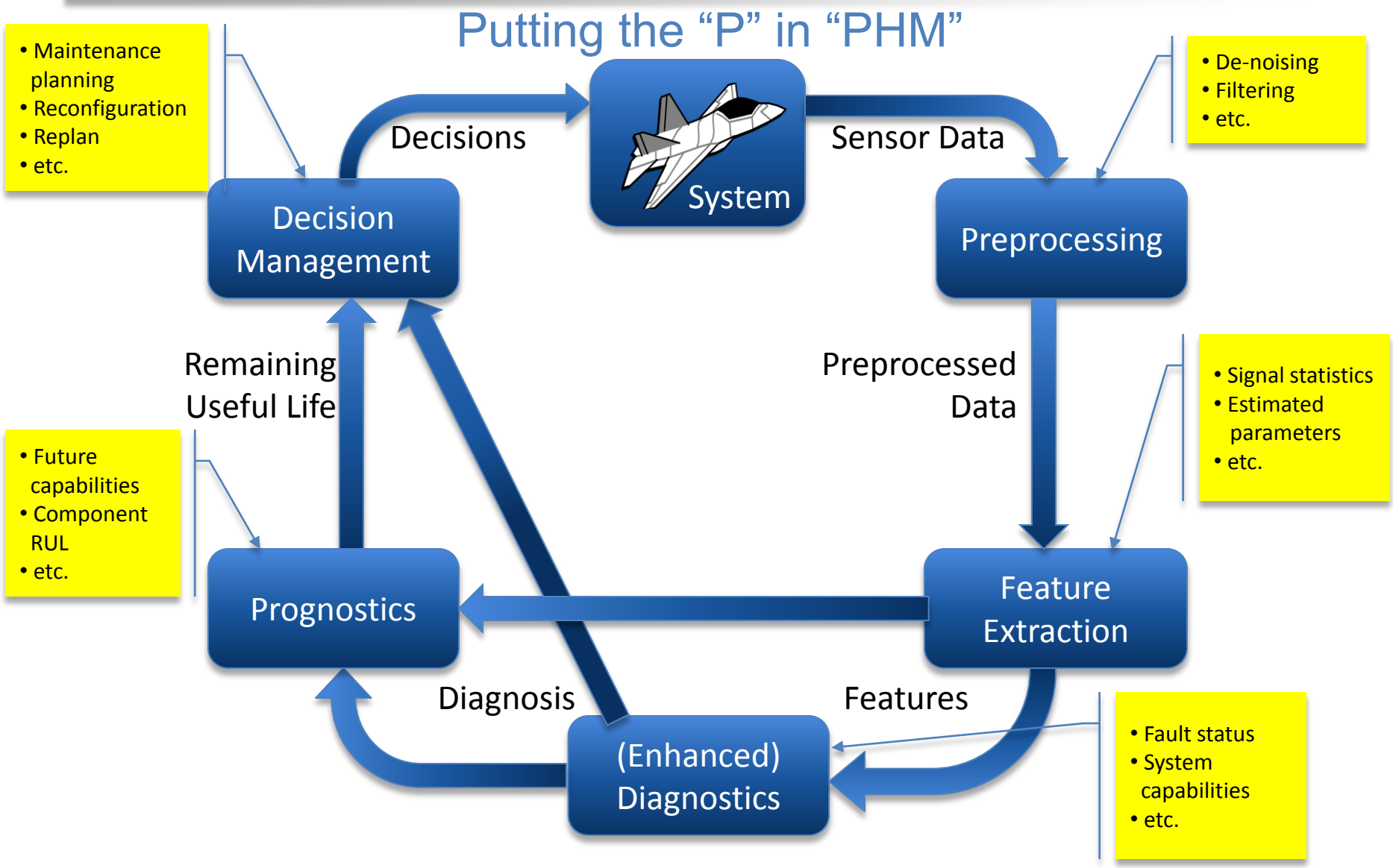
Yogi Berra

Evolution of Maintenance Practices

From Reactive to Preemptive



Prognostics & Health Management



Definitions

So what is “Prognostics” anyway?

- prog·nos·tic
  - M-W.com – “Something that foretells”
  - PHM Community – “Estimation of the *Remaining Useful Life* of a component”

- Remaining Useful Life (RUL) – The amount of time a component can be expected to continue operating within its stated specifications.
  - Dependent on future operating conditions
    - Input commands
    - Environment
    - Loads

Some Different Perspectives

i.e., who cares?

Maintainers

- Scheduling Mx
- Opportunistic Mx
- System Uptime
- Minimize unnecessary Mx

Logisticians

- Spares Positioning
- Reduced Spares Count
- Logistics Footprint

Engineers

- Requirements Satisfaction
- Robustness
- Design for PHM

Regulatory Bodies

- Safety
- Avoid Catastrophic Failures
- Minimize impact on other (healthy) systems

Fleet Management

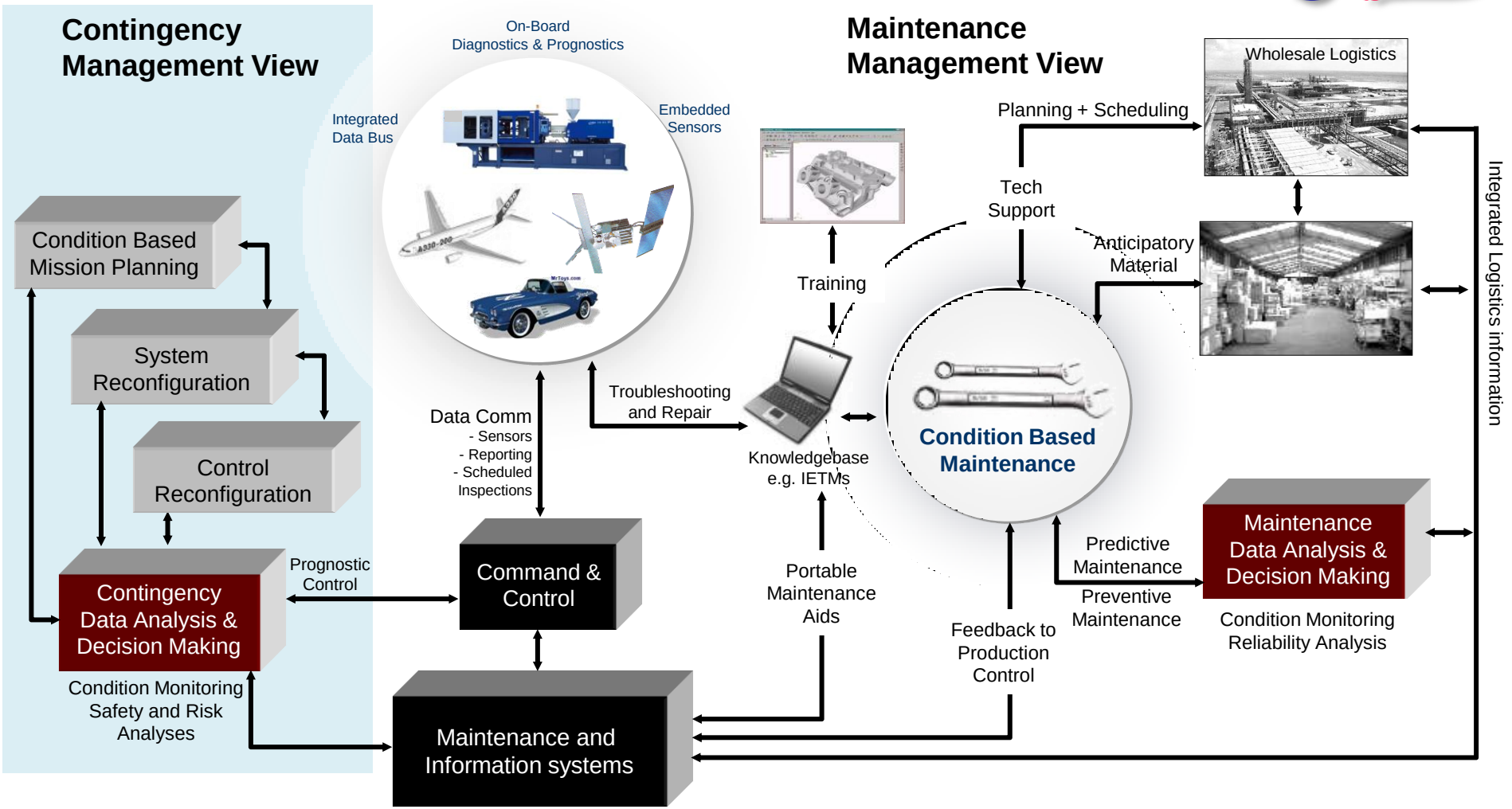
- Fleet Health
- Lifecycle Cost
- Mission Capability
- Mission Planning
- Minimize downtime

Program Mgmt

- Meeting customer expectations

Not Just for Maintenance!

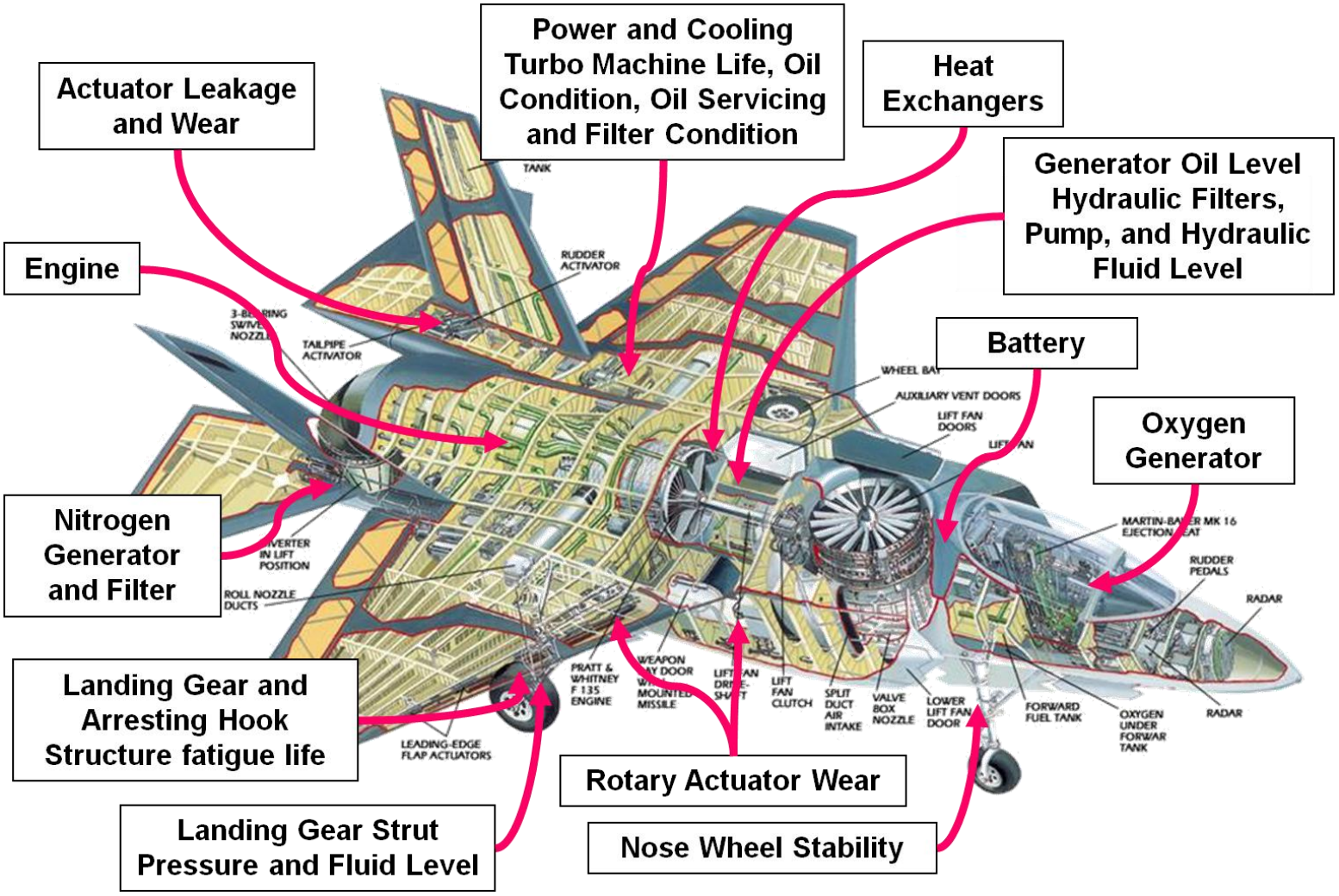
Health Management



- Schematic adapted from: A. Saxena, *Knowledge-Based Architecture for Integrated Condition Based Maintenance of Engineering Systems*, PhD Thesis, Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta May 2007.
- Liang Tang, Gregory J. Kacprzynski, Kai Goebel, Johan Reimann, Marcos E. Orchard, Abhinav Saxena, and Bhaskar Saha, *Prognostics in the Control Loop*, Proceedings of the 2007 AAAI Fall Symposium on Artificial Intelligence for Prognostics, November 9-11, 2007, Arlington, VA.

F-35 Prognostic Candidates

(Some of them, anyway)



Prognostic Algorithm Categories

- Type I: Reliability Data-based
  - Use population based statistical model
  - These methods consider historical time to failure data which are used to model the failure distribution. They estimate the life of a typical component under nominal usage conditions.
  - Ex: Weibull Analysis
- Type II: Stress-based
  - Use population based fault growth model – learned from accumulated knowledge
  - These methods also consider the environmental stresses (temperature, load, vibration, etc.) on the component. They estimate the life of an average component under specific usage conditions.
  - Ex: Proportional Hazards Model
- Type III: Condition-based
  - Individual component based data-driven model
  - These methods also consider the measured or inferred component degradation. They estimate the life of a specific component under specific usage and degradation conditions.
  - Ex: Cumulative Damage Model, Filtering and State Estimation

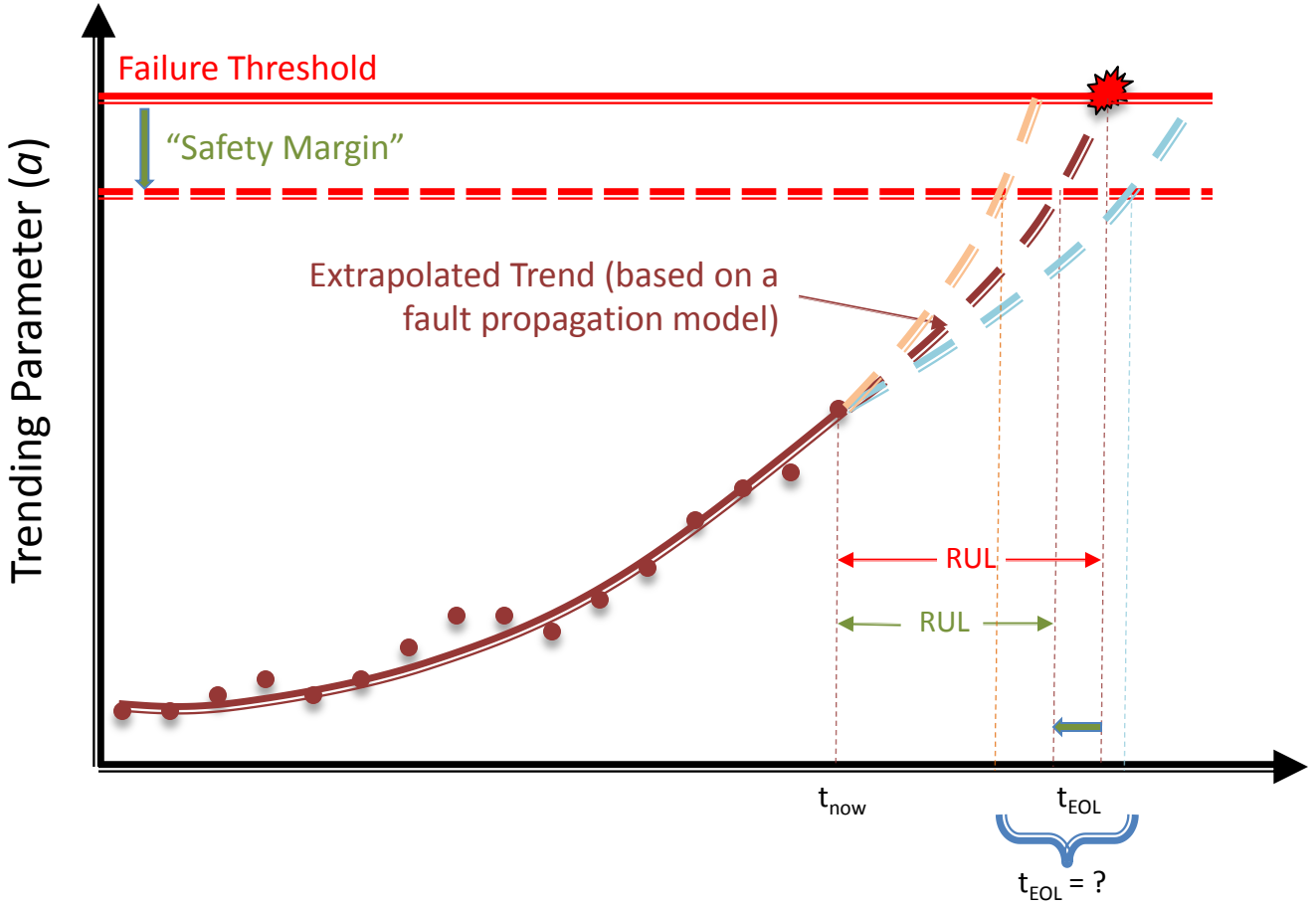
Trends, RUL, & Uncertainty

"In theory there is no difference between theory and practice. In practice, there is."

