When Will it Break?

BEST lab 30 year reunion

in honor of
Alice Agogino
UC Berkeley

Kai Goebel, Ph.D.
kai.goebel@nasa.gov
http://prognostics.nasa.gov
8. Aug, 2015

Stuff breaks

Image credit: http://www.improvisedlife.com/2014/03/19/dept-impermanence/#lightbox/2/

What can we do about it?

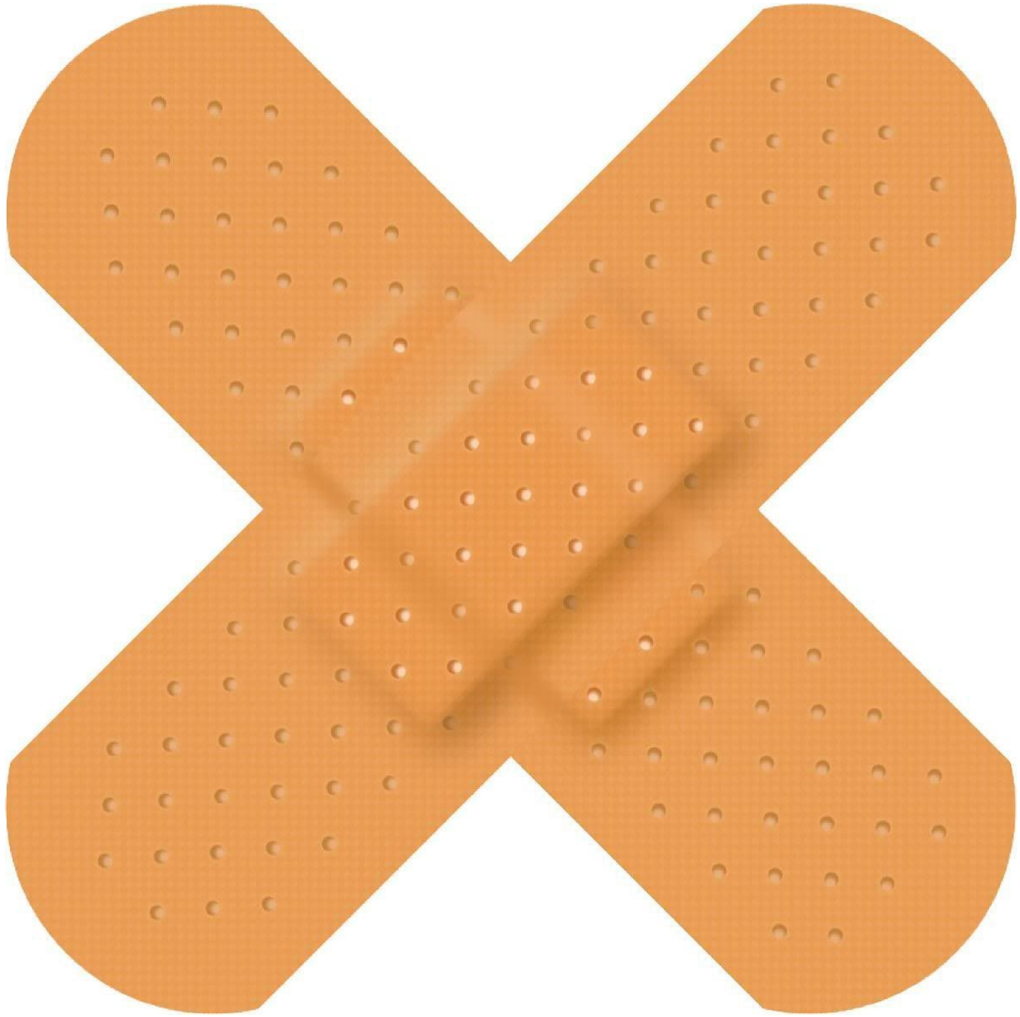
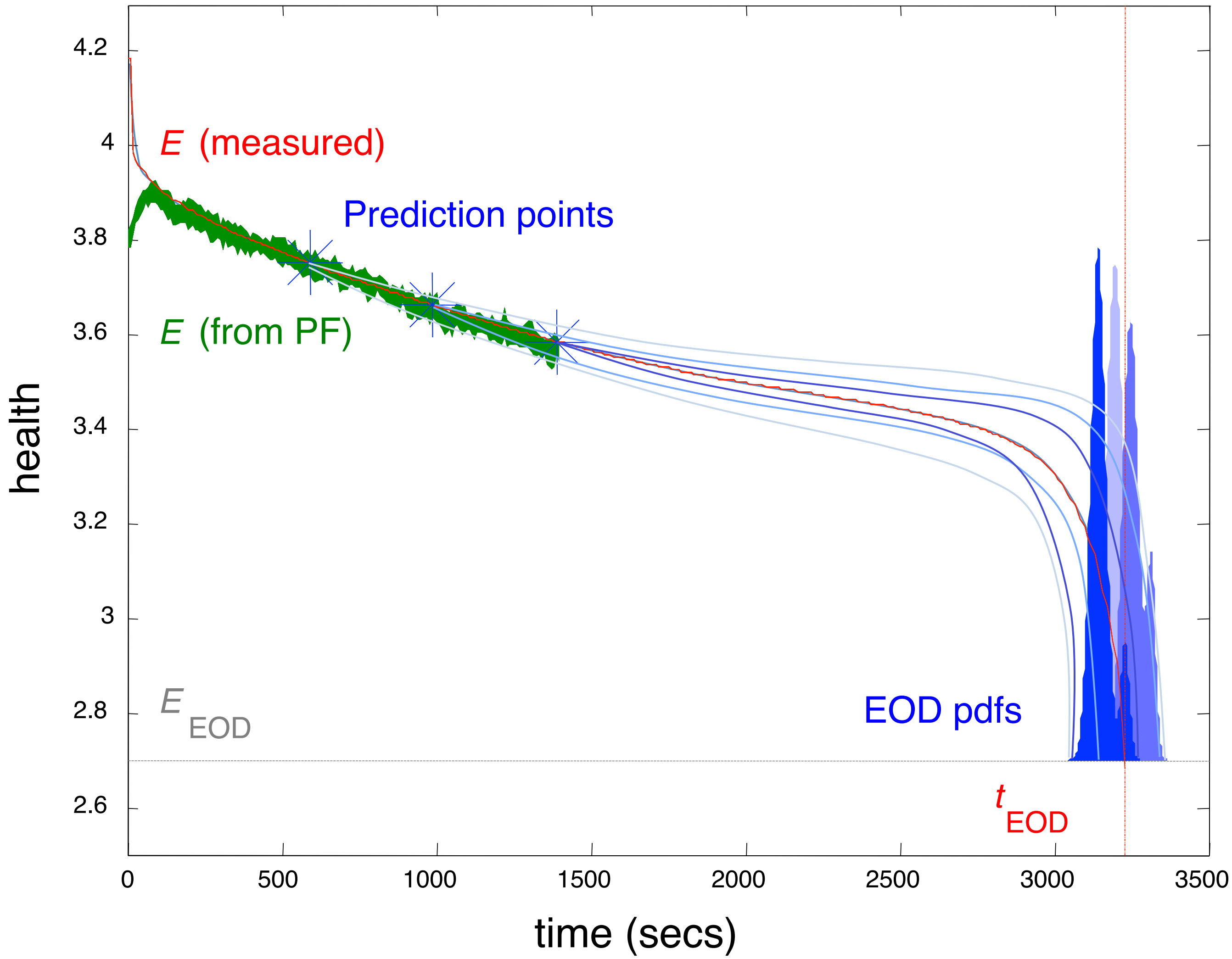


Image credit: http://www.ebay.com/itm/JDM-Band-Aid-Decal-Vinyl-Bandage-cover-dents-dings-funny-sticker-decal-DRIFT-CAR-/391149550941

Prediction

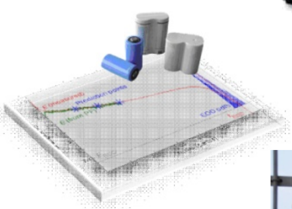
Image credit: http://www.pinetarpress.com/wp-content/uploads/2013/03/crystalball1.jpg

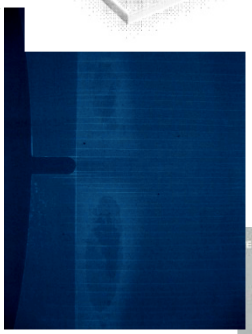
Prognostics

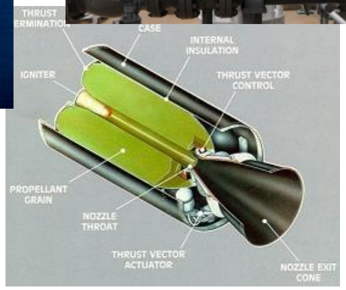


Application Examples

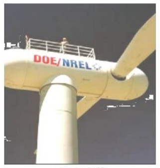
- Electro-Mechanical Actuators
- Electrochemical Storage
- Electronics
- Valves, Pumps
- Composite Materials
- Solid Rocket Motor Casing
- Rover
- UAV
- Wind Turbines
- Biomass







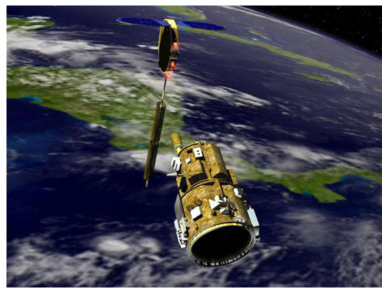

NWTC, Golden, Colorado


- last slide

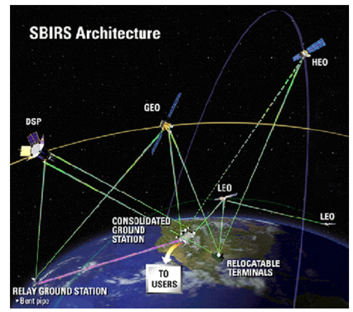
Case Studies

OCO

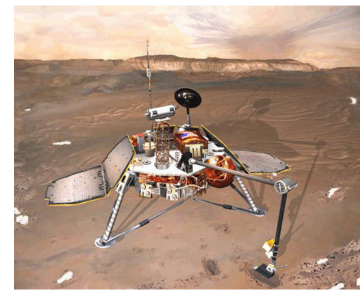
DART



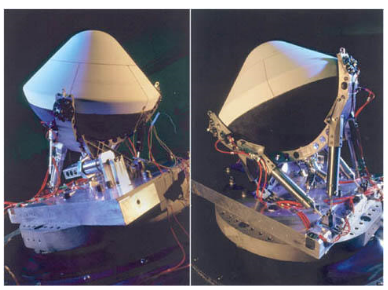
SBIRS



MPL



Deep Space 2



Mars Global Surveyor

Apollo 13

Space Shuttle

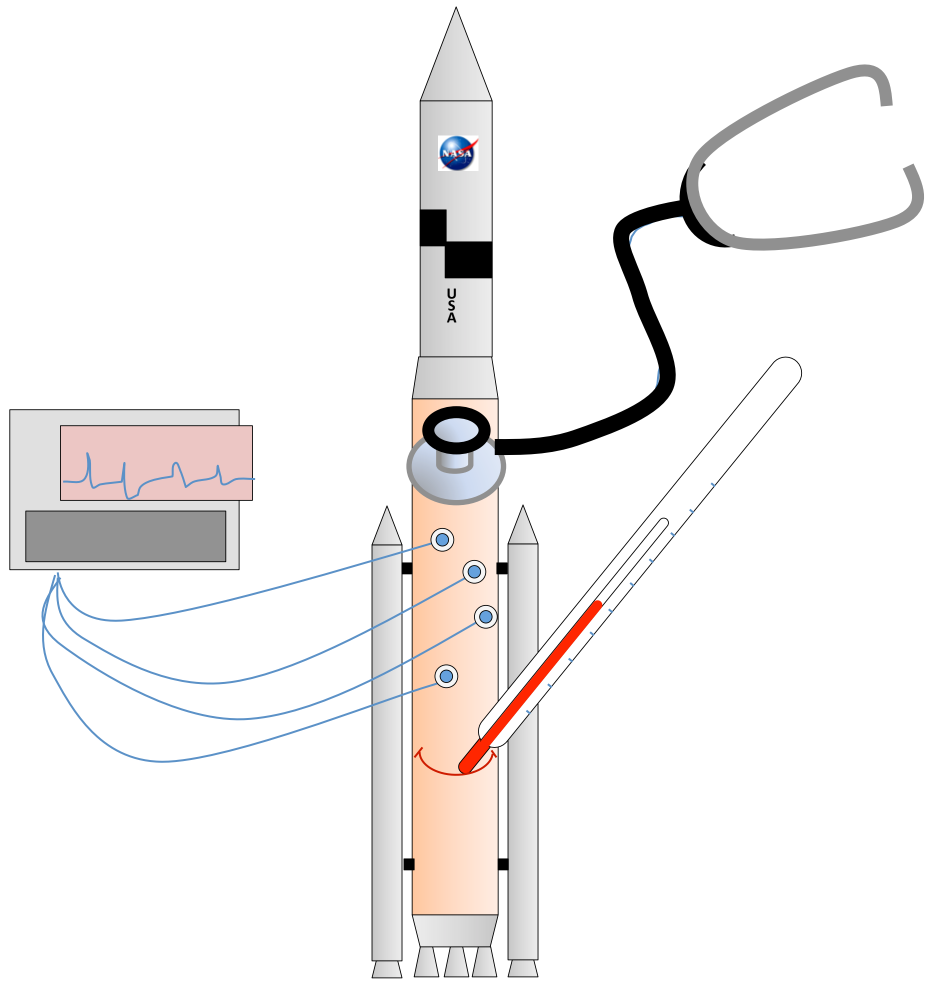
Source: http://www.popsci.com/military-aviation-amp-space/gallery/2009-03/top-10-nasa-probe-failures
http://www.gerhards.net/albums/spaceshuttle/SpaceShuttle.jpg

Case Studies

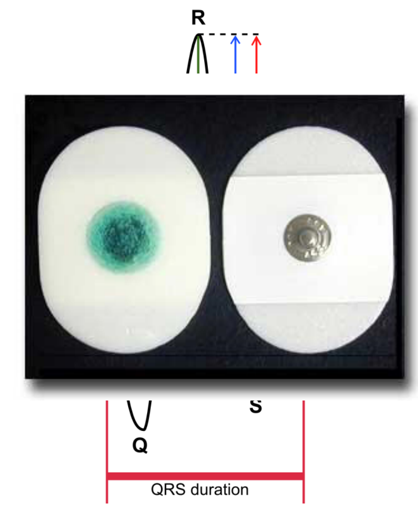
Aloha Flight 243

Delta Flight 1288

Health Determination



ECG



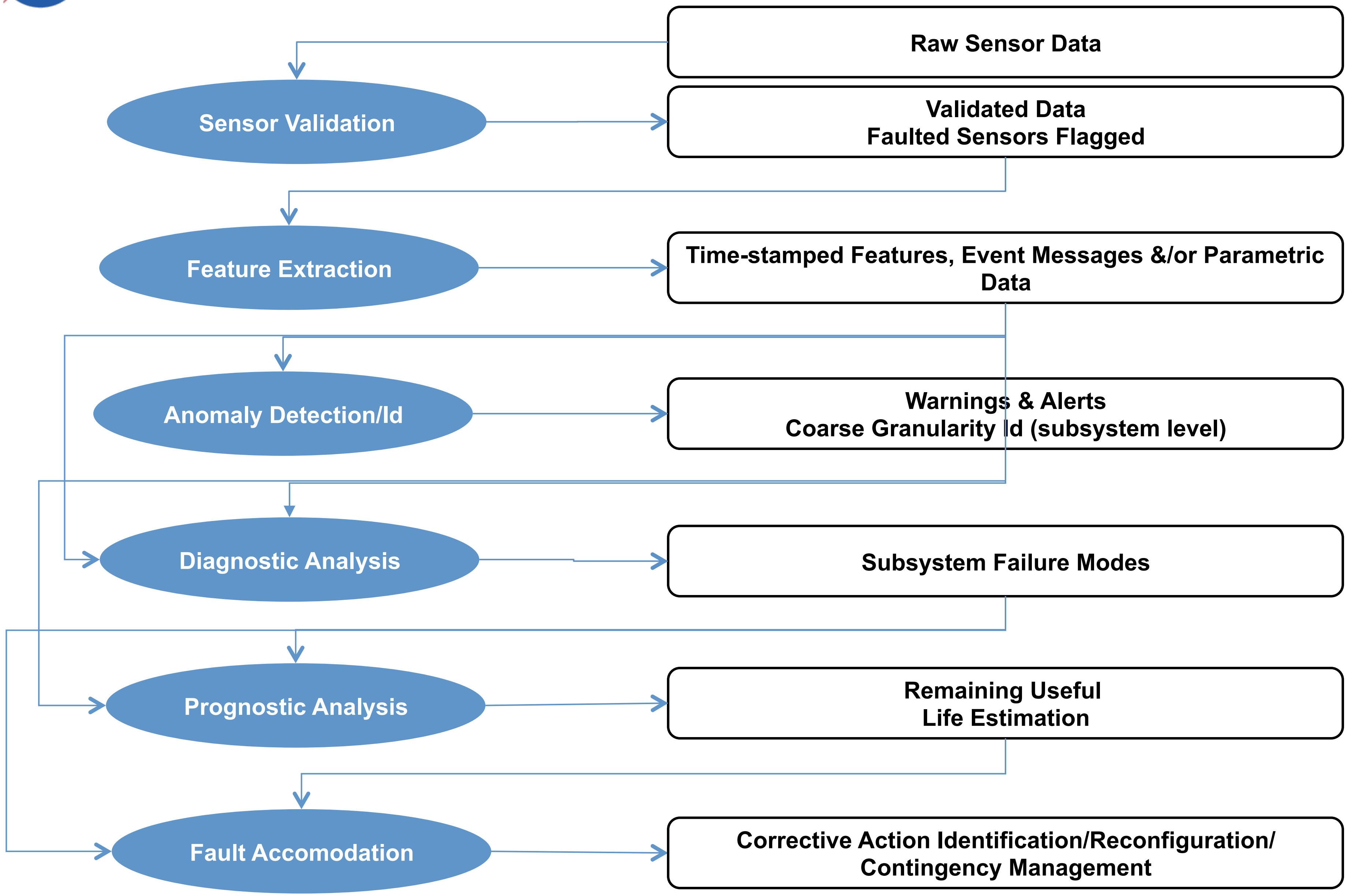
- Are probes working well?
- How does one interpret the ECG signal?
- Do things appear to be within normal bounds?
- If not, what is the diagnosis?
- Given the diagnosis, what is the prognosis?
- Suggest therapy

Sources: wikimedia.org, www.kappamedical.com

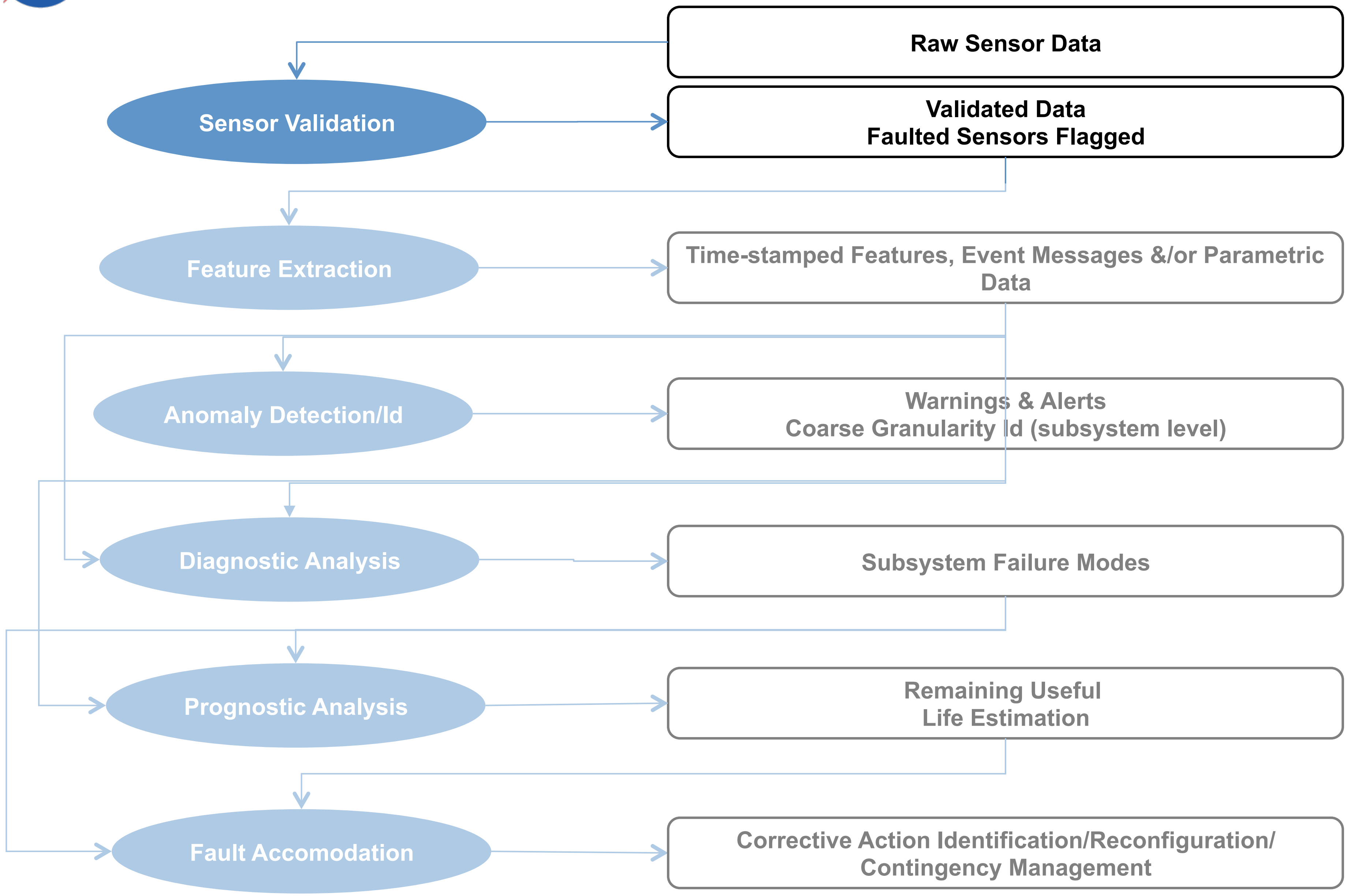
ISHM

- Sensor Validation
- Feature Extraction
- Abnormal Condition Detection
- Diagnostics
- Prognostics
- Mitigation

Systems Health Management

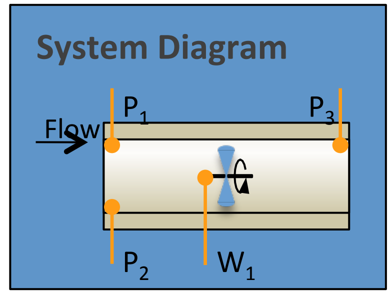


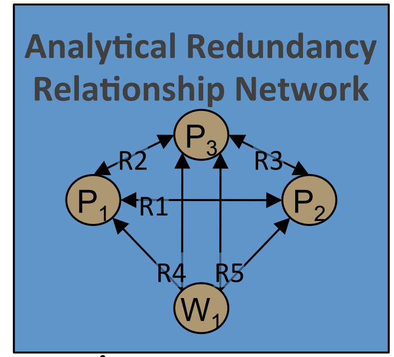
Systems Health Management



Sensor Validation

- **Acquire Data** from sensors to be validated and from other sources to determine system operating mode





- **Estimate** output of each sensor using known/derived relationships with other sensors

$$\text{R1}: \hat{P}_{1,1} = P_2$$
$$\text{R2}: \hat{P}_{1,2} = C_{2,2}P_3 + C_{2,1}$$
$$\text{R3}: \hat{P}_{2,3} = C_{3,2}P_3 + C_{3,1}$$
$$\text{R4}: \hat{W}_{1,4} = C_{4,2}(P_1 - P_3)^{1/2} + C_{4,1}$$
$$\text{R5}: \hat{W}_{1,5} = C_{5,2}(P_2 - P_3)^{1/2} + C_{5,1}$$

- **Detect** and flag breakdown of any relationships by comparing residuals (i.e., difference between measurement & estimate) to pre-defined thresholds