Yogi Berra

Trends and Thresholds

First, the basics …



How much do you trust the Threshold?

What if the Fault Model is "wrong"?

What about other types of uncertainty?

Types of Uncertainties

You just had to go and make things difficult!

- Model uncertainties – Epistemic
  - Numerical errors
  - Unmodeled phenomenon
  - System model *and* Fault propagation model
- Input uncertainties – Aleatoric
  - Initial state (damage) estimate
  - Manufacturing variability
- Measurement uncertainties – Prejudicial
  - Sensor noise
  - Sensor coverage
  - Loss of information during preprocessing
  - Approximations and simplifications
- Operating environment uncertainties
  - Unforeseen future loads / environment
  - Variability in the usage history data

Systematic uncertainties due to things we could know in principle, but don't in practice.

Statistical uncertainties that may change every time the system is run.

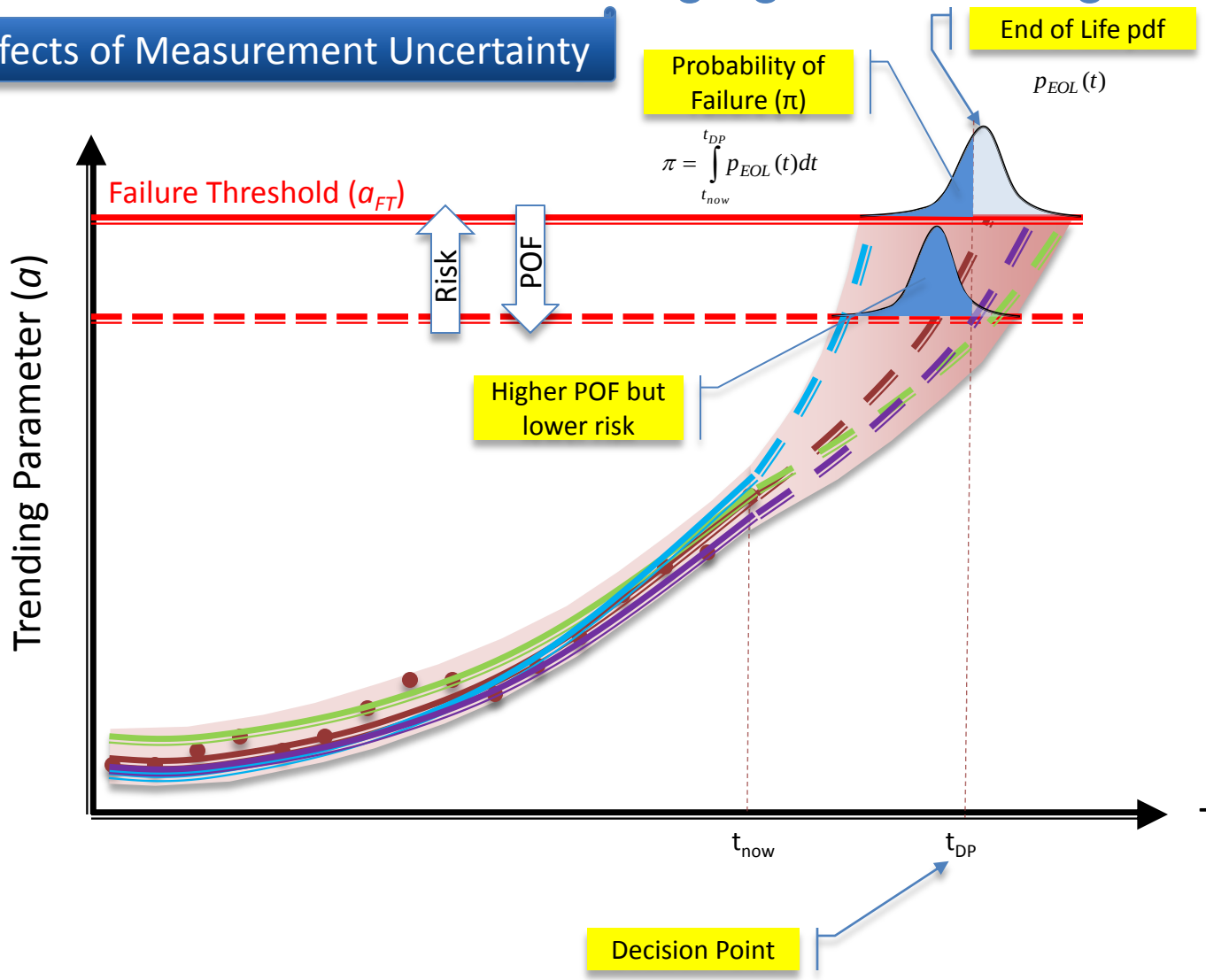
Unknown uncertainties due to the way data are collected or processed.

Can be a mix of any of the above.

Trends and Thresholds Revisited

… now things get interesting!

Effects of Measurement Uncertainty



Band of uncertainty around measurement points

Many possible Models may "fit" the measurements

Use statistics to extrapolate the uncertainty into the future

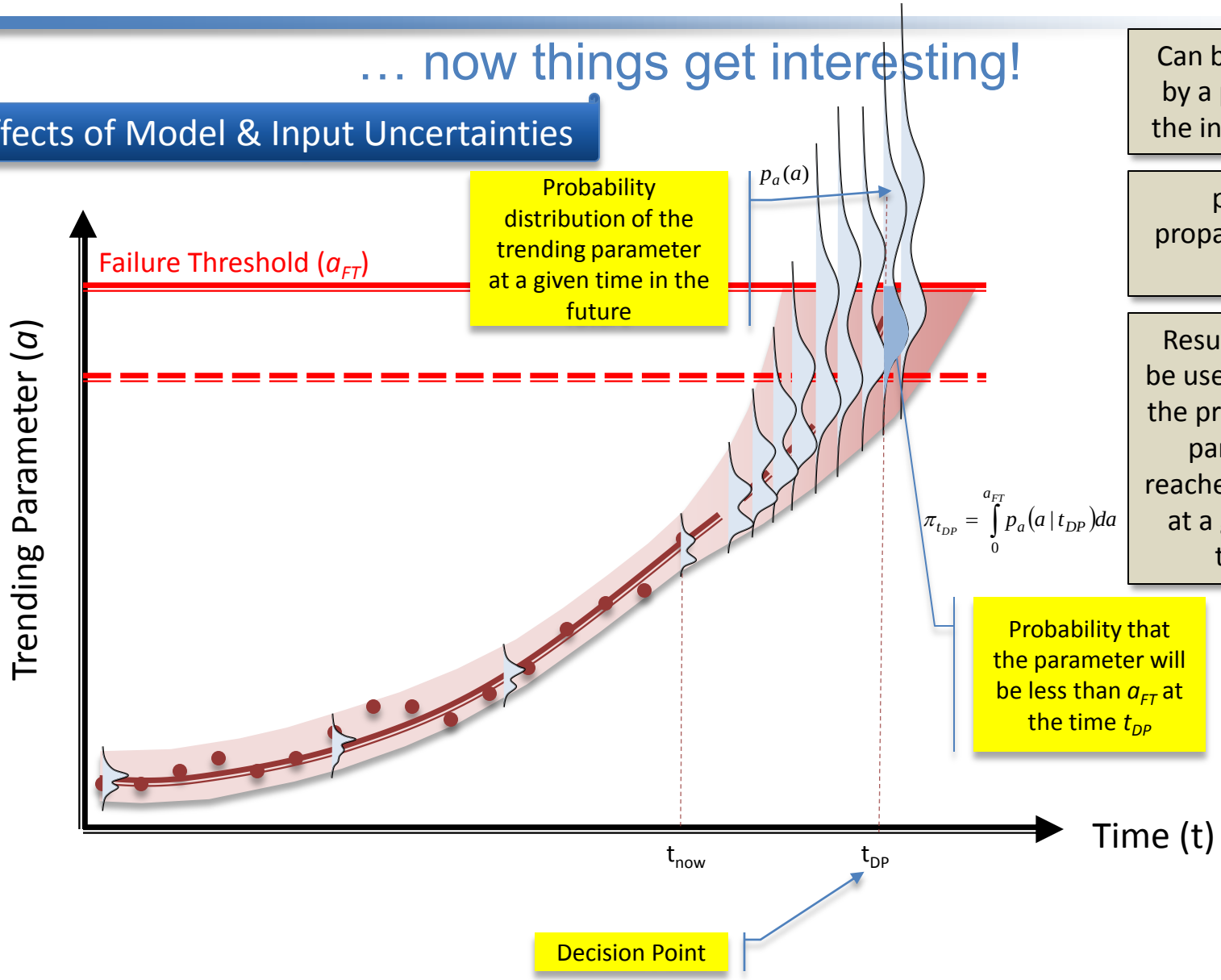
Resulting pdf can be used to determine the probability of EOL occurring between two future time points

Risk vs POF

Trends and Thresholds Revisited

… now things get interesting!

Effects of Model & Input Uncertainties



Can be represented by a pdf describing the initial conditions

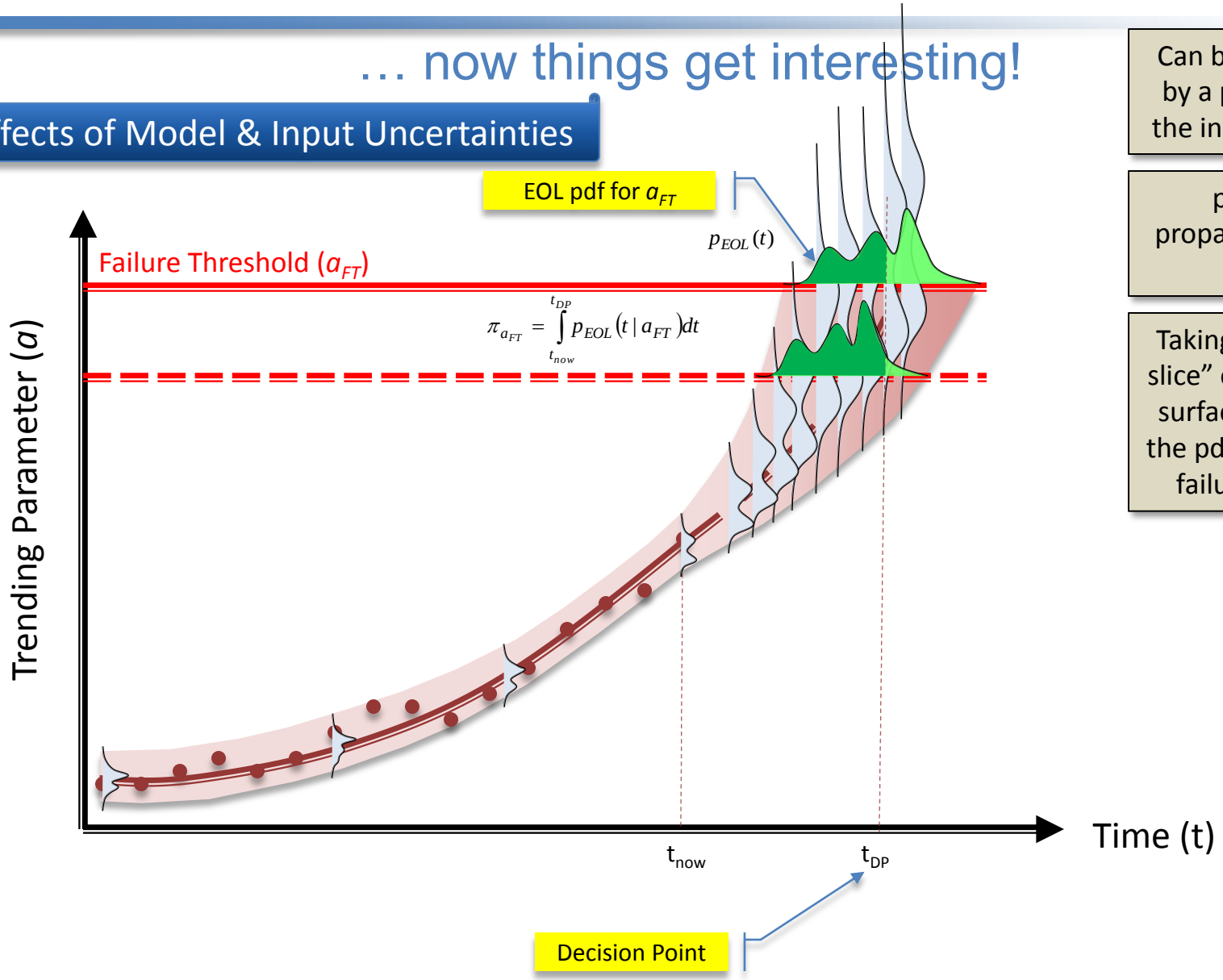
pdf is then propagated forward in time

Resulting pdf's can be used to determine the probability that a parameter has reached a given value at a given point in the future

Trends and Thresholds Revisited

… now things get interesting!

Effects of Model & Input Uncertainties



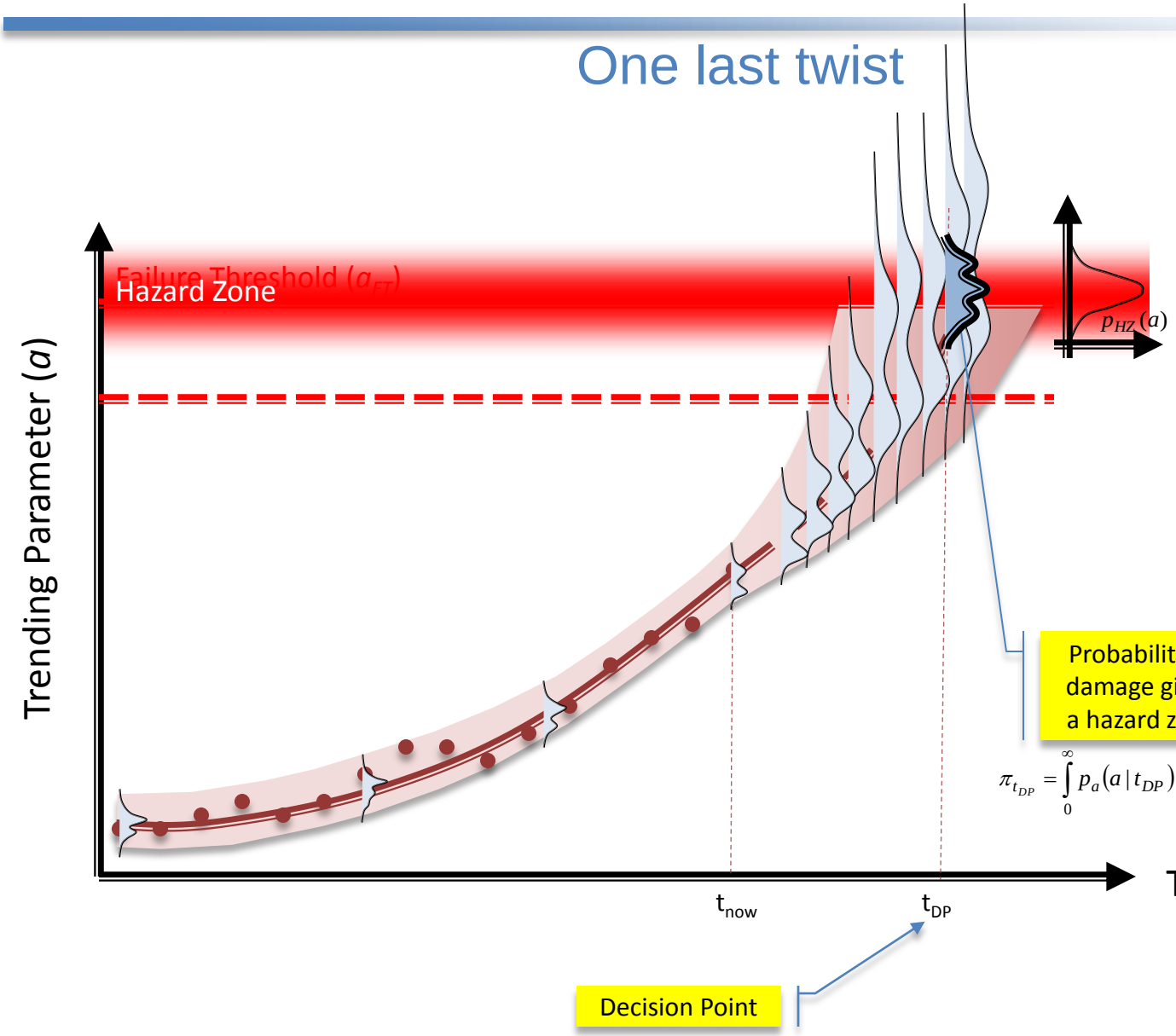
Can be represented by a pdf describing the initial conditions

pdf is then propagated forward in time

Taking a "horizontal slice" of the resulting surface at a_{FT} yields the pdf of EOL at that failure threshold

Trends and Thresholds Revisited

One last twist



Instead of a single value, the threshold could be defined as a distribution – "Hazard Zone'

Probability of damage is now taken as the integral of the product of the two pdf's

Note that "Risk" is now much more difficult to quantify

Prognostic Methods

Data-Based or Physics-Based Models? – That is the question!