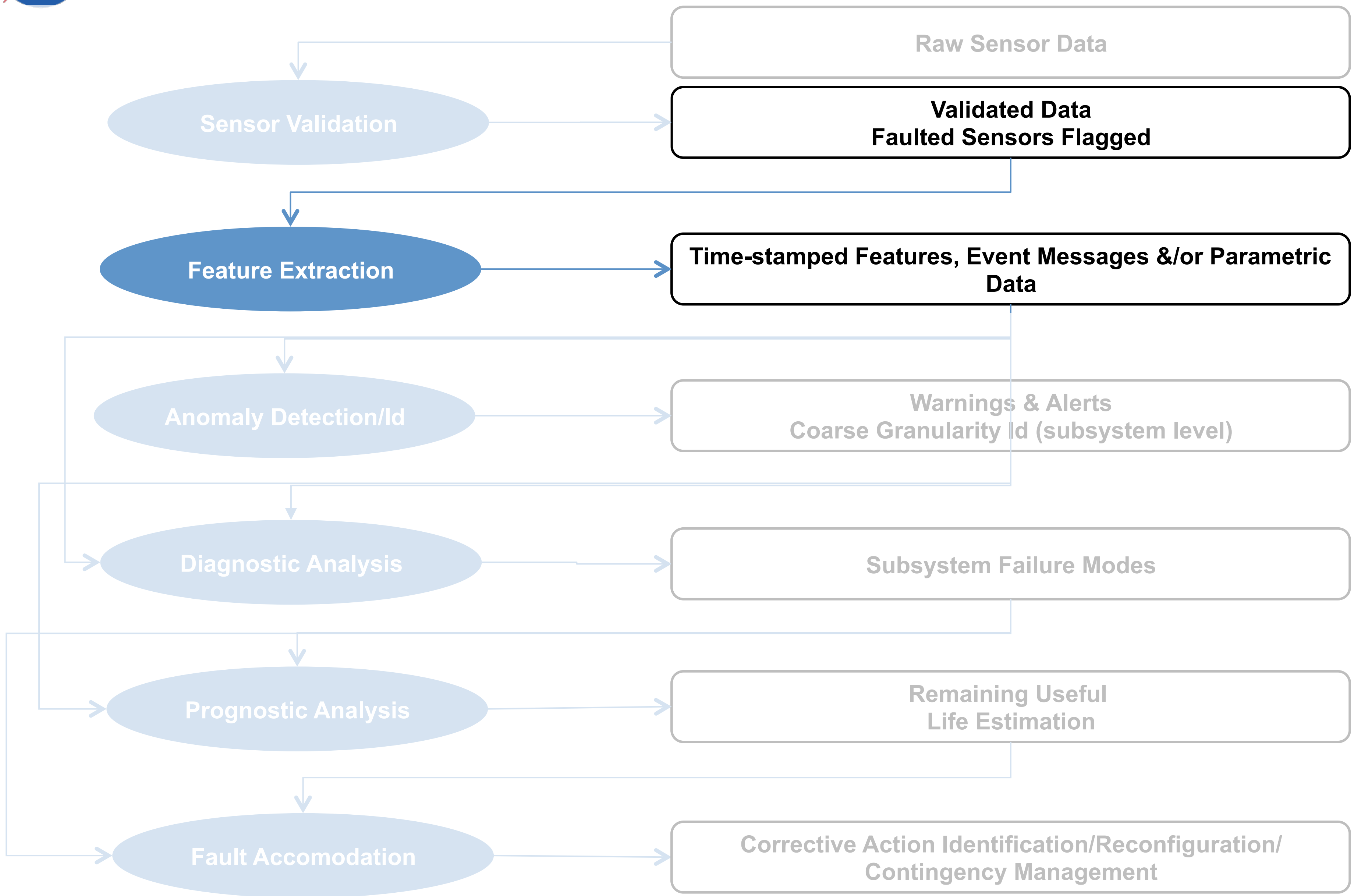
$$\text{if } \left|\hat{P}_{1,1} - P_1\right| \le T_1, \text{then R1} = \text{qualified, else R1} = \text{failed}$$
$$\vdots$$
$$\text{if } \left|\hat{W}_{1,5} - W_1\right| \le T_5, \text{then R5} = \text{qualified, else R5} = \text{failed}$$

- **Decide** if sensor has failed based on number and frequency of failed relationships
- **Disqualify** sensor and notify system/ user

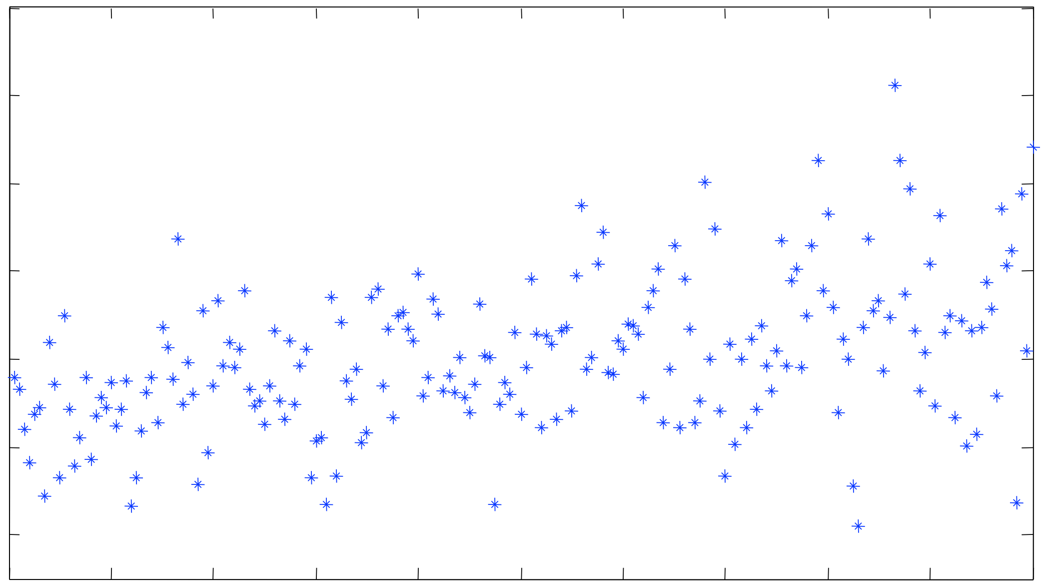
No. Active ARRs for a Signal	No. Failed ARRs Required to Disqualify the Signal
3	3
4	4
5	4

Courtesy K. Melcher, NASA Glenn

Systems Health Management



Feature Extraction



- Questions:
  - How do we extract information (features) from raw sensor data?
  - How do we extract useful features from raw sensor data?
  - How do we select the best features from raw data in order to detect and identify fault (failure) modes?
  - How do we select the best features from raw data in order to predict remaining life?

Feature Extraction

- Good features have the following attributes:
  1. Explainable in physical terms
  2. High correlation with fault/fault progression
  3. Mathematically definable
  4. Characterized by large interclass mean distance and small interclass variance
  5. Uncorrelated with other features
  6. Insensitive to extraneous variables
  7. Computationally inexpensive to measure

Where is the Information?

- Relating fault (failure) mechanisms to the fundamental physics of complex dynamic systems
- Fault (failure) modes induce changes in:
  - The energy (power) of the system
  - Its entropy
  - Power spectrum
  - Signal magnitude
  - Chaotic behavior
  - Other

How do we get the Information?

- How are system functional changes (symptoms) monitored or measured in terms of measurable system states (outputs)?
- Measurable quantities:
  - Vibration
  - Temperature
  - Pressure
  - Etc.
- Extracting information
  - Time domain
  - Frequency domain
  - Chaotic domain

Features of Features

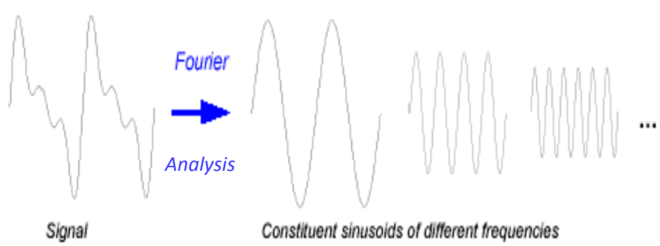
- Derived Features (or Features of Features)
  - Continue further processing of primary features in order to arrive at unique, uncorrelated (distinguishable) fault (failure) signatures.
- Examples
  - Statistical moments (Skewness, Kurtosis)
  - Linear/non-linear combinations of features
- The Tools
  - Genetic Programming
  - Genetic Algorithm
  - Other optimization tools

Transforms

Fourier transform

- Sines and cosines as basis functions

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \boxed{e^{-j\omega t}} dt$$

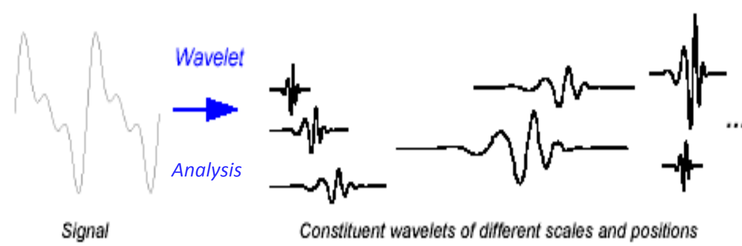


- In transforming to the frequency domain, time information is completely lost

Wavelet transform

- Infinite possible basis functions

$$F(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \boxed{\psi\left(\frac{t-b}{a}\right)} dt$$

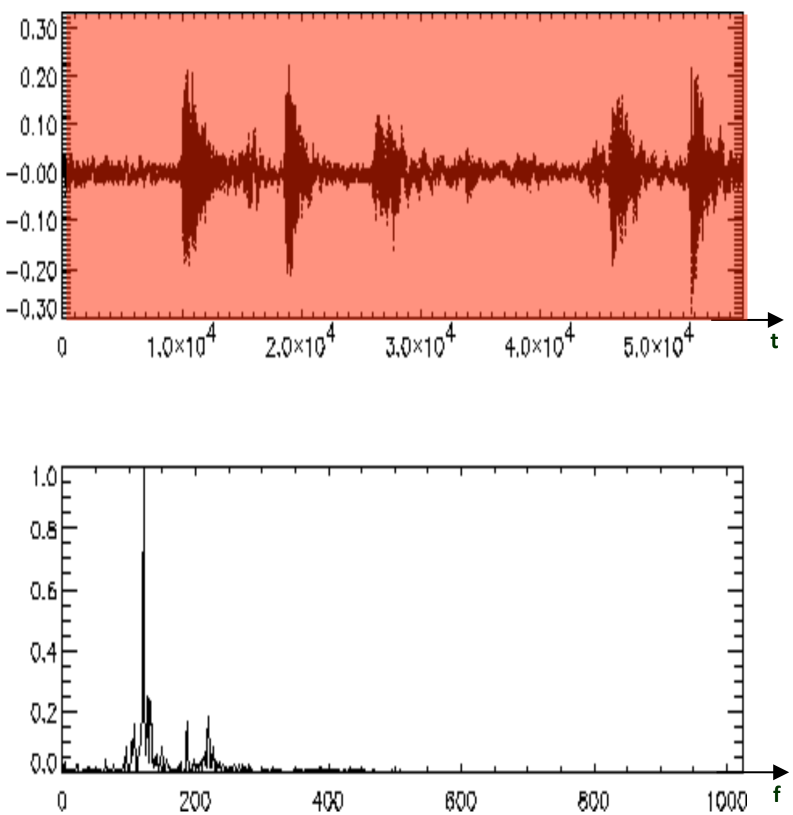


- A systematic windowing technique with variable-sized windows (dictated by *a*)

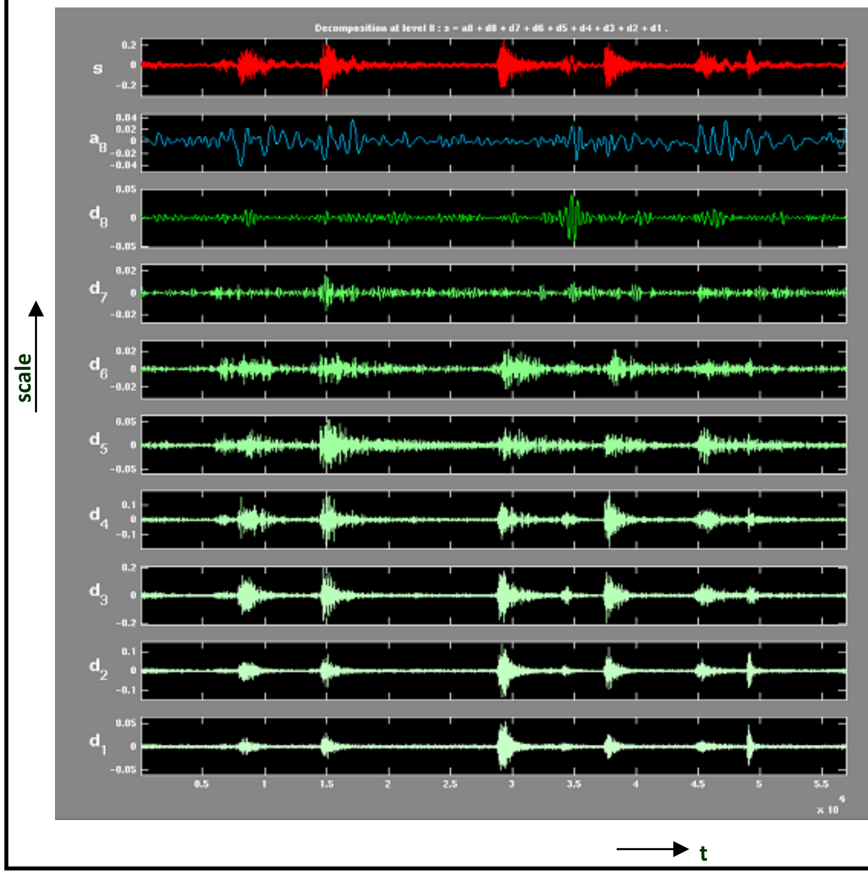
What are wavelets?

- Signal analysis technique complementary to traditional Fourier analysis
- Represent signal as linear combination of scaled and shifted versions of some generic function called the 'wavelet function of the mother wavelet'
    - Fractal-like
- Retains frequency information on a time-specific basis using varying resolutions or scales
    - Addresses the time-frequency tradeoff
    - Large scale amplifies gross signal features
    - Small scale amplifies finer signal features
    - Perception-like
- Best suited for detection of spikes, singularities and transients

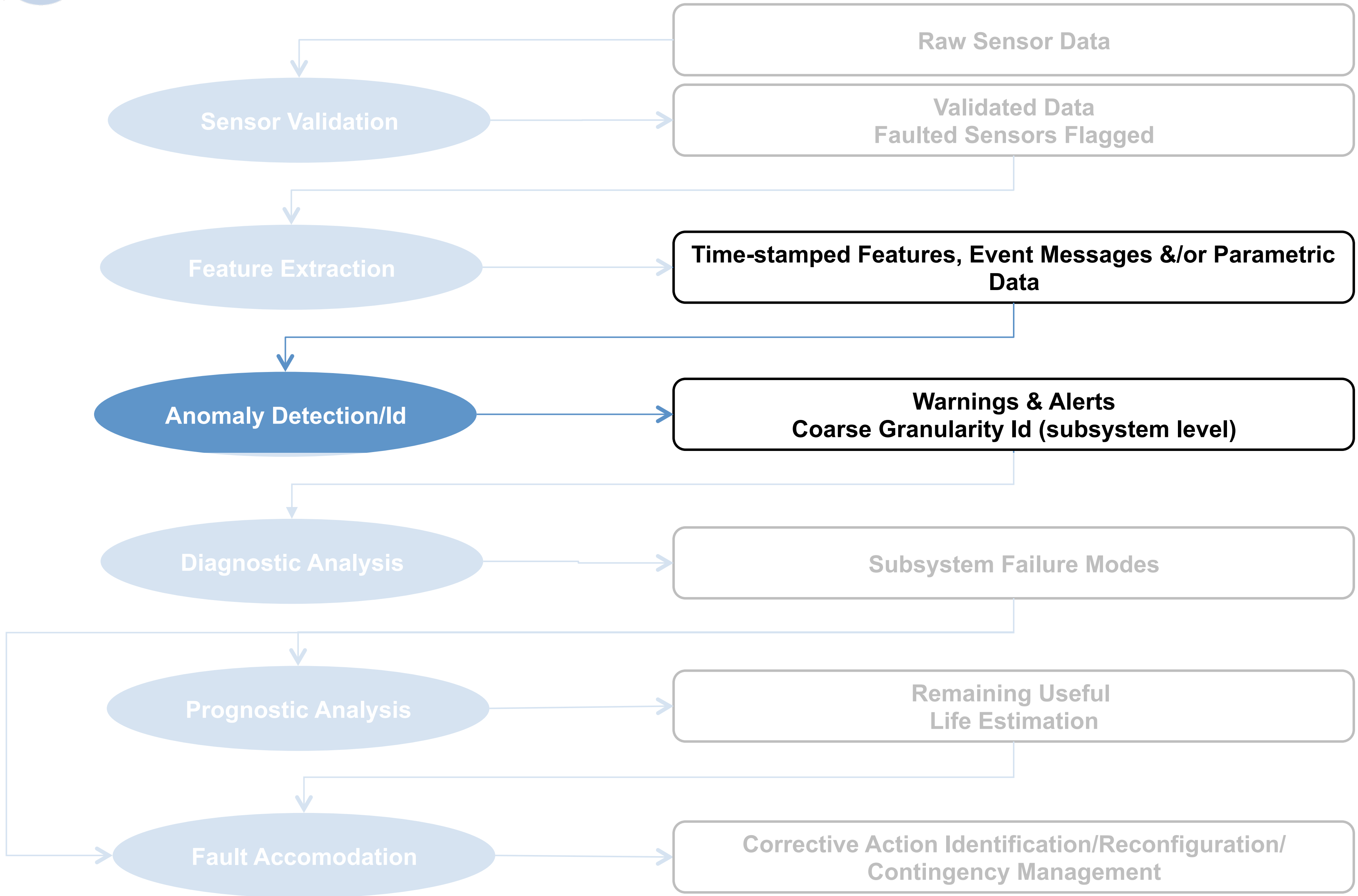
Fourier



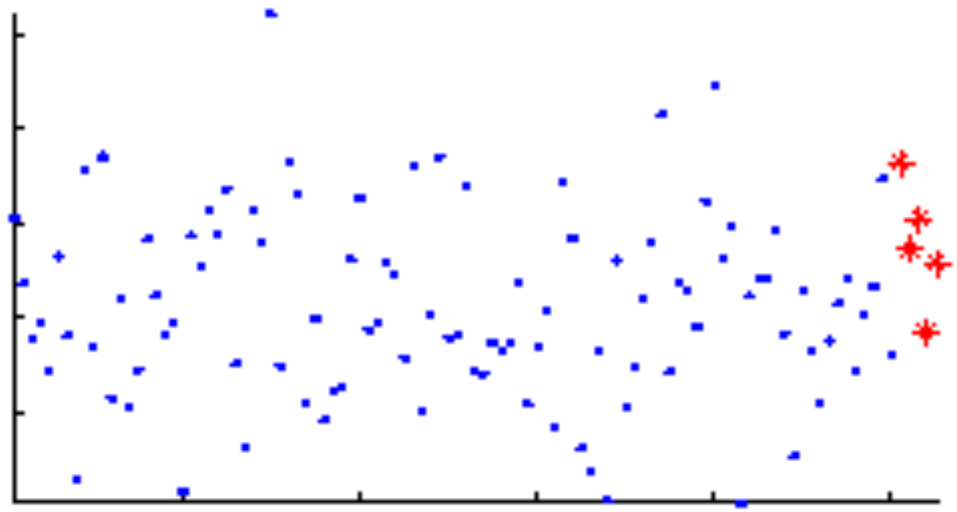
Wavelets



Systems Health Management



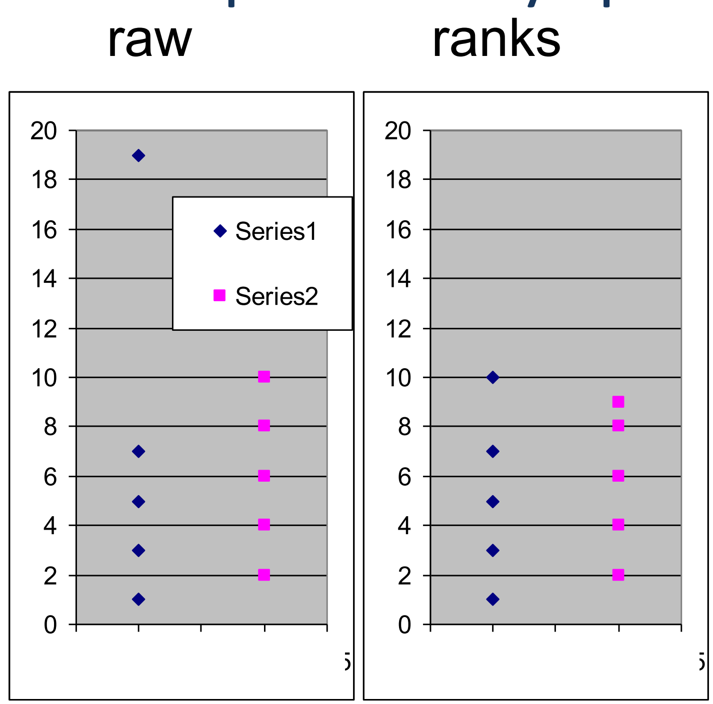
Anomaly Detection



- Are the last five points indicative of abnormal condition?

Rank Permutation Test

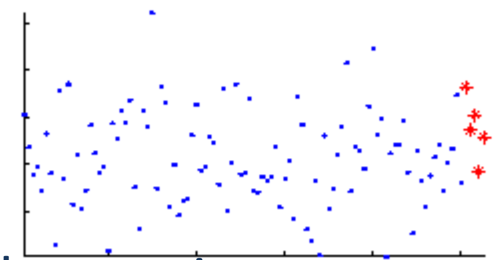
– Transform features from "raw feature space" to "rank permutation probability space"



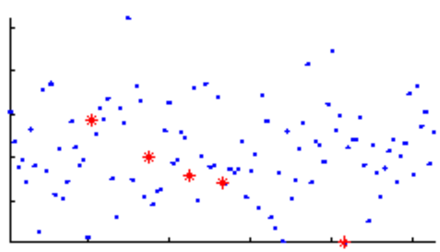
– Perform hypothesis test in rank space

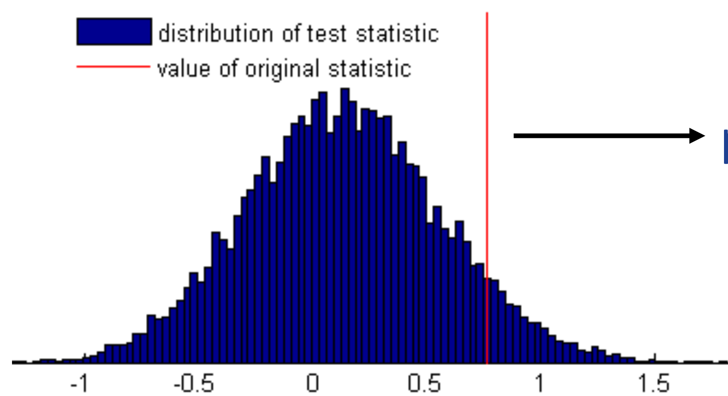
Courtesy: Neil Eklund

Permutation Test



1. Determine a testable null hypothesis
2. Choose a test statistic
   - here: sum of ranks
3. Compute the test statistic for the original observations
4. Permute the observations, and recalculate the test statistic; repeat
5. Accept or reject null hypothesis using permutation distribution