Sources of Knowledge

- FMEA / FMECA
  - Failure modes
  - Effects (and Criticality) – which failure modes to go after
- Fault Tree Analysis
  - Propagation Models
- Designers / Reliability Engineers
  - System knowledge and insight
  - Expected / nominal behavior of the system
- Seeded Failure Testing / Accelerated Life Testing
  - Data
  - Failure signatures
  - Effects of environmental conditions
- Fielded Systems
  - Sensors measurements
  - Maintenance logs
  - Fleet Statistics
  - Performance Validation

Data-Driven Methods

- Model is based solely on data collected from the system
- Some system knowledge may still be handy:
  - What the system ‘is’
  - What the failure modes are
  - What sensor information is available
  - Which sensors may contain indicators of fault progression (and how those signals may ‘grow’)
- *General* steps:
  - Gather what information you can (if any)
  - Determine which sensors give good trends
  - Process the data to “clean it up” – try to get nice, monotonic trends
  - Determine threshold(s) either from experience (data) or requirements
  - Use the model to predict RUL
    - Regression / trending
    - Mapping (e.g., using a neural network)
    - Statistics

Data-Driven Method Example

PHM2008 Data Challenge

- No knowledge of system (just a bunch of data)
- 218 sets of data (“runs”)
- 24 Signals
  - 3 described as “operational settings”
  - 21 described as “sensor measurement *n*”
- At the start of each run, the system is healthy
  - Although perhaps not at 100%
- At some point during each run, a fault develops and grows to ‘failure’ at the end of the run

Data-Driven Method Example

PHM2008 Data Challenge

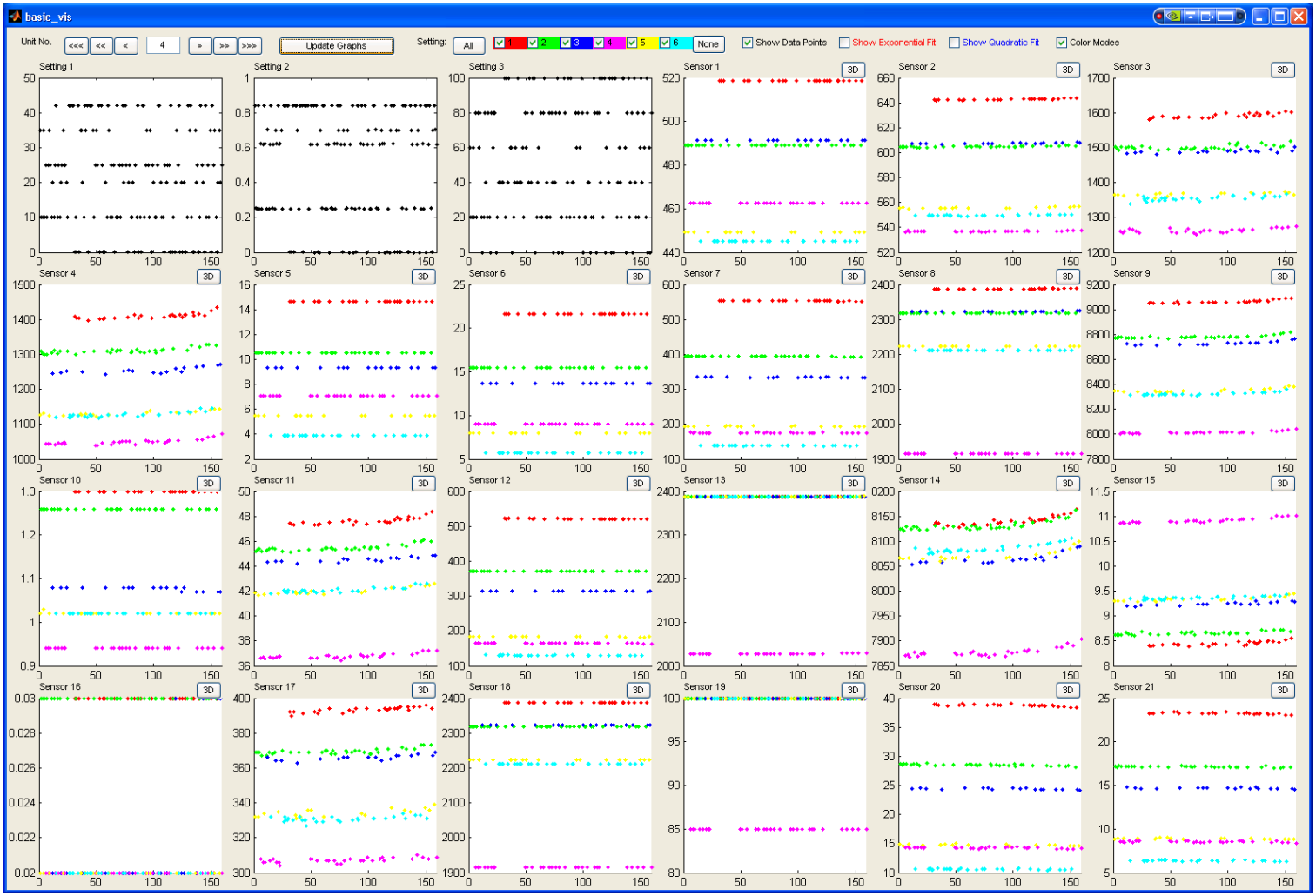
Operational Settings

Use Op Settings to determine different modes of operation

Raw Data Plots for a Single Run

Data-Driven Method Example

PHM2008 Data Challenge

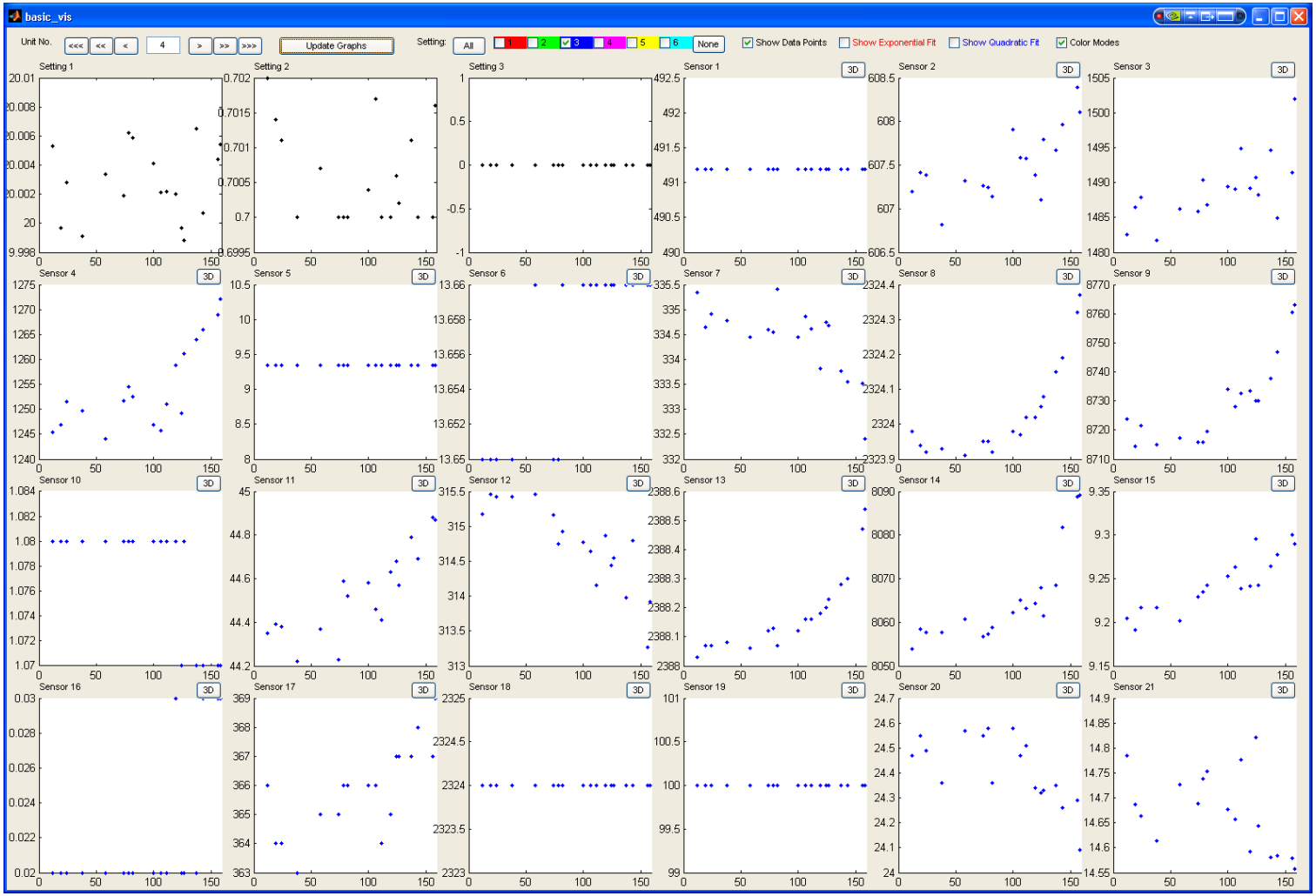


Consider a single mode

Modes Parsed and Highlighted

Data-Driven Method Example

PHM2008 Data Challenge



Let's look at a single sensor

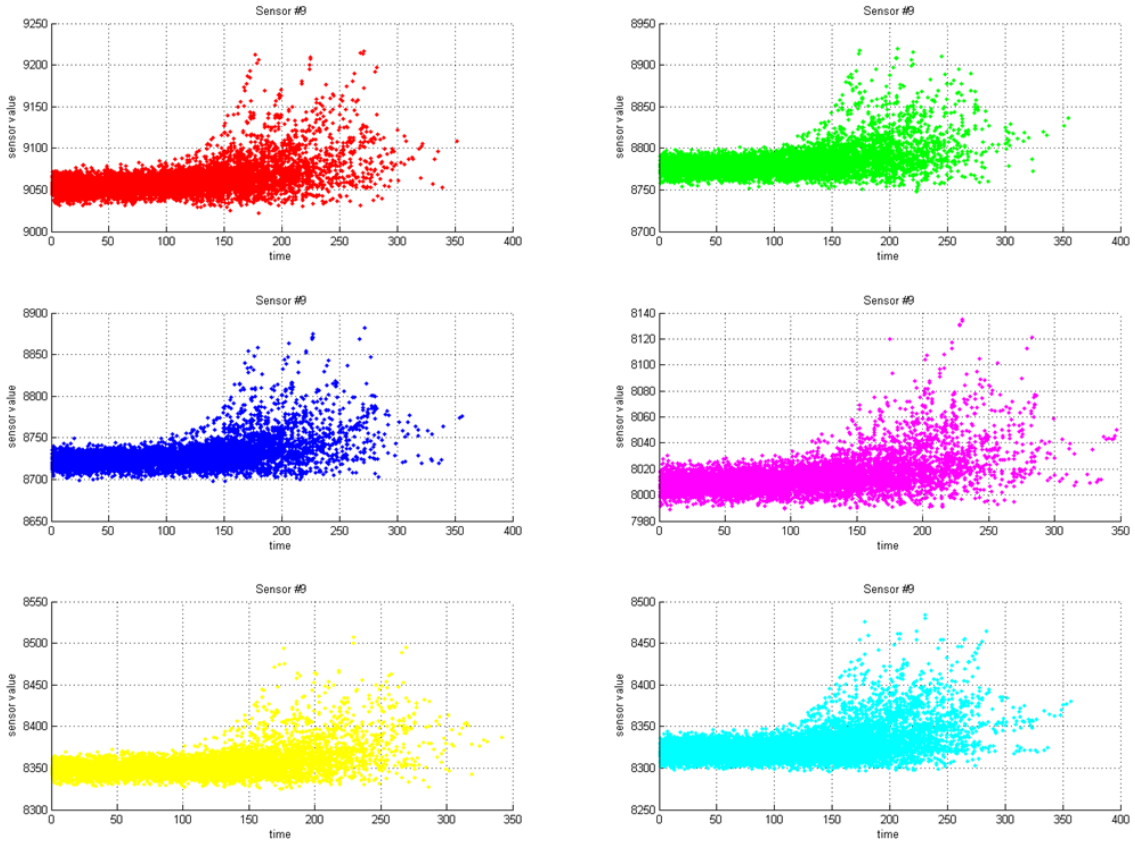
Raw Data Plots for a Single Unit & Mode

Data-Driven Method Example

PHM2008 Data Challenge

Mode 1
Mode 2
Mode 3
Mode 4
Mode 5
Mode 6

Sensor 9



Different sensors show different trends – Op mode and Failure mode dependent

Raw Data Plots for a Single Sensor

Data-Driven Method Example

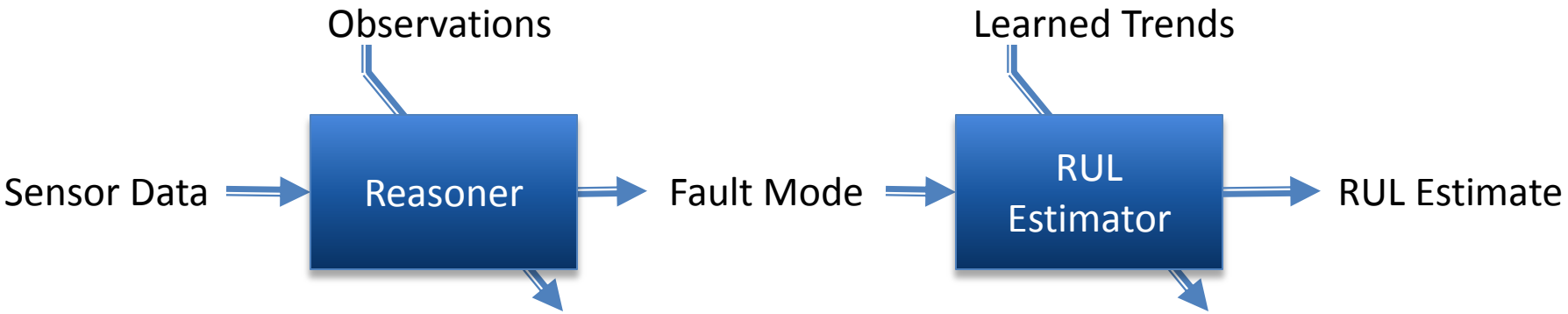
PHM2008 Data Challenge

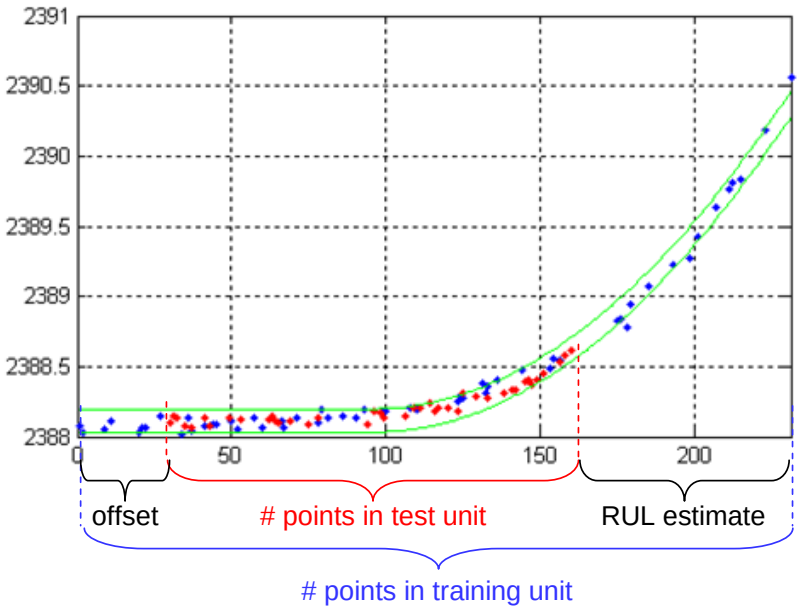
Sensor	Observations
1	Single-valued for each operational setting across all units. No useful information.
2	All operational settings tend to show slight "up" trend as failure progresses.
3	All operational settings tend to show slight "up" trend as failure progresses.
4	All operational settings tend to show slight "up" trend as failure progresses.
5	Single-valued for each operational setting across all units. No useful information.
6	Dual-valued for each operational setting across all units. The lower value (in each operational setting) appears to be confined to the earlier cycles of each unit.
7	All operational settings tend to show slight "down" trend as failure progresses, perhaps slightly more pronounced in operational setting 1.
8	Operational settings 1, 2, and 3 show "up" trend as failure progresses for all units. Operational settings 4, 5, and 6 show a mix of "up" and "down" trends as failure progresses. Many curves appear to be rather pronounced (i.e., sharp up or down trend as failure progresses).
9	All operational settings show "up" trend as failure progresses for most units, though some units appear flat. Many curves appear to be rather pronounced (i.e., sharp up or down trend as failure progresses).
10	Operational settings 1 and 2 are single-value across all units. Operational settings 3, 4, 5, and 6 are dual-valued across all units.
11	All operational settings tend to show slight "up" trend as failure progresses.
12	All operational settings tend to show slight "down" trend as failure progresses, perhaps slightly more pronounced in operational setting 1.
13	Operational settings 1, 2, and 3 show "up" trend as failure progresses for all units. Operational settings 4, 5, and 6 show a mix of "up" and "down" trends as failure progresses. Many curves appear to be rather pronounced (i.e., sharp up or down trend as failure progresses).
14	All operational settings show "up" trend as failure progresses for most units, though some units appear flat. Many curves appear to be rather pronounced (i.e., sharp up or down trend as failure progresses).
15	All operational settings tend to show slight "up" trend as failure progresses.
16	Operational settings 1, 2, 4, 5, and 6 are single-valued across all units. Operational setting 3 is dual-valued across all units with the lower value confined to the earlier cycles of each unit.
17	All operational settings tend to show slight "up" trend as failure progresses. Signals are discrete valued (no fractional values, only integral).
18	Single-valued for each operational setting across all units. No useful information.
19	Single-valued for each operational setting across all units. No useful information.
20	All operational settings tend to show very slight "down" trend as failure progresses, perhaps slightly more pronounced in operational setting 1.
21	All operational settings tend to show very slight "down" trend as failure progresses, perhaps slightly more pronounced in operational setting 1.

Observations → Fault Modes → Reasoner

Data-Driven Method Example

PHM2008 Data Challenge





Data-Driven Methods

Pros & Cons

- Pros
  - Easy and Fast to implement
    - Several off-the-shelf packages are available for data mining
  - May identify relationships that were not previously considered
    - Can consider all relationships without prejudice

- Cons
  - Requires lots of data and a “balanced” approach
    - Most of the time, lots of run-to-failure data are not available
    - Very real risk of “over-learning” the data
    - Conversely, there’s also a risk of “over-generalizing”
  - Results may be counter- (or even un-)intuitive
    - Correlation does not always imply causality!
  - Can be computationally intensive, both for analysis and implementation

- Example techniques
  - Regression analysis
  - Neural Networks (NN)
  - Bayesian updates
  - Relevance vector machines (RVM)

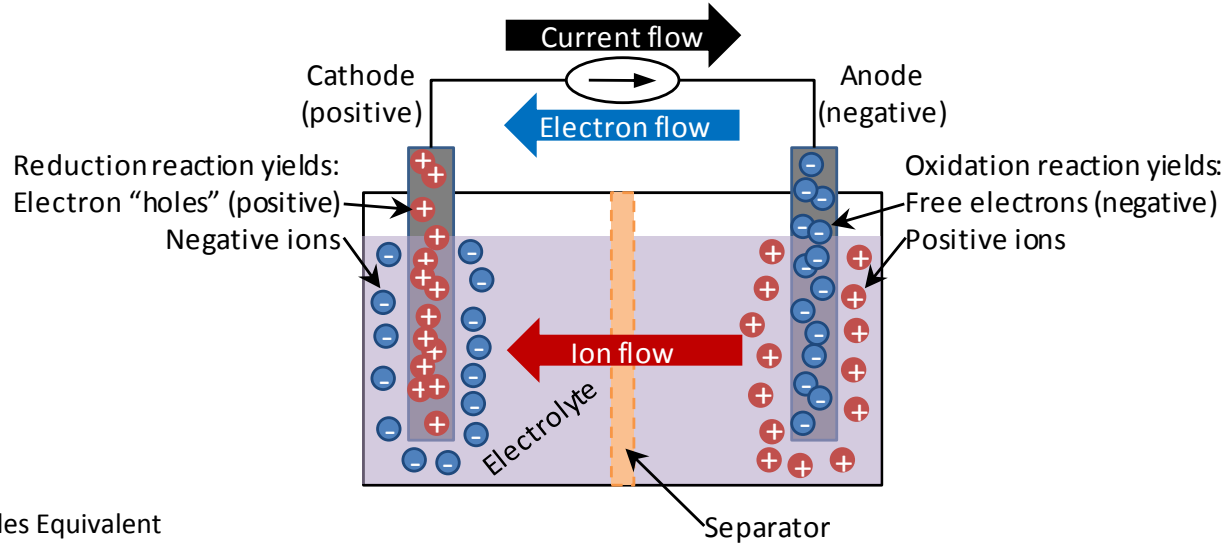
Physics-Based Methods

- Description of a system's underlying physics using suitable representation
- Some examples:
  - Model derived from "First Principles"
    - Encapsulate fundamental laws of physics
      - PDEs
      - Euler-Lagrange Equations
  - Empirical model chosen based on an understanding of the dynamics of a system
    - Lumped Parameter Model
    - Classical 1st (or higher) order response curves
  - Mappings of stressors onto damage accumulation
    - Finite Element Model
    - High-fidelity Simulation Model

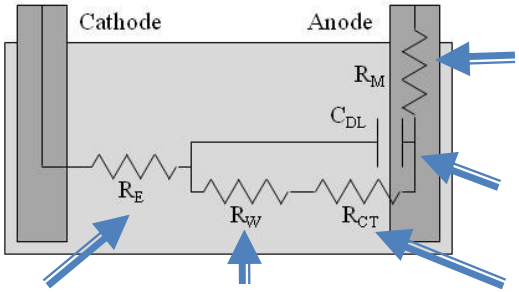
- Something in the model correlates to the failure mode(s) of interest

Physics-Based Method Example

Lithium Ion Battery



Randles Equivalent Impedance Model



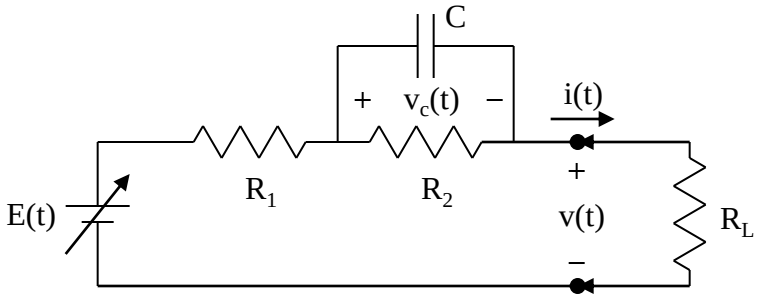
Metal Resistance

Double-Layer Capacitance

Electrolyte Resistance

Warburg (Diffusion) Resistance

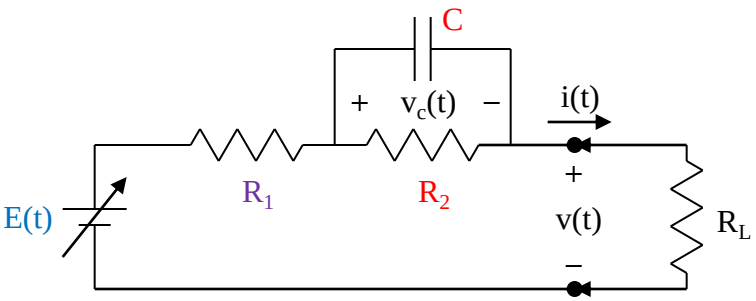
Charge Transfer Resistance



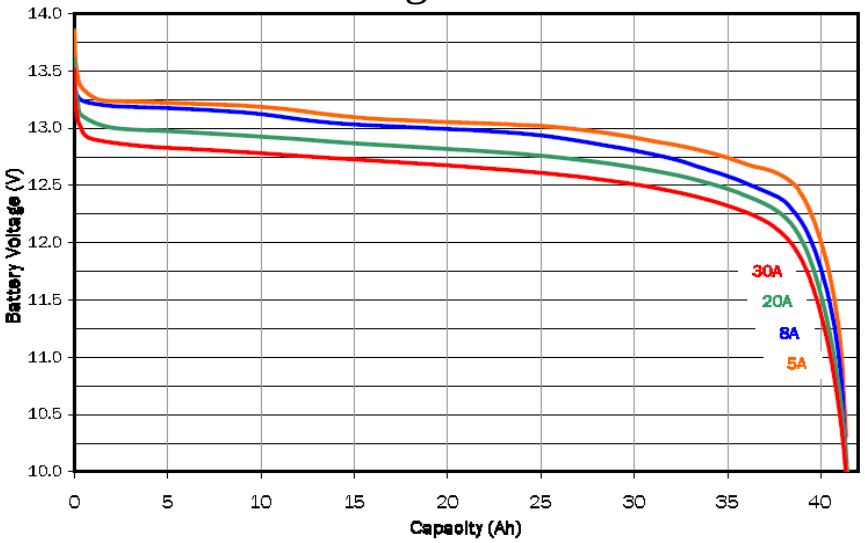
1st Order Lumped Parameter Model

Physics-Based Method Example

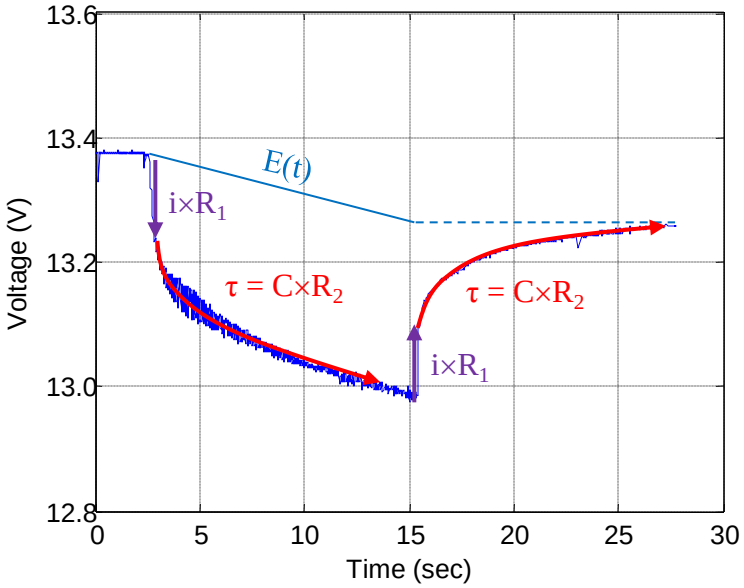
Lithium Ion Battery



Discharge Curves

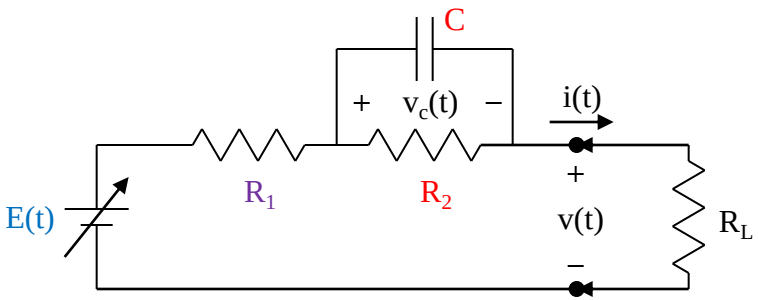


Battery Pulse Data



Physics-Based Method Example

Lithium Ion Battery



- As the battery ages, changes in the electro-chemical properties manifest in changes to R_1, R_2, and C
- Usage and/or BIT data are used to continuously estimate the impedance values
- Regression analysis is used to correlate the impedance values to battery capacity (State of Health)

Physics-Based Models

Pros & Cons

- Pros
  - Results tend to be intuitive
    - Based on modeled phenomenon
    - And when they're not, they're still instructive (e.g., identifying needs for more fidelity or unmodeled effects)
  - Models can be reused
    - Tuning of parameters can be used to account for differences in design
  - If incorporated early enough in the design process, can drive sensor requirements (adding or removing)
  - Computationally efficient to implement

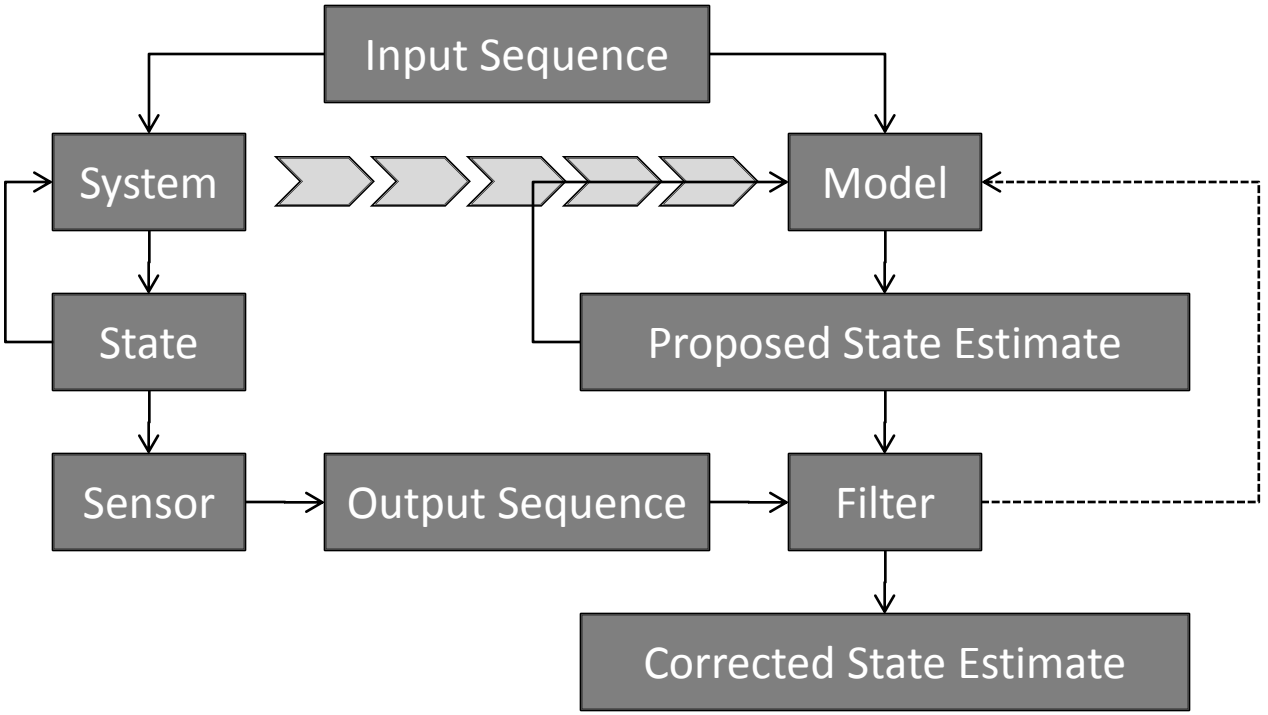
- Cons
  - Model development requires a thorough understanding of the system
  - High-fidelity models can be computationally intensive

- Examples
  - Paris-Erdogan Crack Growth Model
  - Taylor tool wear model
  - Corrosion model
  - Abrasion model

Algorithms

- Assess the current damage state
- Utilize physics-based model in prescribed fashion for prediction
- Use the model in forward mode
  - Take future conditions into consideration
  - Explore different paths
  - Handle uncertainty (unlikely to be Gaussian)
- Criteria for Algorithm Choice (example)
  - Estimate the state in the presence of noise
    - Use algorithms such as filters.
  - Deal with system nonlinearities and non-parametric noise
    - Use suitable techniques such as Monte-Carlo.
  - Run real-time
    - Use appropriate algorithms such as Particle Filters (Markov Chain Monte Carlo).