p=0.072

Courtesy: Neil Eklund

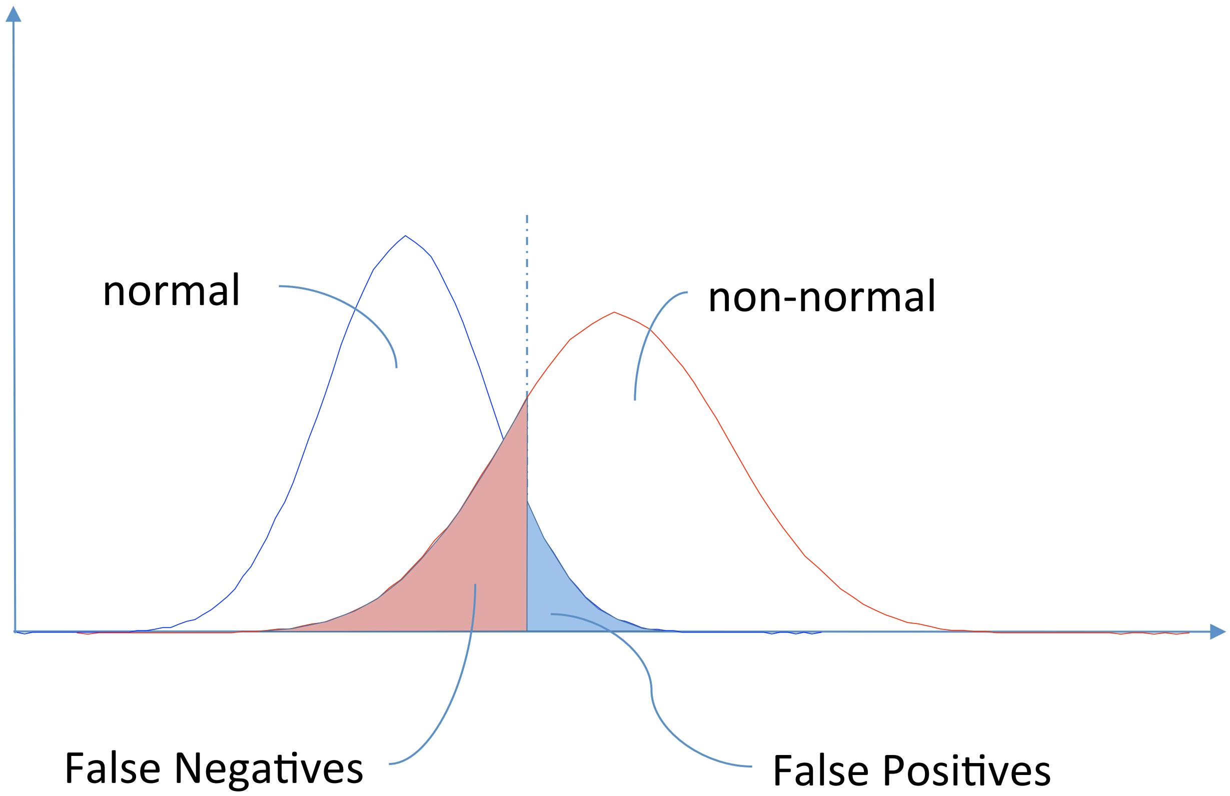
Rank Permutation Test

Advantages

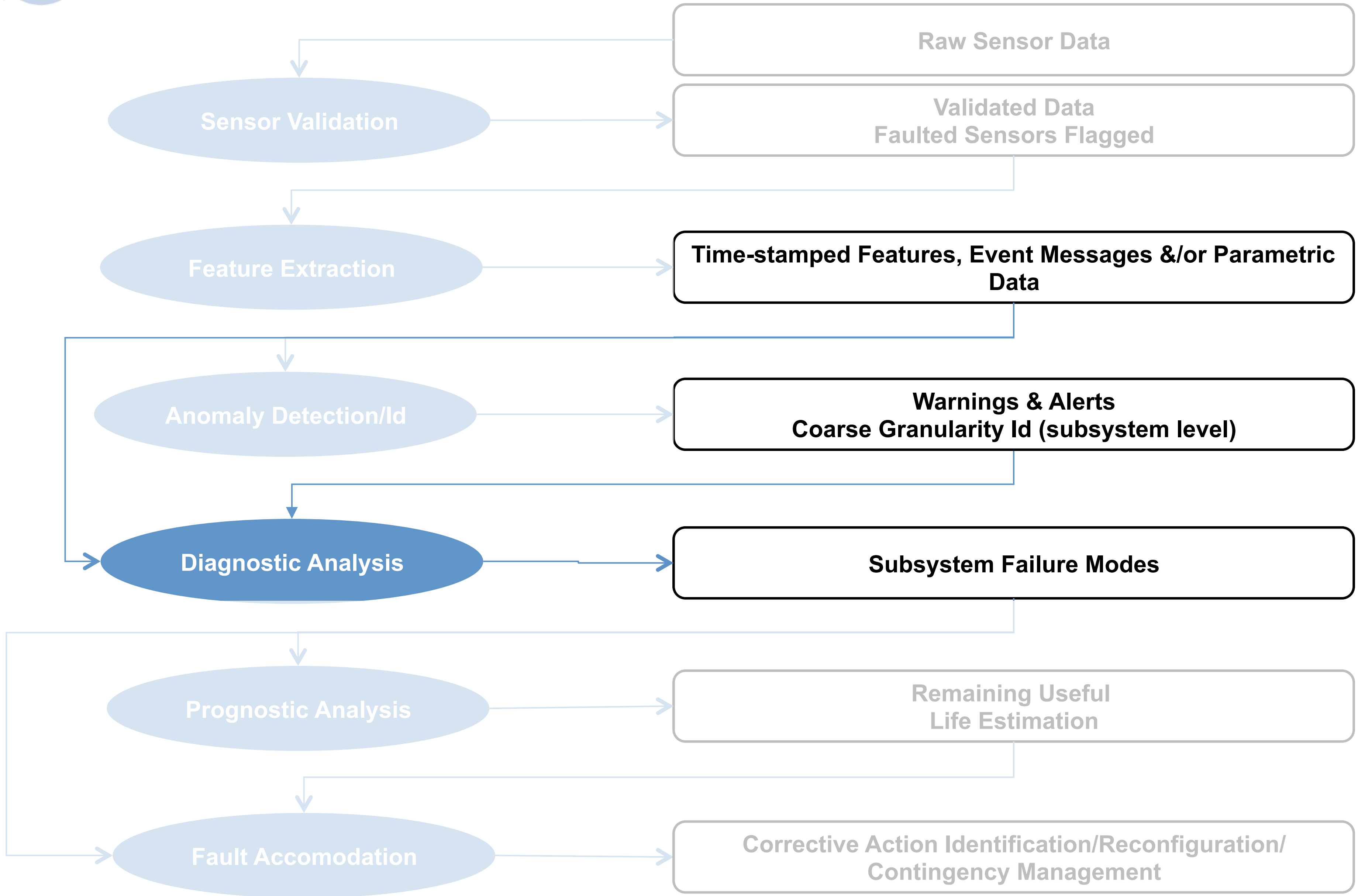
- Boosts classification rate by making events that are statistically improbable more pronounced
- Diminishes the effect of noise and outliers
- Permits pre-calculation of permutation distribution
  - Important for real time applications with limited computing power
- Computation becomes mostly a vector sorting