A “Filter” Approach



Damage Estimation with Particle Filters

- Particle filters (PFs) are state observers that can be applied to general nonlinear processes with non-Gaussian noise
  - Approximate state distribution by set of discrete weighted samples:

$$\{(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i), w_k^i\}_{i=1}^N$$

  - Suboptimal, but approach optimality as N→ ∞
- Parameter evolution described by random walk:

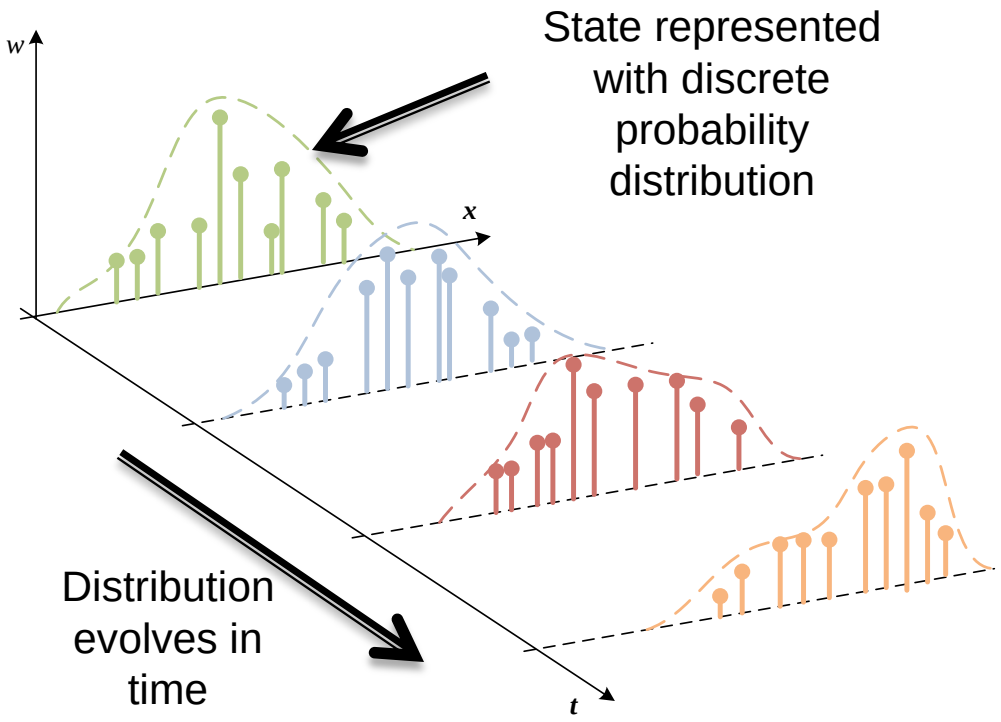
$$\theta_k = \theta_{k-1} + \xi_{k-1}$$

  - Selection of variance of random walk noise is important
  - Variance must be large enough to ensure convergence, but small enough to ensure precise tracking
- PF approximates posterior as

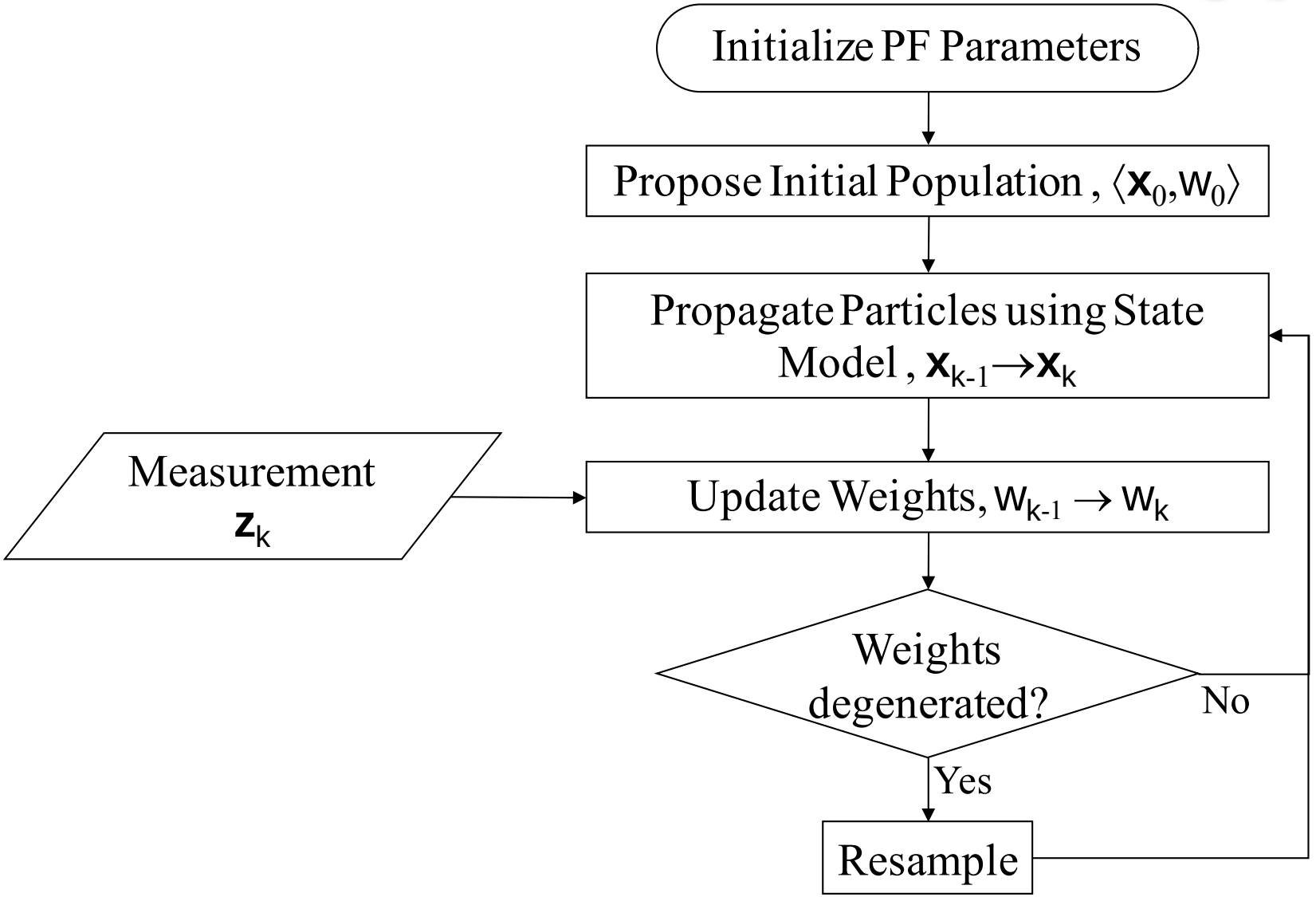
$$p(\mathbf{x}_k, \boldsymbol{\theta}_k | \mathbf{y}_{0:k}) \approx \sum_{i=1}^{N} w_k^i \delta_{(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i)}(d\mathbf{x}_k d\boldsymbol{\theta}_k)$$

Particle Filters

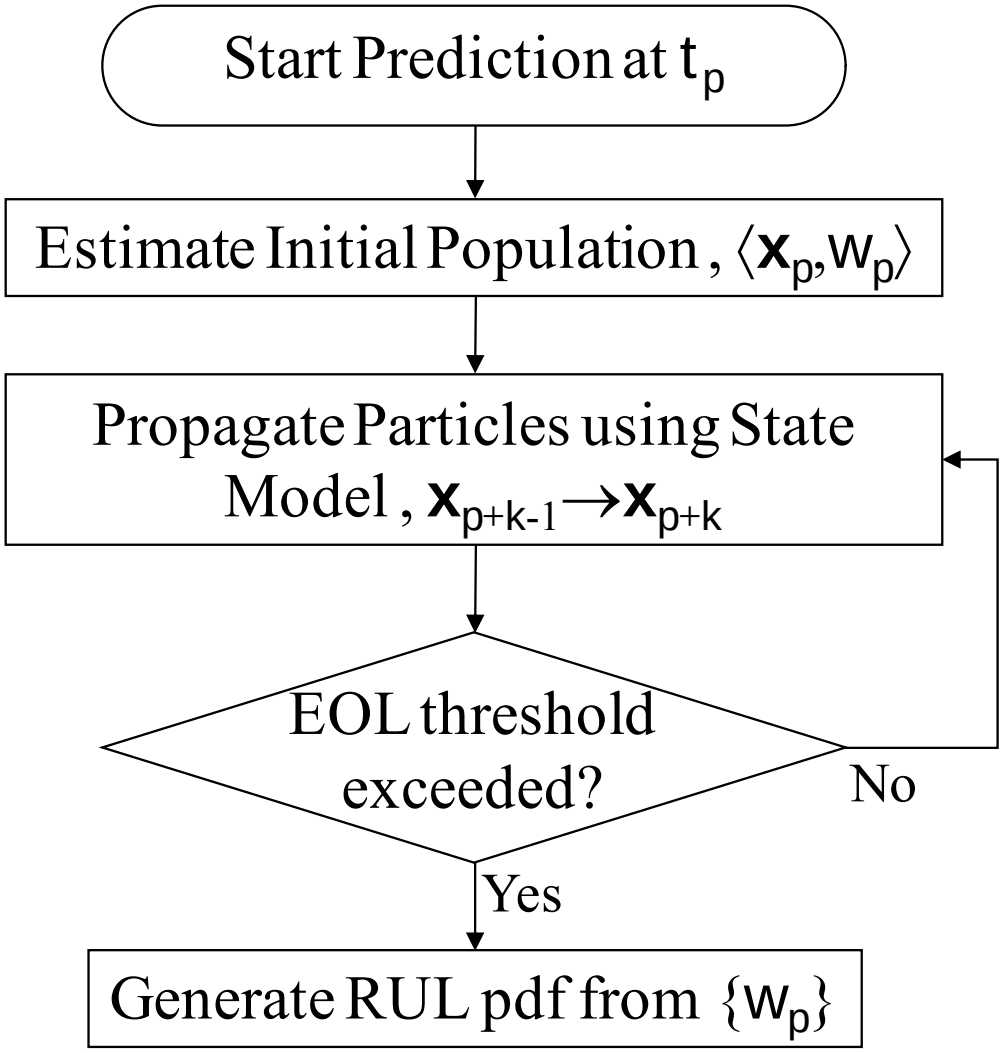
- Employ *particle filters* for joint state-parameter estimation
  - Represent probability distributions using set of weighted samples
  - Help manage uncertainty (e.g., sensor noise, process noise, etc.)
  - Similar approaches have been applied successfully to actuators, batteries, and other prognostics applications



Particle Filter (PF) Tracking



Particle Filter Prediction



Prediction

- General idea
  - Propagate each particle forward until EOL reached (condition on EOL evaluates to true)
  - Use particle weights for EOL weights
- Particle filter computes

$$p(\mathbf{x}_{k_P}, \boldsymbol{\theta}_{k_P} | \mathbf{y}_{0:k_P}) \approx \sum_{i=1}^{N} w_{k_P}^i \delta_{(\mathbf{x}_{k_P}^i, \boldsymbol{\theta}_{k_P}^i)} (d\mathbf{x}_{k_P} d\boldsymbol{\theta}_{k_P})$$

- Prediction n steps ahead approximated as

$$p(\mathbf{x}_{k_P+n}, \boldsymbol{\theta}_{k_P+n} | \mathbf{y}_{0:k_P}) \approx \sum_{i=1}^{N} w_{k_P}^i \delta_{(\mathbf{x}_{k_P+n}^i, \boldsymbol{\theta}_{k_P+n}^i)} (d\mathbf{x}_{k_P+n} d\boldsymbol{\theta}_{k_P+n})$$

- Similarly, EOL prediction approximated as

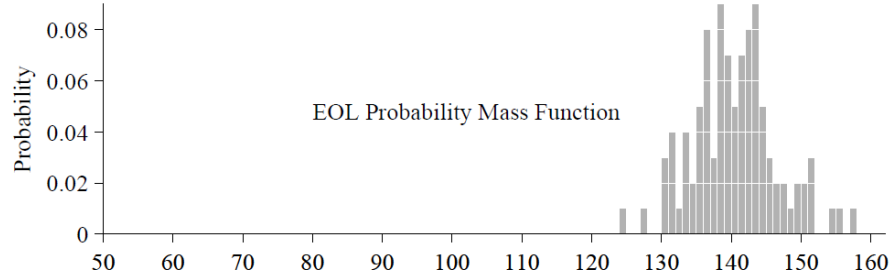
$$p(EOL_{k_P} | \mathbf{y}_{0:k_P}) \approx \sum_{i=1}^{N} w_{k_P}^i \delta_{EOL_{k_P}^i} (dEOL_{k_P})$$

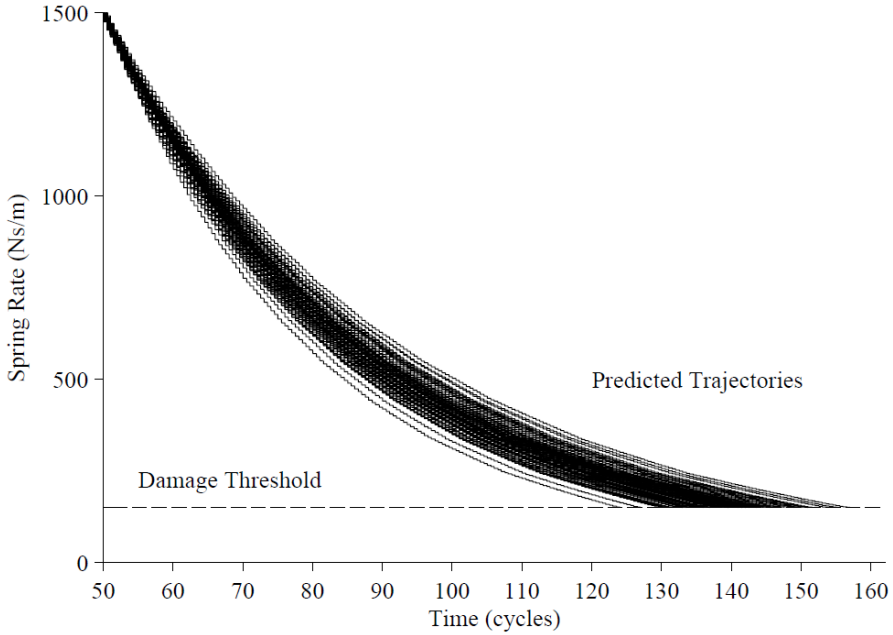
Prediction

Algorithm 2 EOL Prediction

Inputs: $\{(\mathbf{x}^i_{k_P}, \boldsymbol{\theta}^i_{k_P}), w^i_{k_P}\}_{i=1}^N$
Outputs: $\{EOL^i_{k_P}, w^i_{k_P}\}_{i=1}^N$
for $i = 1$ **to** N **do**
  $k \leftarrow k_P$
  $\mathbf{x}^i_k \leftarrow \mathbf{x}^i_{k_P}$
  $\boldsymbol{\theta}^i_k \leftarrow \boldsymbol{\theta}^i_{k_P}$
  while $T_{EOL}(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k) = 0$ **do**
    Predict $\hat{\mathbf{u}}_k$
    $\boldsymbol{\theta}^i_{k+1} \sim p(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}^i_k)$
    $\mathbf{x}^i_{k+1} \sim p(\mathbf{x}_{k+1}|\mathbf{x}^i_k, \boldsymbol{\theta}^i_k, \hat{\mathbf{u}}_k)$
    $k \leftarrow k + 1$
    $\mathbf{x}^i_k \leftarrow \mathbf{x}^i_{k+1}$
    $\boldsymbol{\theta}^i_k \leftarrow \boldsymbol{\theta}^i_{k+1}$
  end while
  $EOL^i_{k_P} \leftarrow k$
end for

Hypothesized inputs





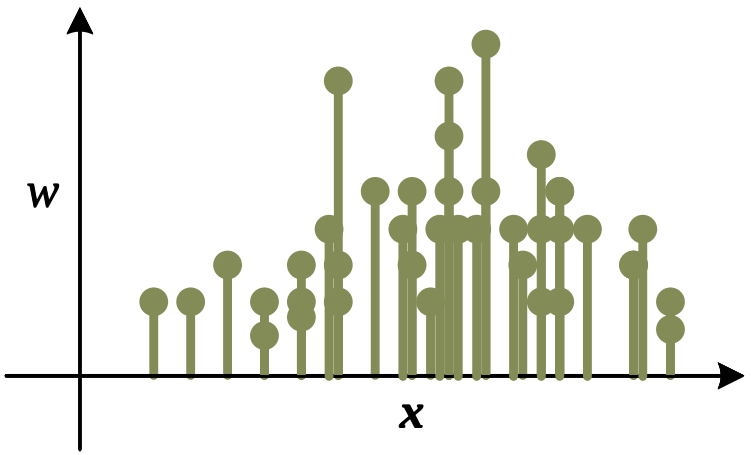
Spring damage progression EOL prediction

Sampling Importance Resampling (SIR) PF

- Begin with initial particle population
- Predict evolution of particles one step ahead
- Compute particle weights based on likelihood of given observations
- Resample to avoid degeneracy issues
  - Degeneracy is when small number of particles have high weight and the rest have very low weight
  - Avoid wasting computation on particles that do not contribute to the approximation

Algorithm 1 SIR Filter

Inputs: $\{(\mathbf{x}^i_{k-1}, \boldsymbol{\theta}^i_{k-1}), w^i_{k-1}\}^N_{i=1}, \mathbf{u}_{k-1:k}, \mathbf{y}_k$
Outputs: $\{(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k), w^i_k\}^N_{i=1}$
for $i = 1$ **to** N **do**
  $\boldsymbol{\theta}^i_k \sim p(\boldsymbol{\theta}_k|\boldsymbol{\theta}^i_{k-1})$
  $\mathbf{x}^i_k \sim p(\mathbf{x}_k|\mathbf{x}^i_{k-1}, \boldsymbol{\theta}^i_{k-1}, \mathbf{u}_{k-1})$
  $w^i_k \leftarrow p(\mathbf{y}_k|\mathbf{x}^i_k, \boldsymbol{\theta}^i_k, \mathbf{u}_k)$
end for
$W \leftarrow \sum_{i=1}^{N} w^i_k$
for $i = 1$ to N **do**
  $w^i_k \leftarrow w^i_k / W$
end for
$\{\mathbf{x}^i_k, \boldsymbol{\theta}^i_k, w^i_k\}^N_{i=1} \leftarrow \texttt{Resample}(\{\mathbf{x}^i_k, \boldsymbol{\theta}^i_k, w^i_k\}^N_{i=1})$



Initialization

Hybrid Models

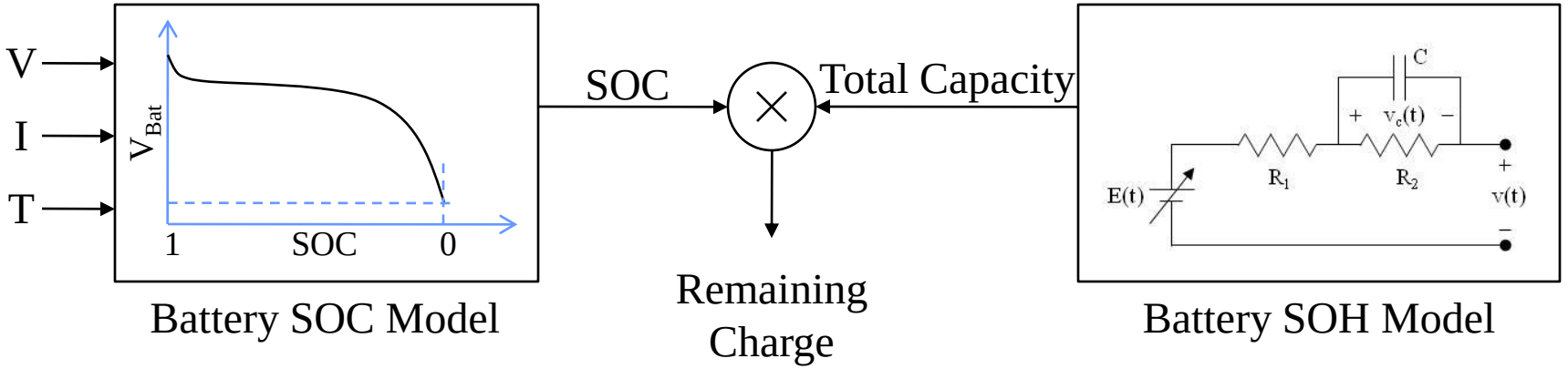
The best of both worlds

- In practice, many implementations pull from both Data-Driven and Physics-Based Model methods
  - Use data to learn model parameters
  - Use knowledge about the physical process to determine the type of regression analysis to apply (linear, polynomial, exponential, etc.)
  - Data-Driven System Model in conjunction with a Physics-Based Fault Model (or vice-versa)
  - Identify potential correlations physics model and correlate using a data-based approach
  - Data fusion – have one of each!

Hybrid Example

Lithium Ion Battery Revisited

- Regression analysis used to trend circuit parameters (R_1, R_2, C)
- Battery State of Health (SOH) Model
  - Correlates total charge capacity to SOH
- Battery State of Charge (SOC) Model
  - Correlates voltage, current, and temperature to SOC
- Together they can yield both the life remaining on the current charge as well as when the battery will need to be replaced



Hybrid Models

Pros & Cons

- Pros
  - Combines the strengths of each approach
  - Robustness in design
    - Use data where system knowledge is lacking
    - Use physics where data is lacking
  - Results are both intuitive and match observations
  - Can “mix and match” approaches to customize for the current situation

- Cons
  - Though the goal of a hybrid approach is to pull the best from each approach, where each approach is used, it still carries its disadvantages
    - Need for data
    - Portions may still be computationally intensive
    - Need for in-depth system knowledge

- Examples
  - Particle Filters
  - Kalman Filters
  - etc.

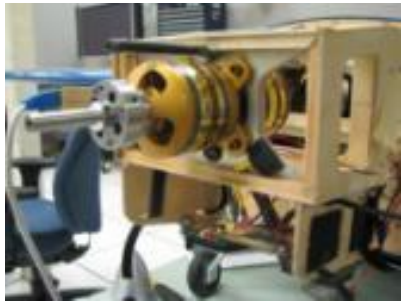
Application Example

Putting it al together

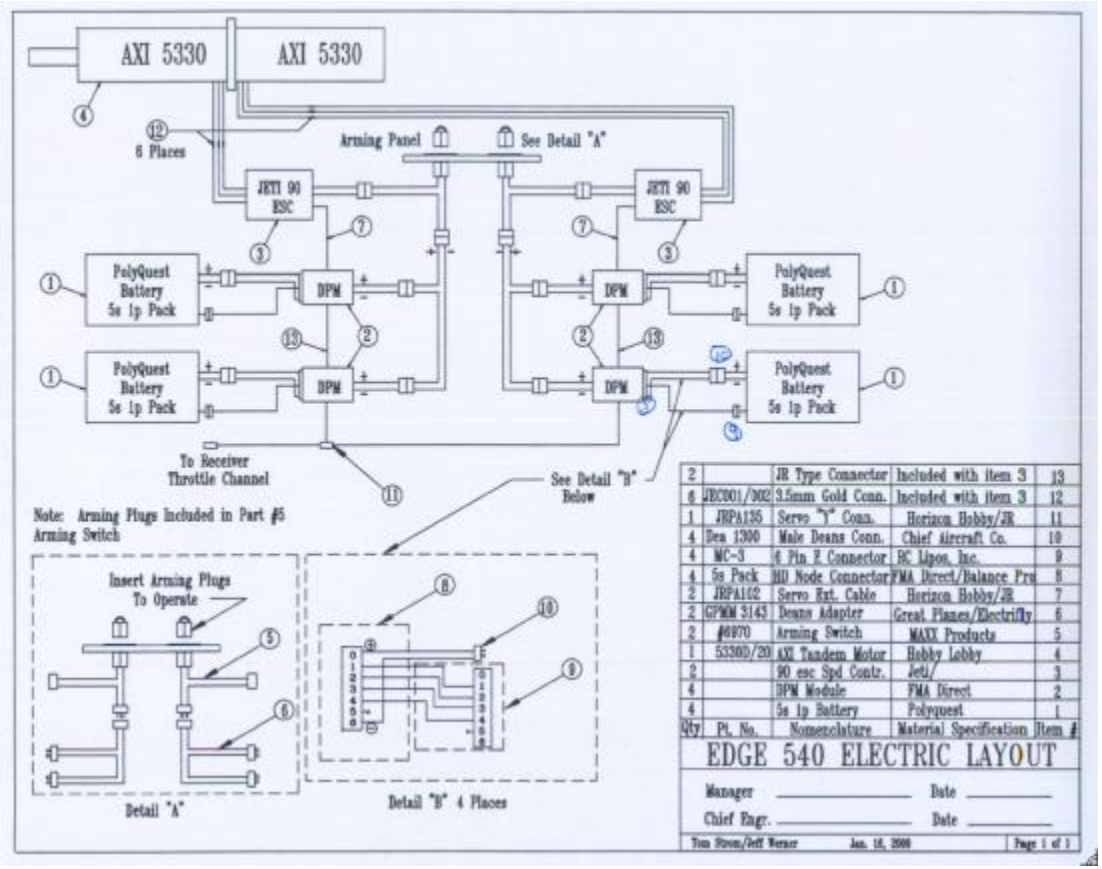
Prognostics for Electric UAV

- Flight Tests with unmanned electric aircraft
  - Battery performance not as anticipated
  - SOC and remaining life information desirable during operations

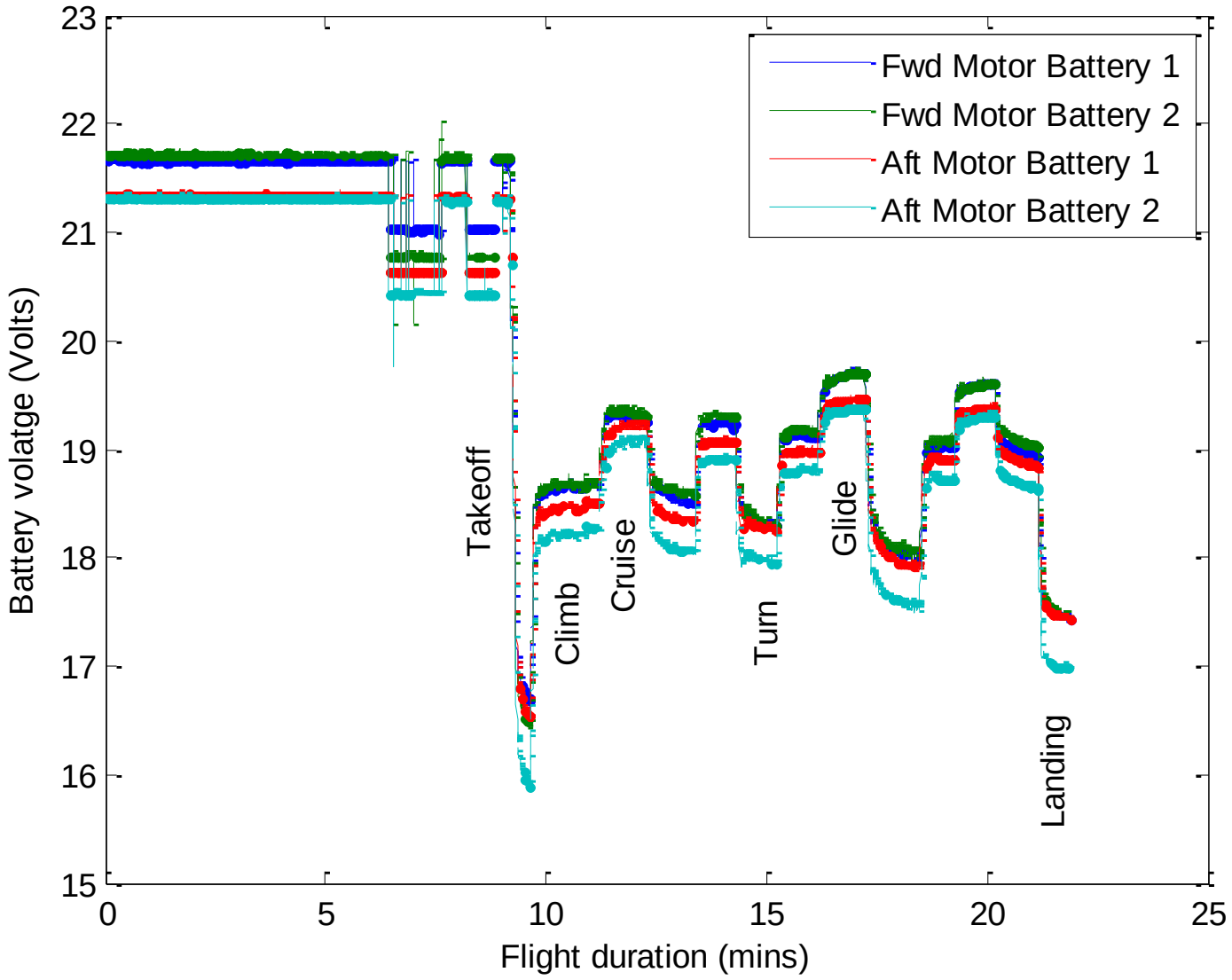
Edge 540 Electric UAV



- 2 motors
- 4 batteries (each with 5 cells in series)
- 2 batteries in series power each motor



Flight Profile



Model Adaptation

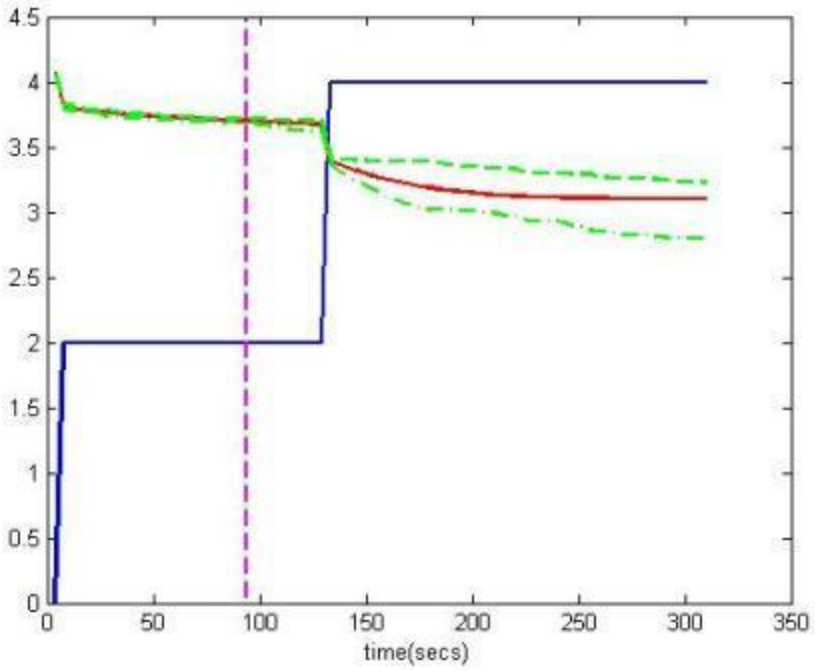
- Modified system model:

$$\mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \boldsymbol{\alpha}_{k-1}\right) + \omega_{k-1}$$