Courtesy: Neil Eklund

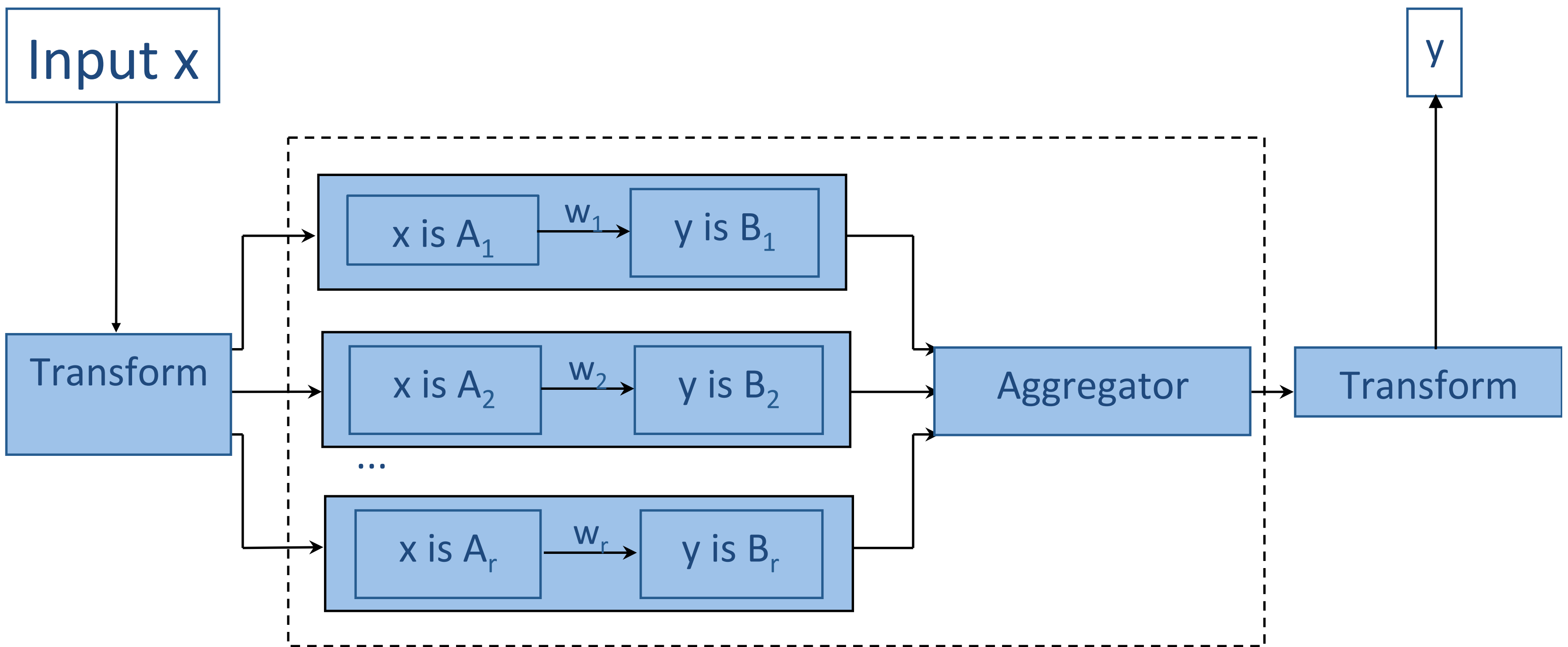
Performance Measures



Systems Health Management



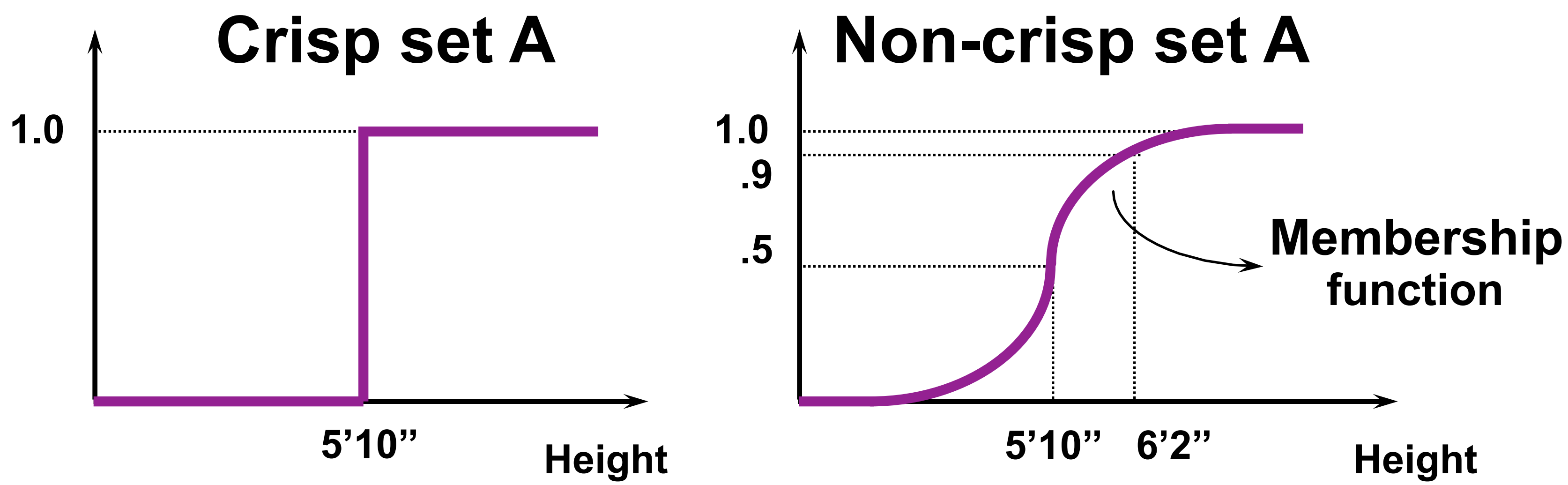
Inference System



Multi-valued Sets

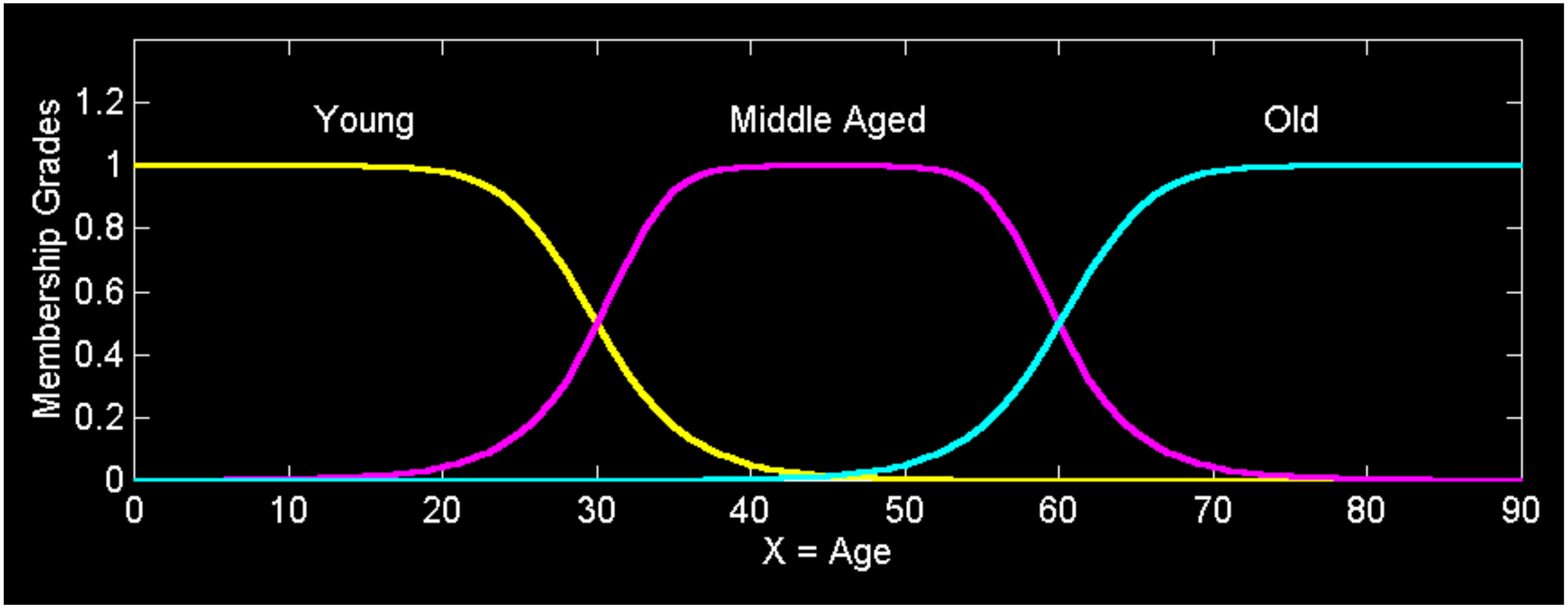
- Binary Logic vs. Multi-valued Logic:
- Sets with crisp and non-crisp boundaries, respectively

A = Set of tall people



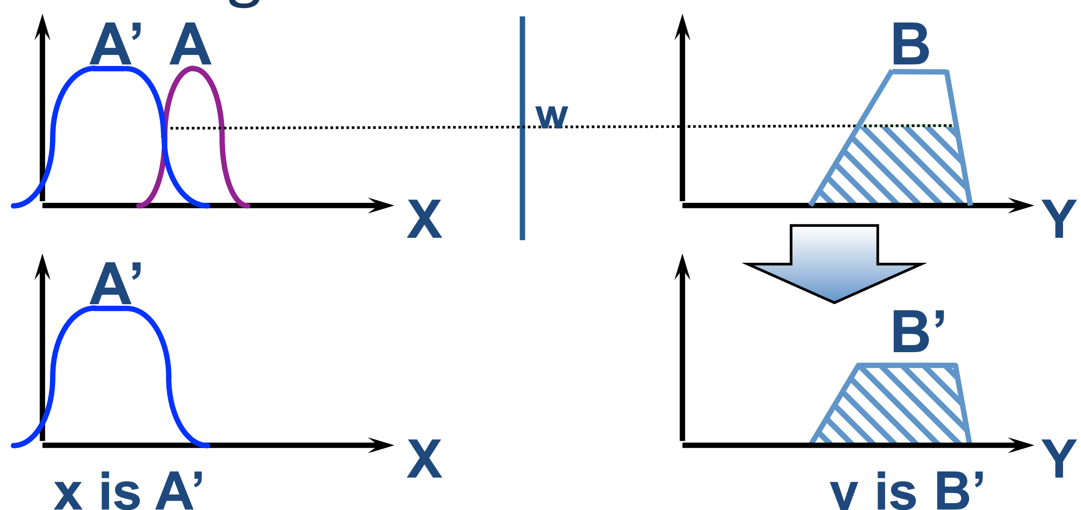
Capturing Uncertainty

- Partitions formed by the linguistic values "young", "middle aged", and "old":



Single Rule, Single Antecedent

- Graphical Representation:
  - find degree of match w between $\mu_A(x)$ and $\mu_{A'}(x)$
  - intuitively: degree of belief for antecedent which gets propagated; result should be not greater than w



Reasoning

- Single rule with multiple antecedents

  Facts: x is A’ and y is B’

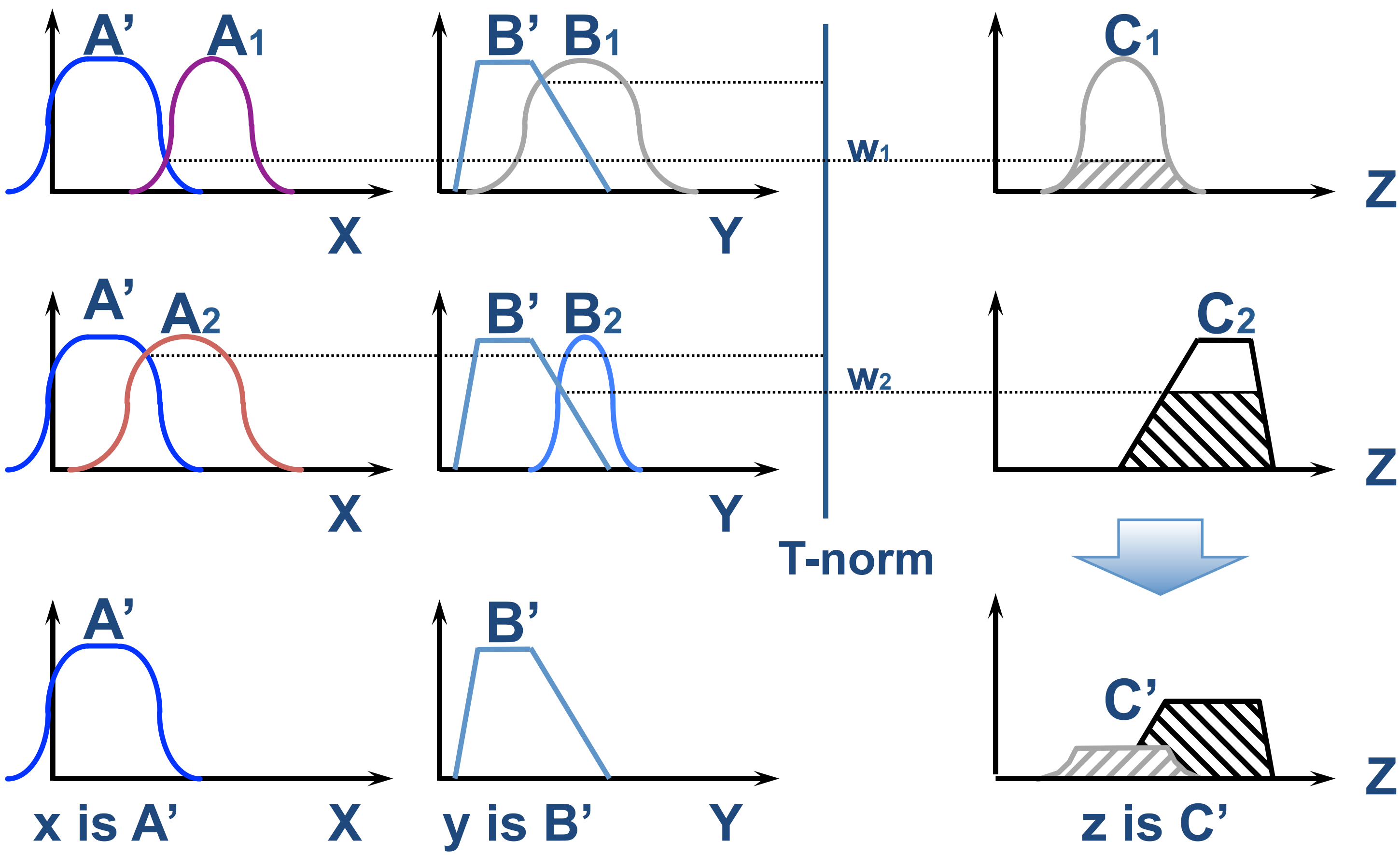
  Rule: if x is A and y is B then z is C

  Conclusion: z is C’

$$
\begin{aligned}
\mu_{C'}(z) &= \vee_{x,y}\left[\mu_{A'}(x) \wedge \mu_{B'}(y)\right] \wedge \left[\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)\right] \\
&= \vee_{x,y}\left[\mu_{A'}(x) \wedge \mu_{B'}(y) \wedge \mu_A(x) \wedge \mu_B(y)\right] \wedge \mu_C(z) \\
&= \underbrace{\left\{\vee_x\left[\mu_{A'}(x) \wedge \mu_A(x)\right]\right\}}_{w_1} \wedge \underbrace{\left\{\vee_x\left[\mu_{B'}(y) \wedge \mu_B(y)\right]\right\}}_{w_2} \wedge \mu_C(z) \\
&= \underbrace{\left(w_1 \wedge w_2\right)}_{\textit{firing strength}} \wedge \mu_C(z)
\end{aligned}
$$

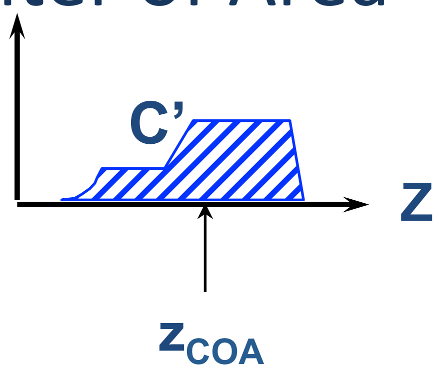
Reasoning

- Graphical representation:



Transform: multi-valued to crisp

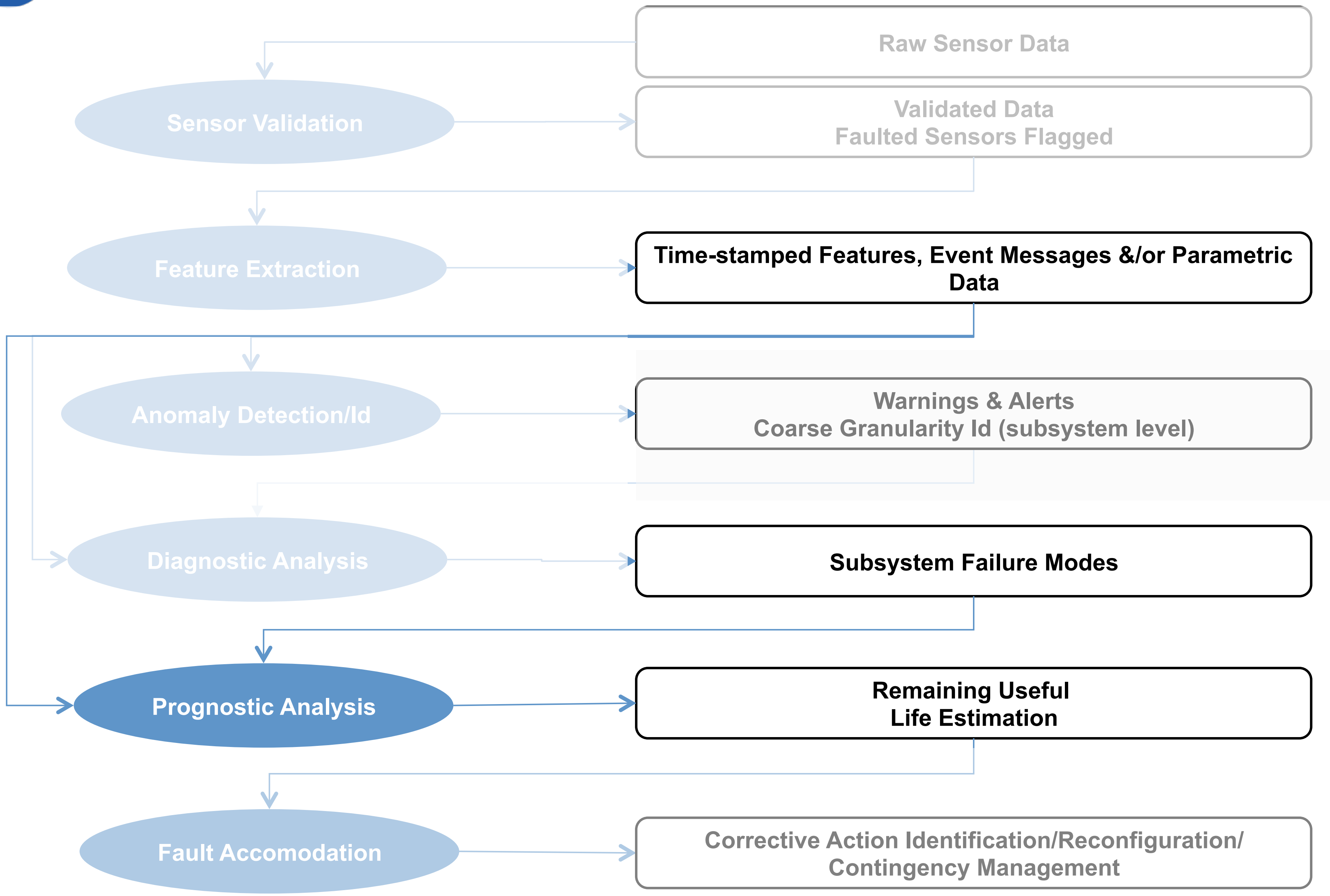
- Center of Area



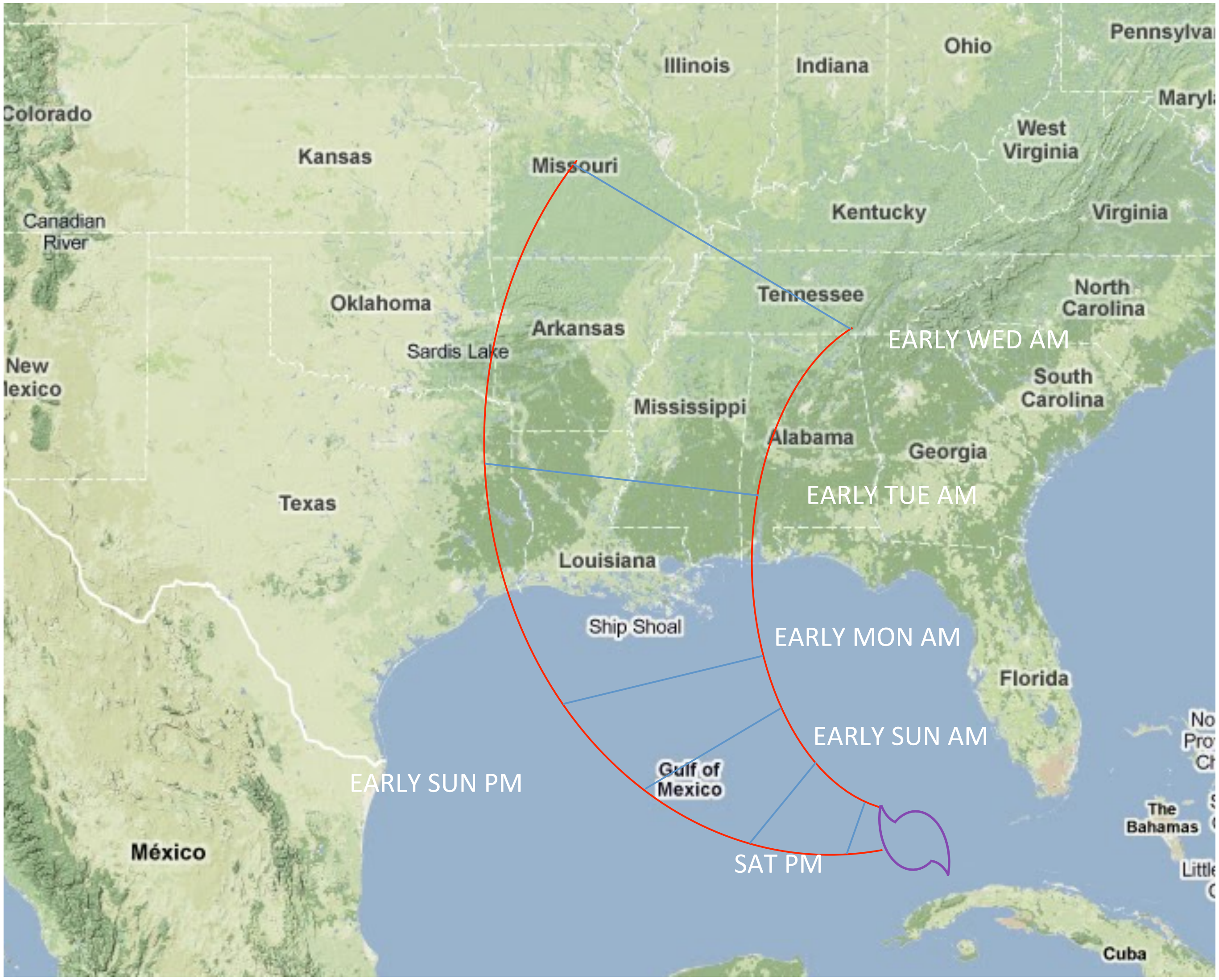
+ intuitive
+ smooth
- comp. burden

$$z_{COA} = \frac{\int_z \mu_A(z) z dz}{\int_z \mu_A(z) dz}$$

Systems Health Management

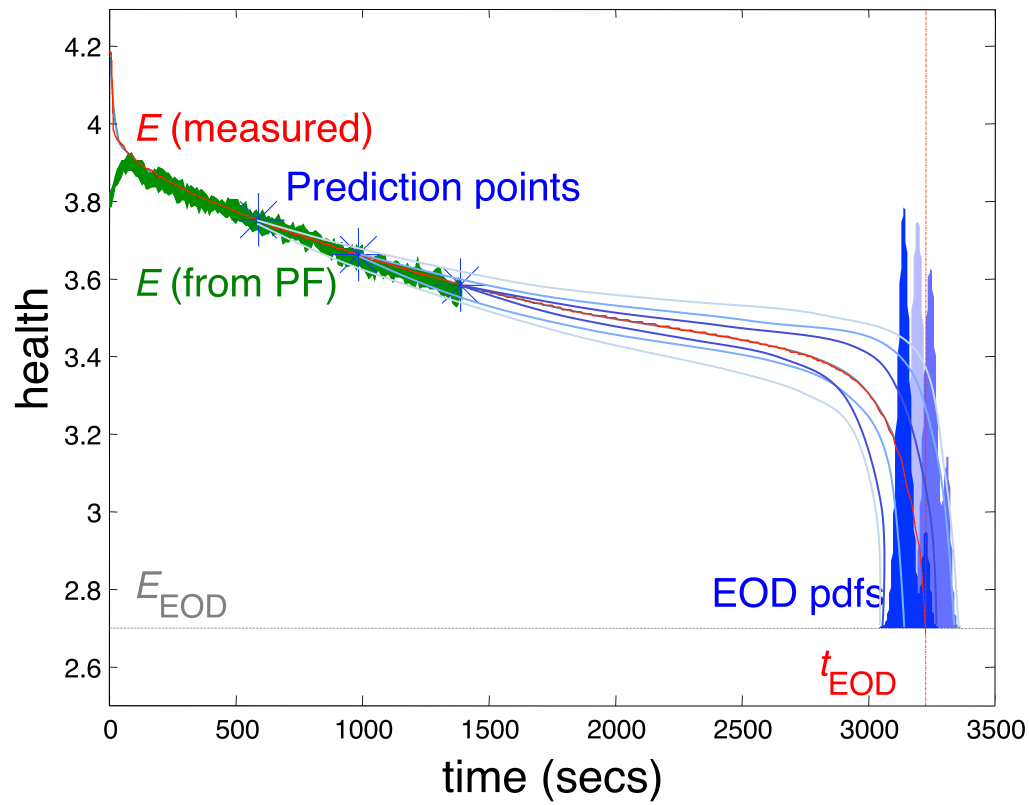


Prediction



Sources: weather.com; maps.google.com

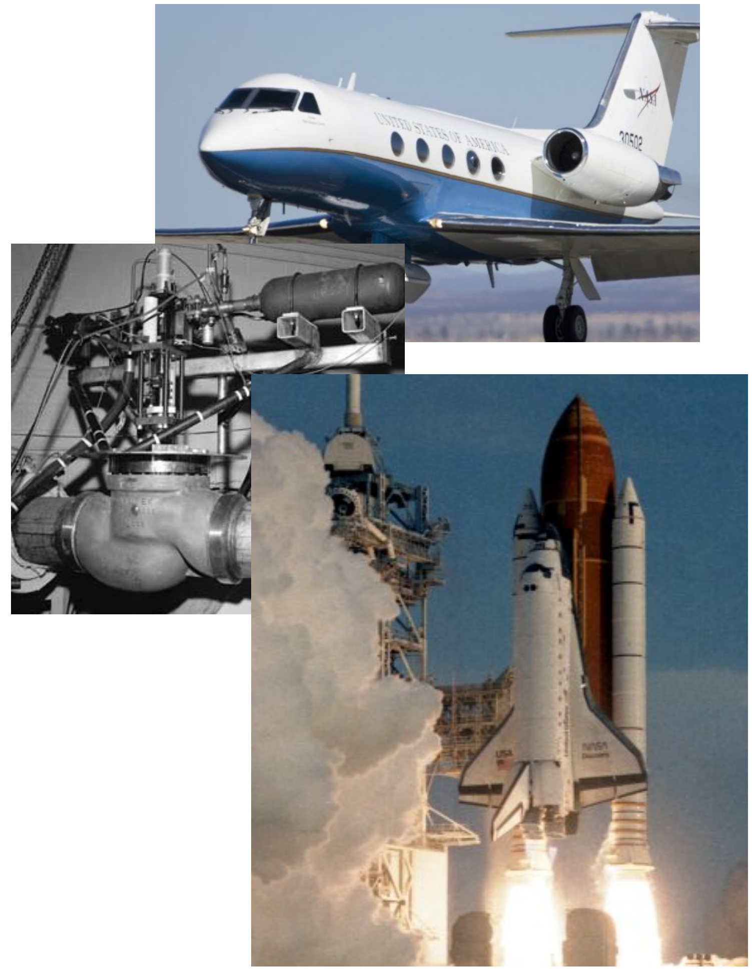
Prognostics



Definition: *Predict damage progression of a fault based on current and future operational and environmental conditions to estimate the time at which a component no longer fulfils its function within desired specs ("Remaining Useful Life")*

Motivation

- Key to condition-based maintenance
  - Improve mission safety
  - Avoid shutdowns/launch scrap
  - Reduce unscheduled maintenance
  - Improved operational efficiency
- Challenges
  - Examples of fault progression are difficult to find due to periodic maintenance and component replacement
  - Sensor noise makes it hard to distinguish small, gradual deviations in performance
  - Limited sensor sets
    - e.g., only discrete open/closed sensors for valves