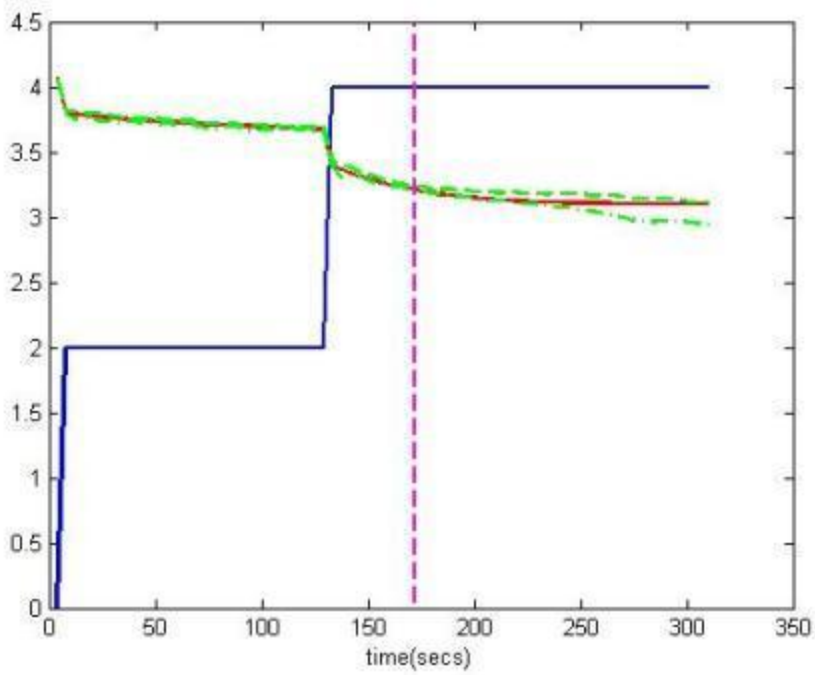
$$\boldsymbol{\alpha}_k = \mathbf{g}\left(\boldsymbol{\alpha}_{k-1}\right) + \zeta_{k-1}$$

$$\mathbf{z}_k = \mathbf{h}\left(\mathbf{x}_k\right) + \nu_{k-1}$$

Model parameters



Future load profile is known
Prediction before model adaptation to changing load



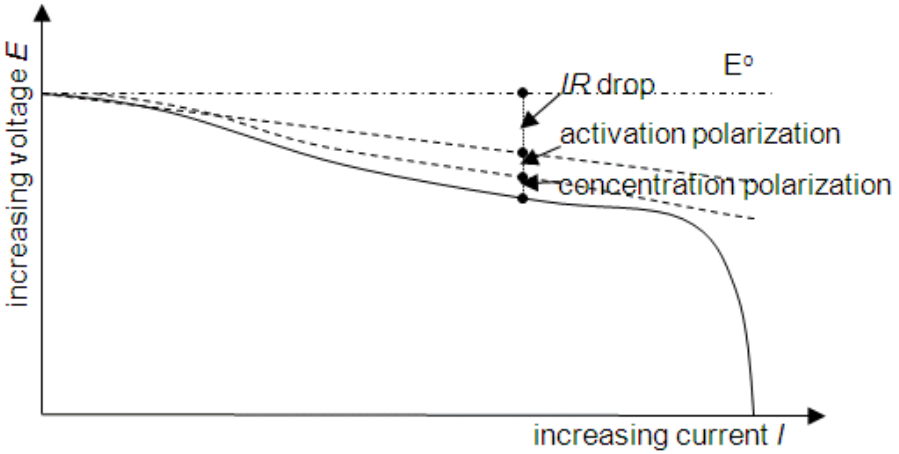
Future load profile is known
Prediction after model adaptation to changing load

Adapting Model Parameters

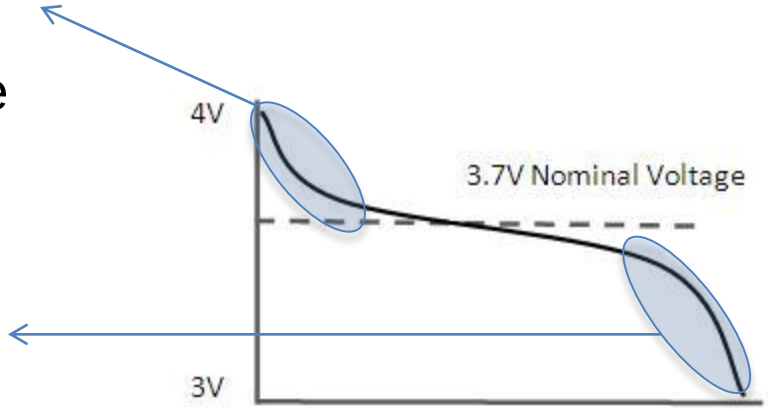
- Easy solution: is to pick a *Gaussian random walk*
- Update model/system noise variance based on convergence
- Have rules to update model parameters instead of just a random walk
  - May be based on sensitivity analysis of parameters
- The more complicated the parameter update scheme, the higher the computational load
  - Needs to be justified by performance requirements
- Need to explore the model design space

Modeling SOC

- IR drop
  - Effect of lumped passive elements, R, C
  - Temperature effect on ion-mobility and hence R



- Activation polarization (AP)
  - Affects initial discharge curve



- Concentration polarization (CP)
  - Affects final discharge curve

Empirical Model

Cell voltage $$E(t_k) = E^o - \Delta E_{IRC}(t_k) - \Delta E_{AP}(t_k) - \Delta E_{CP}(t_k)$$

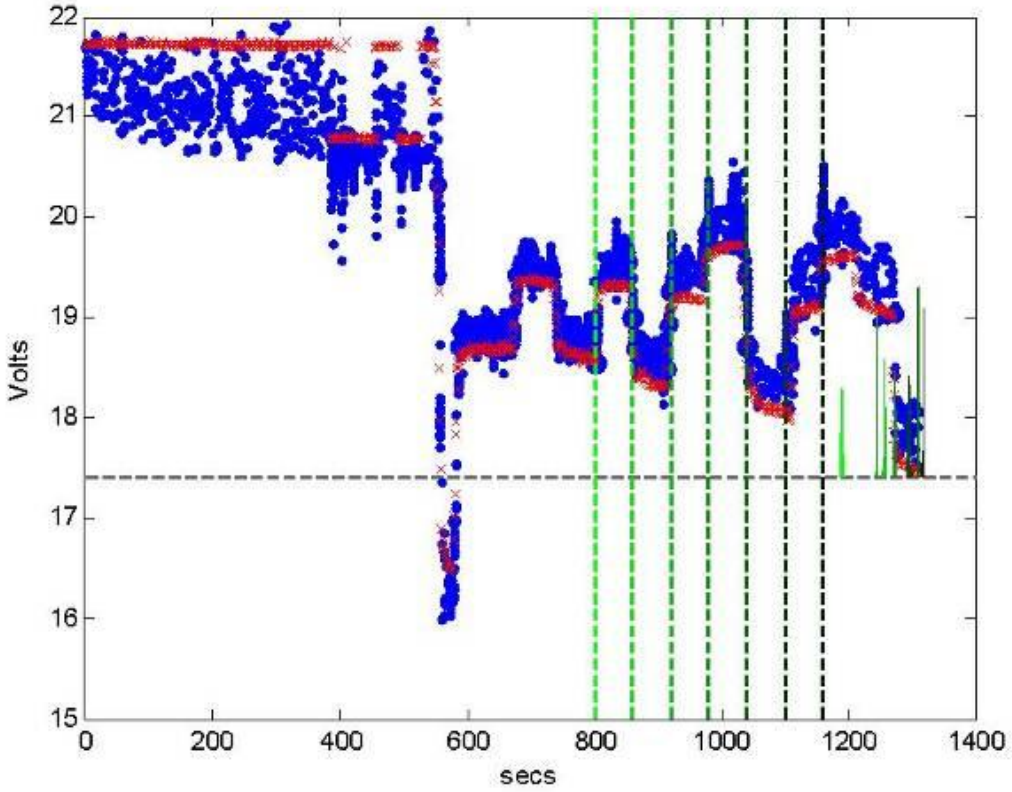
where $$\Delta E_{IRC}(t_k) = \Delta I_k \alpha_6 (1 - \exp(-\alpha_7 (t_k - t_{\Delta I_k}))) - \alpha_1 t_k ,$$

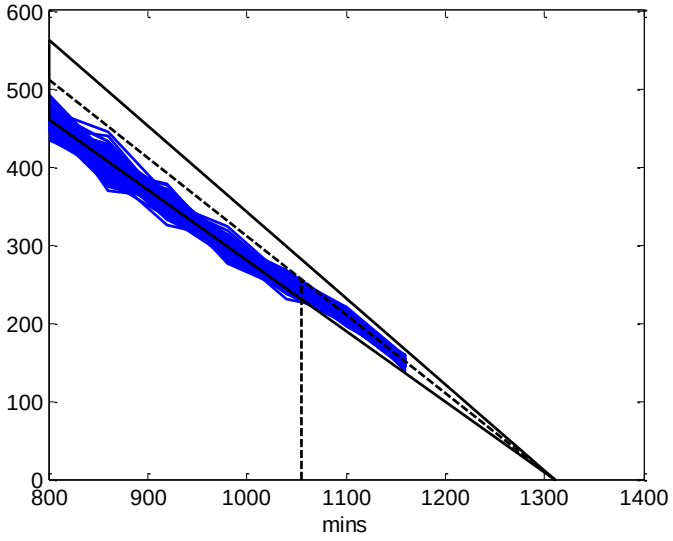
$$\Delta E_{AP}(t_k) = \alpha_{2,k} \ln(1 + \alpha_{3,k} I_k t_k) ,$$

$$\Delta E_{CP}(t_k) = \alpha_{4,k} \exp(\alpha_{5,k} I_k t_k) .$$

- This level of granularity takes various types of battery behavior into account
  - Parameters of the model are load dependent
  - The R-C response characteristic for changes in load is modeled
- Compared to simpler models, this gives higher accuracy in model output
  - but a corresponding increase in the number of parameters

Results





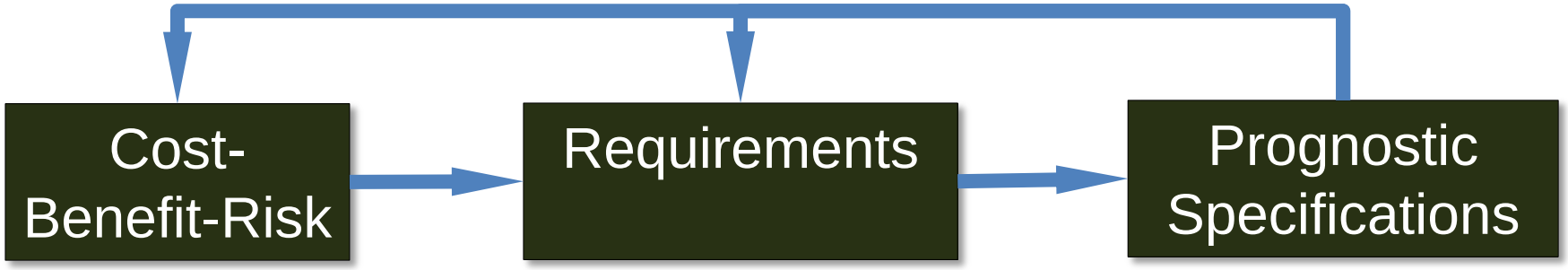
- For statistical validation, we ran model 100 times over the same data
- Parameters chosen
  - α value is chosen to be 0.1
  - λ is chosen to be 0.5
  - i.e. prediction trajectories need to be within 90% accuracy with 50% battery life left).

Other Considerations

Requirements

- Specifications for prognostics
  - Algorithm developers need specifications to which they can build their solution
  - Such specifications require metrics that allow acceptance checking
- Some possible parameters (engineering perspective)
  - Lead time to specify the amount of advanced warning needed for appropriate actions
  - Maximum tolerable probability of proactive maintenance to bound unnecessary maintenance
  - Maximum allowable Probability of Failure (PoF) to bound risk

Deriving Requirements for PHM



Objectives
- Cost-value proposition of PHM
- PHM Design
- Optimal maintenance policy
- Comparison of PHM approaches
- Determine tolerance limits on input uncertainty for desired performance
- Optimal life cycle cost

Inputs
- Mission goals and objectives
- Constraints on resources and time
- Cost function

Integrate from various levels
- Mission planning level
- Maintenance level
- Operation (runtime) level
- Fleet vs. system levels
- …

Process
- Requirement gathering
- Requirement analysis & conflict resolution
- Requirement prioritization
- Requirement flow down

Methods
- Simulation techniques
- Cost-value optimization
- Sensitivity & Pareto frontiers
- Use of prognostics metrics
- Requirements flow down methodology suitably adapted for PHM
- Techniques for uncertainty
  - Representation
  - Management
  - Quantification

Cost Benefit Analysis

Other Contextual Factors

Situational Cost Factors
- Usage profile
- Type of system
- Type of mission
- Operational environment
- Maintenance structure
- …

Computation Basis
- Cost per unit
- Cost for fleet
- Life Cycle Cost
- Cost per contract period
- Annual cost
- Cost per operational hour
- …

Size and Time Scalability
- Fleet size
- Period of monitoring
- Capital discount rates
- …

Cost / Savings

Cost w/o PHM
- Cost of Contingencies
  - Loss of system/life
  - Downtime
  - Cascaded Contingencies
- Cost of Extra Inventory

Cost with PHM
- Cost of PHM Implementation
  - Development (non-recurring)
    - **Non-recurring cost factors**
      - Algorithm development
      - Hardware/Software design
      - Engineering
      - V&V and testing
      - Qualification/certification
      - …
  - Deployment (recurring)
    - **Recurring cost factors**
      - Support and maintenance
      - Equipment and personnel
      - …
- Cost incurred due to PHM
  - **Cons of PHM**
    - Unused component life
    - False positives
- Savings due to PHM
  - **Savings due to reduced**
    - Spare components
    - Manpower (direct/indirect)
    - Training costs
    - Rate of major accidents
    - Footprints
    - System downtime
    - …

PHM Attributes

Prognostic Performance
- Prediction horizon
- Prediction accuracy
- Prediction precision
- Algorithm coverage

Risk and Uncertainty
- Failure rates
- Future loading conditions
- Logistics efficiency
- …

Cost Benefit Analysis

- Main ideas
  - Return on Investment (ROI)
    - ROI = (Return – Investment) / Investment
  - Cost Savings by Implementing PHM
    - $Savings_{PHM} = Cost_{without\ PHM} – Cost_{with\ PHM}$
  - Cost assignments based on past maintenance records, account logs etc.
    - Cost incurred due to similar components in legacy systems
    - Cost of man hours based on direct/indirect staffing requirements in the past
    - Inflation adjustments, etc.
- Formulate as multi-objective optimization problem
- Factor usually not considered: “when to take an action”
  - Cost of early replacement – *function of Prediction Horizon*
  - Confidence in prognostic algorithm – *function of uncertainty management*
  - Risk absorbing capacity – *function of criticality & confidence in prognostic algorithm*

Requirement Flowdown

Translate broad customer requirements into more easily quantified requirements

- Customer oriented view
  - E.g. CTQ and QFD
- Designer/developer oriented view
  - E.g. “Vee” Model and NASA’s System Engineering Engine
- Most popular methods
  - **CTQ Tree**: Critical-to-Quality tree for quality focused methodology e.g. six-sigma
  - **QFD:** Quality Function Deployment to translate customer requirement into engineering specifications

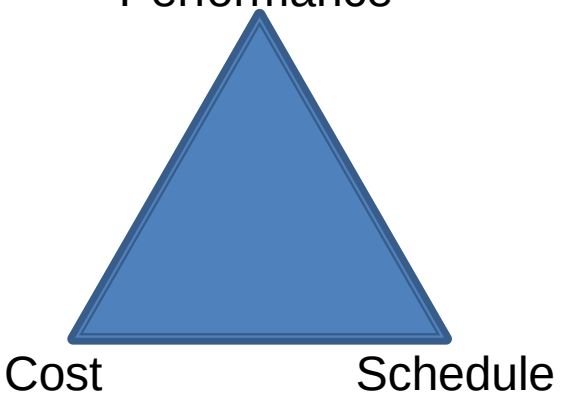
QFD Tools

- Affinity diagrams
- Relations diagrams
- Hierarchy trees
- Process decision program diagrams
- Analytic hierarchy process
- Blueprinting
- House of quality

House of Quality

- Customer requirements
- Technical requirements
- Planning matrix
- Interrelationship matrix
- Technical correlation (roof) matrix
- Technical priorities, benchmarks, and targets

UAV Example: High-Level Considerations



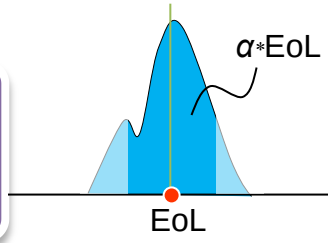
- Performance requirements
  - Run mission safely for 20 minutes
  - Risk of loss < 4%
- Cost Requirements
  - Overall budget
  - Acceptable monetary loss due to incomplete mission (tests) or loss of equipment
- Schedule Requirements
  - Implementation schedule
  - other

Prognostic Performance Metrics

- Metrics Hierarchy

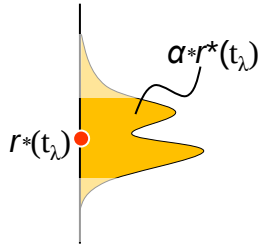
I. Prognostic Horizon

- Does the algorithm predict within desired accuracy around EoL and sufficiently in advance?



II. α-λ Performance

- Further does the algorithm stay within desired performance levels relative to RUL at a given time?



III. Relative Accuracy

- Quantify how well an algorithm does at a given time relative to RUL

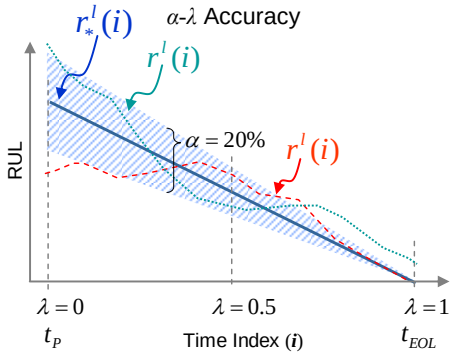
IV. Convergence Rate

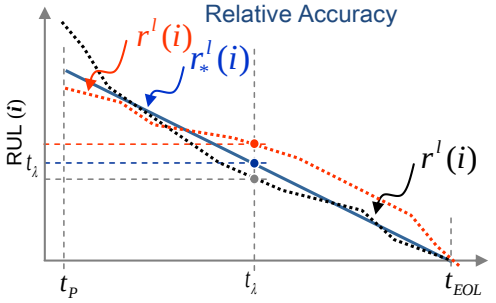
- If the performance converges (i.e. satisfies above metrics) quantify how fast does it converge

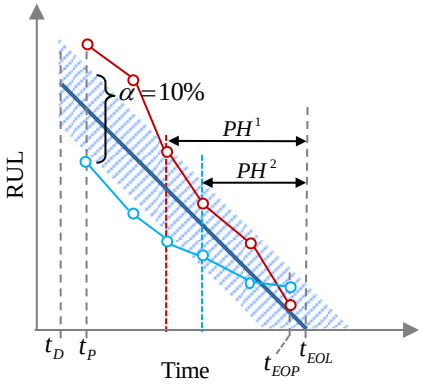
Prognostic Performance Metrics

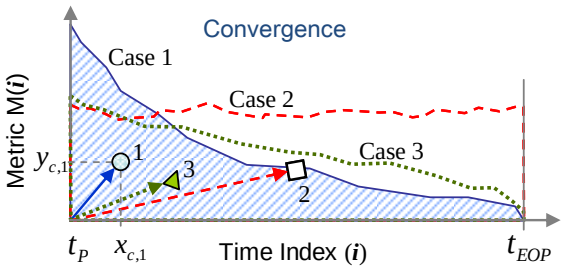
- Metrics have been developed specific to Prognostics for ISHM
- These metrics were applied to
  - A combination of different algorithms and different datasets
- Metrics were evaluated and refined

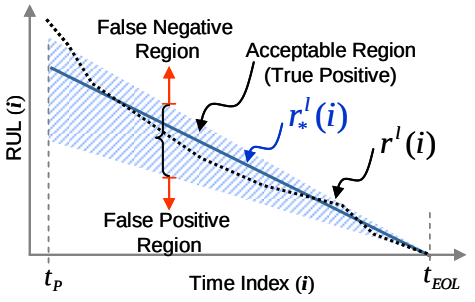
- Prognostics horizon
- *α-λ* performance
- Relative accuracy
- Cumulative relative accuracy
- Convergence











Source: A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, and M. Schwabacher (2008*). Metrics for evaluating performance of prognostic techniques. International Conference on Prognostics and Health Management*, PHM 2008. 6-9 Oct. 2008 Page(s): 1-17.

Prognostic Horizon (PH)

- **Prognostic Horizon** is defined as the difference between the time index i when the predictions first meet the specified performance criteria (based on data accumulated until time index i) and the time index for End-of-Life (EoL). The performance specification may be specified in terms of allowable error bound (α) around true EoL.

$$PH = t_{EoL} - t_{i_{\alpha\beta}}$$

$i_{\alpha\beta}$ is the first time index when predictions satisfy β-criterion for a given α

$$\min\left\{j \mid (j \in p) \wedge \left(\pi[r(j)]_{-\alpha}^{+\alpha}\right) \geq \beta\right\}$$

p is the set of all time indexes when predictions are made

l is the index for l^{th} unit under test (UUT)

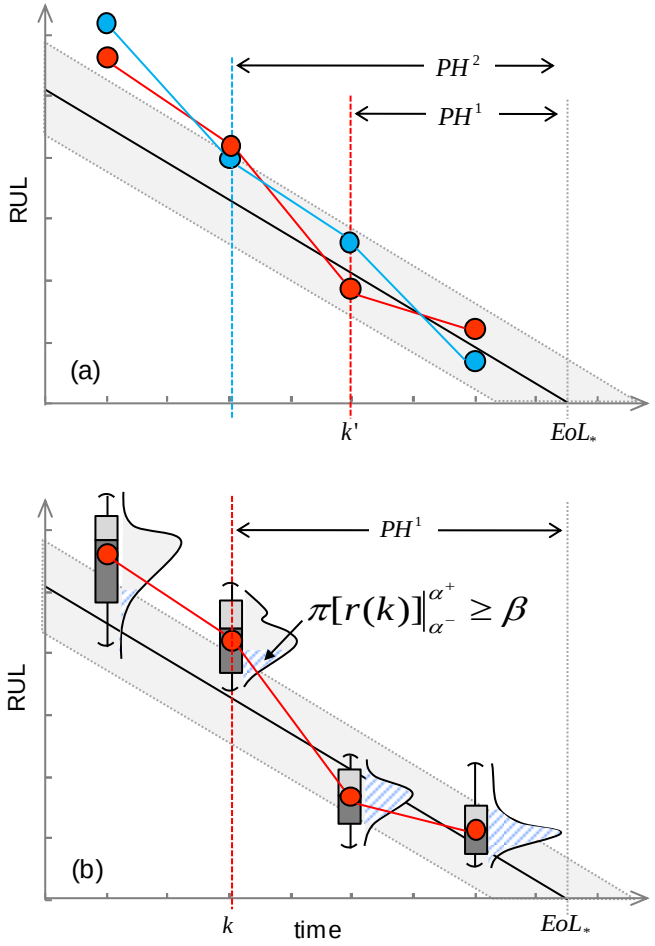
β is the minimum acceptable probability mass

$\pi[r(j)]_{\alpha^-}^{\alpha^+}$ is the probability mass of the prediction between α-bounds given by $\alpha^+ = r_* + \alpha \cdot t_{EoL}$ and $\alpha^- = r_* - \alpha \cdot t_{EoL}$

$r(j)$ is the predicted RUL distribution at time t_j

t_{EoL} is the predicted End-of-Life

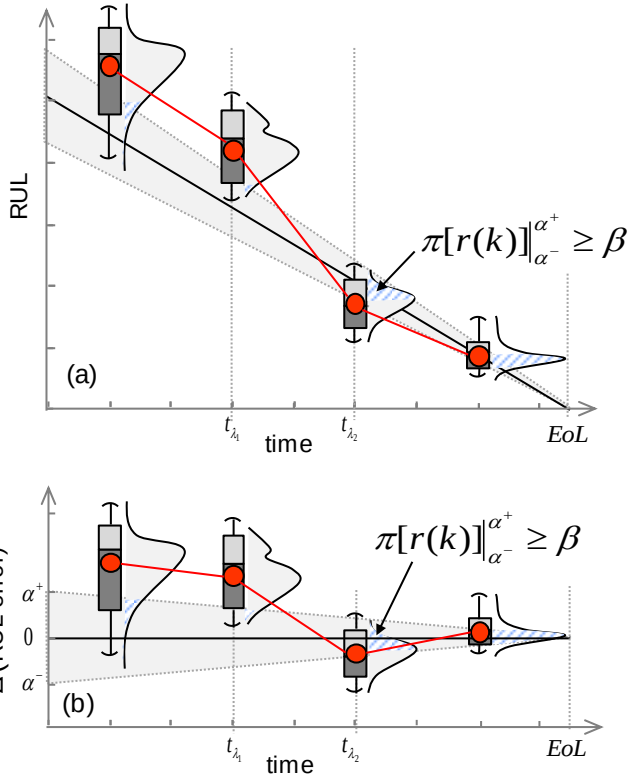
The range of PH is between (t_{EoL}-t_P) and max[0, t_{EoL}-t_{EoP}]



α-λ Accuracy

- ***α-λ Accuracy*** determines whether at given point in time (specified by λ) prediction accuracy is within desired accuracy levels (specified by α). Desired accuracy levels for any time t are expressed a percentage of true RUL at time t.

$$\alpha-\lambda\ Accuracy = \begin{cases} 1 & \text{if} \quad \pi\left[r(i_\lambda)\right]_{-\alpha}^{+\alpha} \geq \beta \\ 0 & \text{otherwise} \end{cases}$$



λ is the time window modifier such that $t_\lambda = t_P + \lambda(t_{EoL} - t_P)$

β is the minimum acceptable probability mass

$r(i_\lambda)$ is the predicted RUL at time t_λ

$\pi\left[r(i_\lambda)\right]_{\alpha^-}^{\alpha^+}$ is the probability mass of the prediction between α-bounds given by $\alpha^+ = r_*(i_\lambda) + \alpha \cdot r(i_\lambda)$ and $\alpha^- = r_*(i_\lambda) - \alpha \cdot r(i_\lambda)$

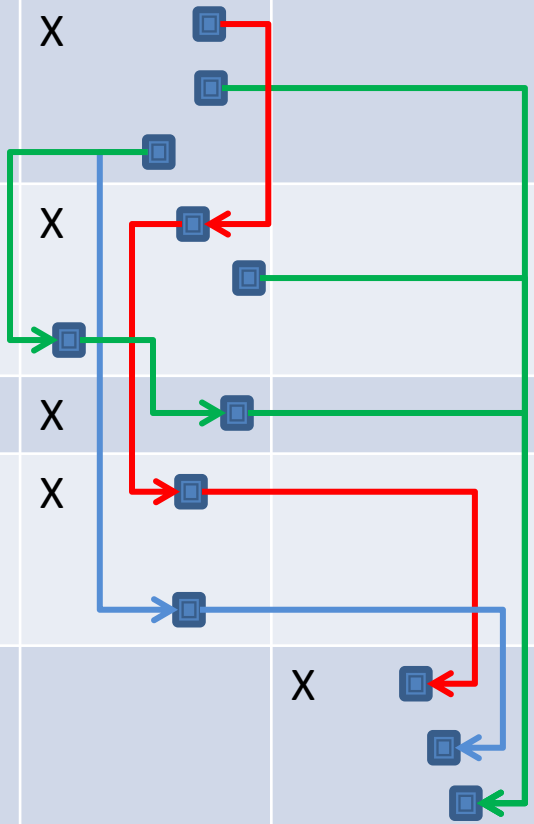
Electric UAV

Case Study

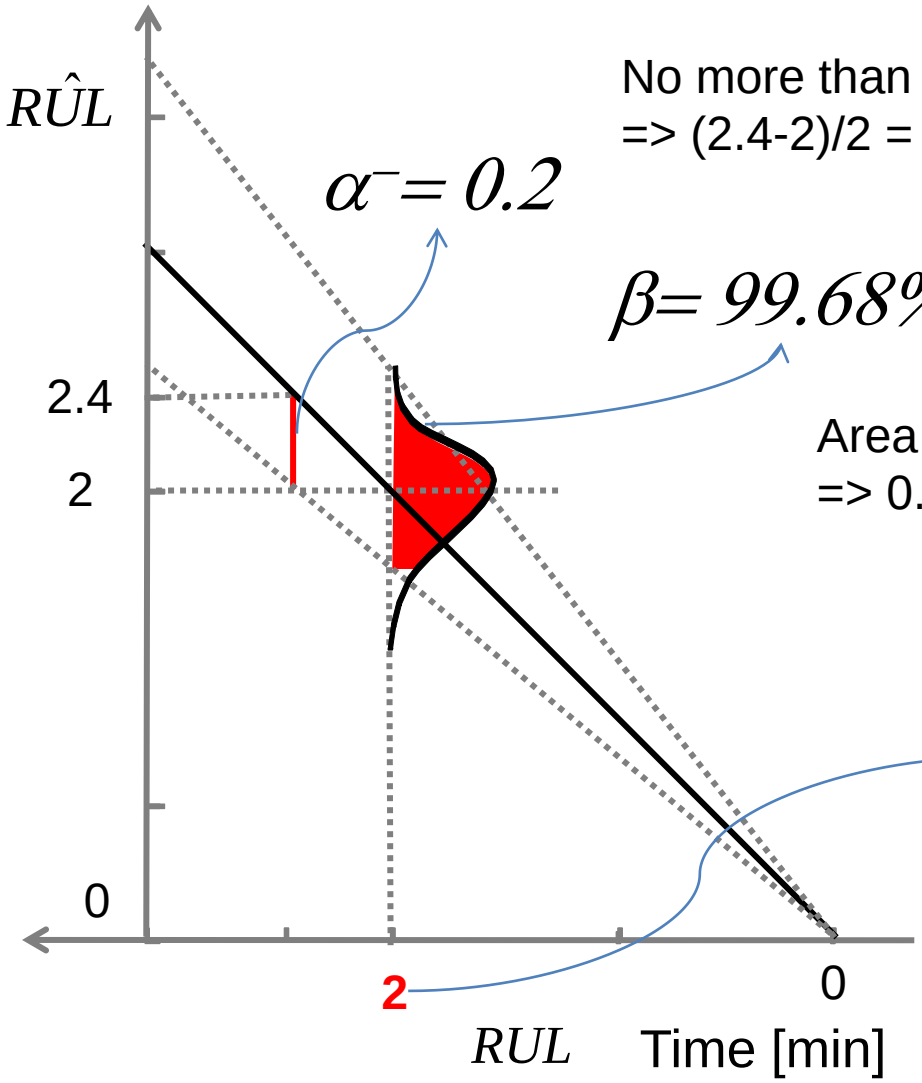
Requirement Dependencies

(for Prognostic Algorithms)

Requirement	Customer/ User	Vendor/ Alg. Developer
• Run mission safely for 20 minutes • Risk of loss < 4% • Cost Requirements	X	
• Ensure safe landing when battery runs low • Proportion of battery problems = 20% • Maximize battery cycle life	X	
• 60% battery issues covered by maint , rest 40% need ISHM	X	
• Determine time to remaining battery cutoff with at least 2 minutes lead time (needed for landing) • but no more than 0.4 minutes too early	X	
• Develop prognostics algorithm with λ = 2 minutes • α = 0.2 • β = 99.68%		X



Deriving Parameters for Metrics



No more than 0.4 minutes early
=> (2.4-2)/2 = α

$\alpha^{-} = 0.2$

$\beta = 99.68\%$

Area outside cone:
=> 0.04*0.2*0.4=1-β

$\lambda = (20-2)/20 = 0.9$

Key Parameters

- β
  - Maximum allowable Probability of Failure (PoF) of the system to bound risk
- λ
  - Lead time to specify the amount of advanced warning needed for appropriate actions
- α
  - Maximum tolerable probability of proactive maintenance to bound unnecessary maintenance

Current Challenges in Prognostics

Where do we go from here?

Some Open Questions

- Design for PHM
  - Take health management into consideration during the **design** stage of the system
    - Assess life cycle cost
- Cost-Benefit Models
  - Quantify the ROI
- Validation and Verification (V&V)
  - In order for a requirement statement to be valid (or at least realistic), you must be able to apply a rigorous V&V methodology to show that the requirement is met
  - However, in a perfect prognostic system, parts are always replaced before they fail
  - Though limited post-mortem analyses may be made, it is infeasible to determine the actual SOH of all pulled components
  - Even if you did know the actual SOH of all pulled components, its difficult to know the RUL pdf of the pulled component
- Certification
  - Towards Maintenance credits
- Uncertainty Management
  - Quantification, representation, propagation, and management
    - We've come a long way, but there's still more to be achieved!

Questions?

Thank you!