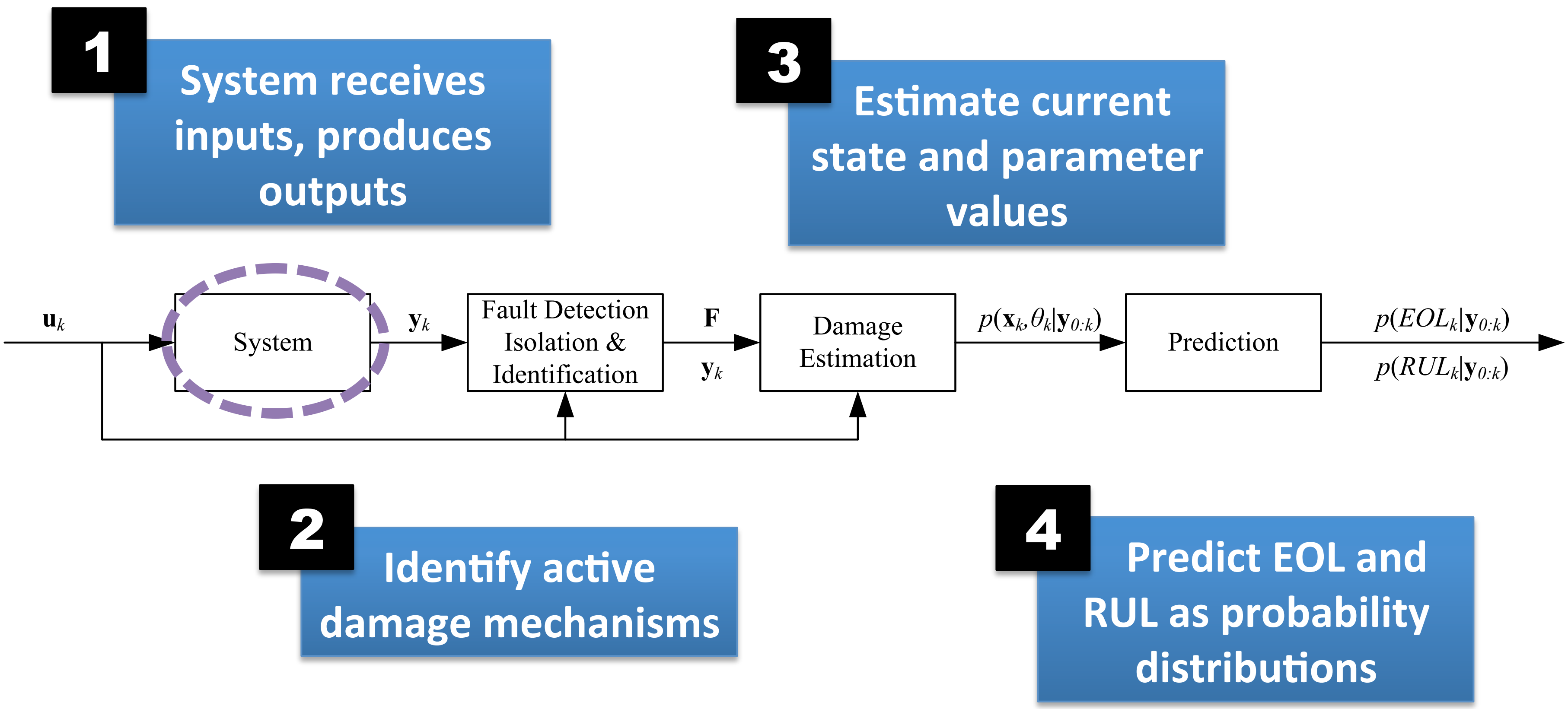
Key Ingredients for Prognostics

- Run-to-failure data
  - Measurement data
  - Ground truth data
  - Operational conditions
    - Load profiles
    - Environmental conditions
  - Failure threshold
- Physics of Failure models
  - For each fault in the fault catalogue
- Uncertainty information
  - Sources of uncertainty
  - Uncertainty characterization

Prognostic Algorithms

- **Data-driven algorithms** rely on large run-to-failure data sets
  - Learn health progression from examples
  - Large set of run-to-failure trajectories needed to correctly train algorithm
  - Need to deal with loss of sensors or lack of sensors
- **Model-based approaches** exploit domain knowledge in the form of a model
  - Use physics knowledge of components and their failures
  - Viable approach when large data sets are not available
  - Can be robust to sensor loss and still work under limited sensing environments
  - Same general approach may be applied to any component/system, only the model changes

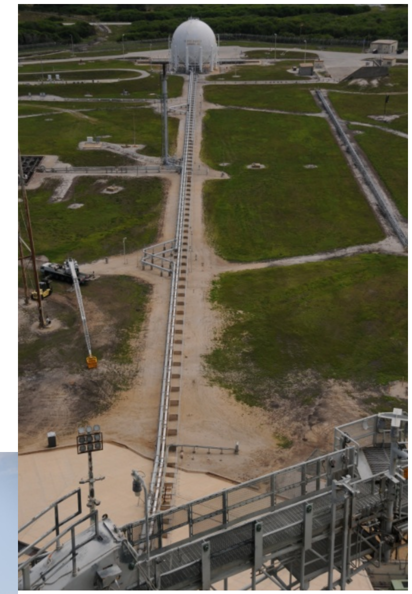
Prognostics Architecture



Case Study

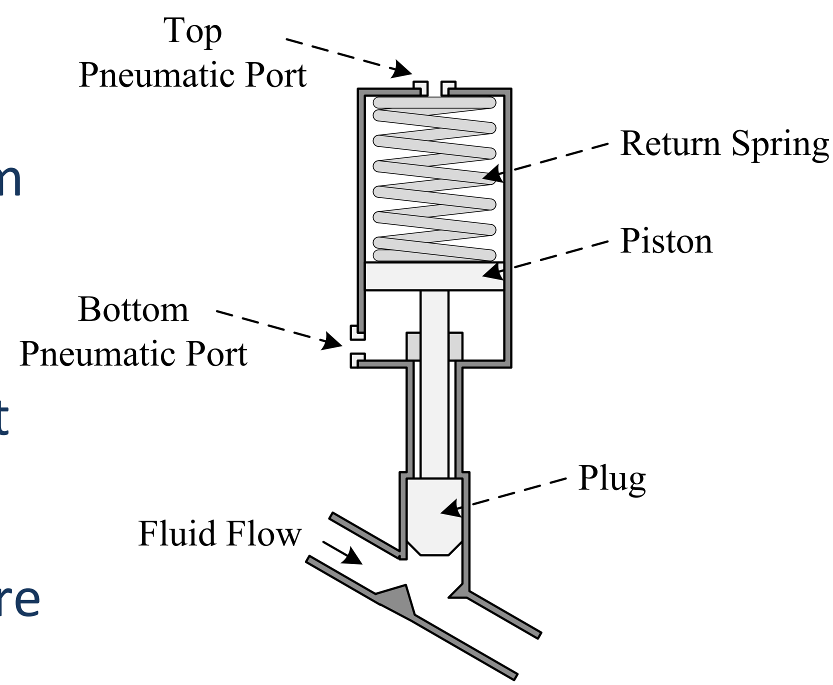
- Shuttle refueling operation
  - Liquid fuels
  - Cryogenic environment
  - Legacy equipment



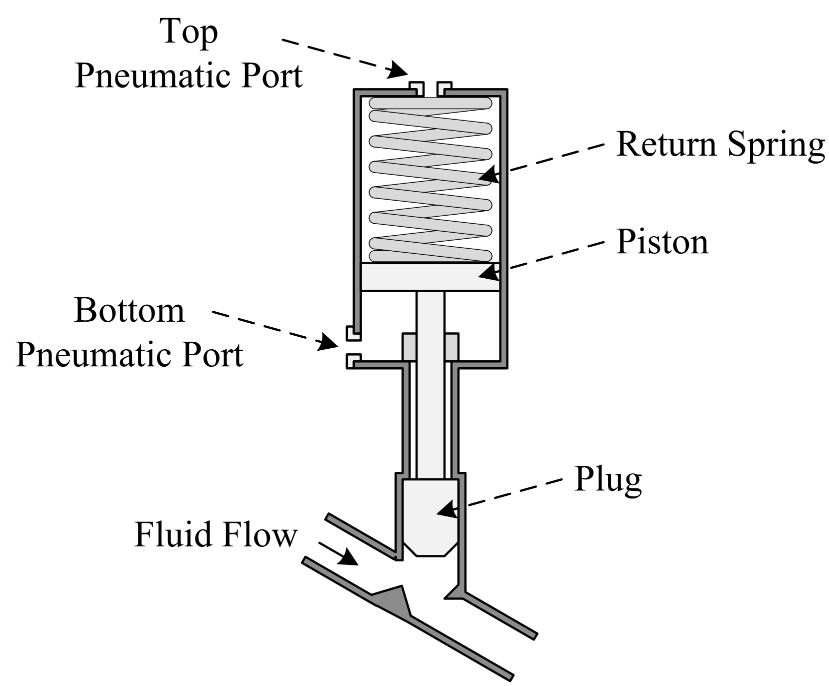
Case Study

- Apply framework to pneumatic valve
  - Complex mechanical devices used in many domains including aerospace
  - Failures of critical valves can cause significant effects on system function
- Pneumatic valve operation
  - Valve opened by opening bottom port to supply pressure and top port to atmosphere
  - Valve closed by opening bottom port to atmosphere and top port to supply pressure
  - Return spring ensures valve will close upon loss of supply pressure



Case Study

- Faults
  - External leaks at ports & internal leaks across piston
  - Friction buildup due to lubrication breakdown, sliding wear, buildup of particulate matter
  - Spring degradation
- Defining EOL
  - Limits defined for open and close times of valves
    - E.g., main fill valve opens in 20 seconds (26 req.), closes in 15 (20 req.)
  - Limits placed on valve leakage rates (pneumatic gas)
  - Valve must be able to fully close upon fail-safe
  - Valve is at EOL when any of above conditions violated (defines C_{EOL})
    - Function of amount of damage, parameterized in model



Physics-based Modeling

- Valve state defined by

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ v(t) \\ m_t(t) \\ m_b(t) \end{bmatrix}$$

Valve position
Valve velocity
Gas mass above piston
Gas mass below piston

- State derivatives given by

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} v(t) \\ \frac{1}{m}\sum F(t) \\ f_t(t) \\ f_b(t) \end{bmatrix}$$

Velocity
Acceleration
Gas flow above piston
Gas flow below piston

- Inputs given by

$$\mathbf{u}(t) = \begin{bmatrix} p_l(t) \\ p_r(t) \\ u_t(t) \\ u_b(t) \end{bmatrix}$$

Fluid pressure (left)
Fluid pressure (right)
Input pressure at top port
Input pressure at bottom port

Physics-based Modeling: Forces

- Piston movement governed by sum of forces, including
  - Pneumatic gas: $(p_b(t) - p_t(t))A_p$
  - Process fluid: $(p_r(t) - p_l(t))A_v$
  - Weight: $-mg$
  - Spring: $-k(x(t) - x_o)$
  - Friction: $-rv(t)$
  - Contact forces: $\begin{cases} k_c(-x), & x < 0 \\ 0, & 0 \le x \le L_s \\ -k_c(x - L_s), & x > L_s, \end{cases}$

$$\boxed{\begin{aligned} p_t(t) &= \frac{m_t(t)R_gT}{V_{t_0} + A_p(L_s - x(t))} \\ p_b(t) &= \frac{m_b(t)R_gT}{V_{b_0} + A_px(t)} \end{aligned}}$$

Valve Stroke Length

Physics-based Modeling: Flows

- Gas flows determined by choked/non-choked orifice flow equations:

$$f_t(t) = f_g(p_t(t), u_t(t))$$
$$f_b(t) = f_g(p_b(t), u_b(t))$$

$$f_g(p_1,p_2) = \begin{cases} C_s A_s p_1 \sqrt{\frac{\gamma}{Z R_g T}\left(\frac{2}{\gamma+1}\right)^{(\gamma+1)/(\gamma-1)}}, & p_1 \geq p_2 \wedge p_1/p_2 \geq \left(\frac{\gamma+1}{2}\right)^{\gamma/(\gamma-1)} \\ C_s A_s p_1 \sqrt{\frac{2}{Z R_g T}\left(\frac{\gamma}{\gamma-1}\right)\left(\left(\frac{p_2}{p_1}\right)^{2/\gamma} - \left(\frac{p_2}{p_1}\right)^{(\gamma+1)/\gamma}\right)}, & p_1 \geq p_2 \wedge p_1/p_2 < \left(\frac{\gamma+1}{2}\right)^{\gamma/(\gamma-1)} \\ C_s A_s p_2 \sqrt{\frac{\gamma}{Z R_g T}\left(\frac{2}{\gamma+1}\right)^{(\gamma+1)/(\gamma-1)}}, & p_1 < p_2 \wedge p_2/p_1 \geq \left(\frac{\gamma+1}{2}\right)^{\gamma/(\gamma-1)} \\ C_s A_s p_2 \sqrt{\frac{2}{Z R_g T}\left(\frac{\gamma}{\gamma-1}\right)\left(\left(\frac{p_1}{p_2}\right)^{2/\gamma} - \left(\frac{p_1}{p_2}\right)^{(\gamma+1)/\gamma}\right)}, & p_1 < p_2 \wedge p_2/p_1 < \left(\frac{\gamma+1}{2}\right)^{\gamma/(\gamma-1)} \end{cases}$$

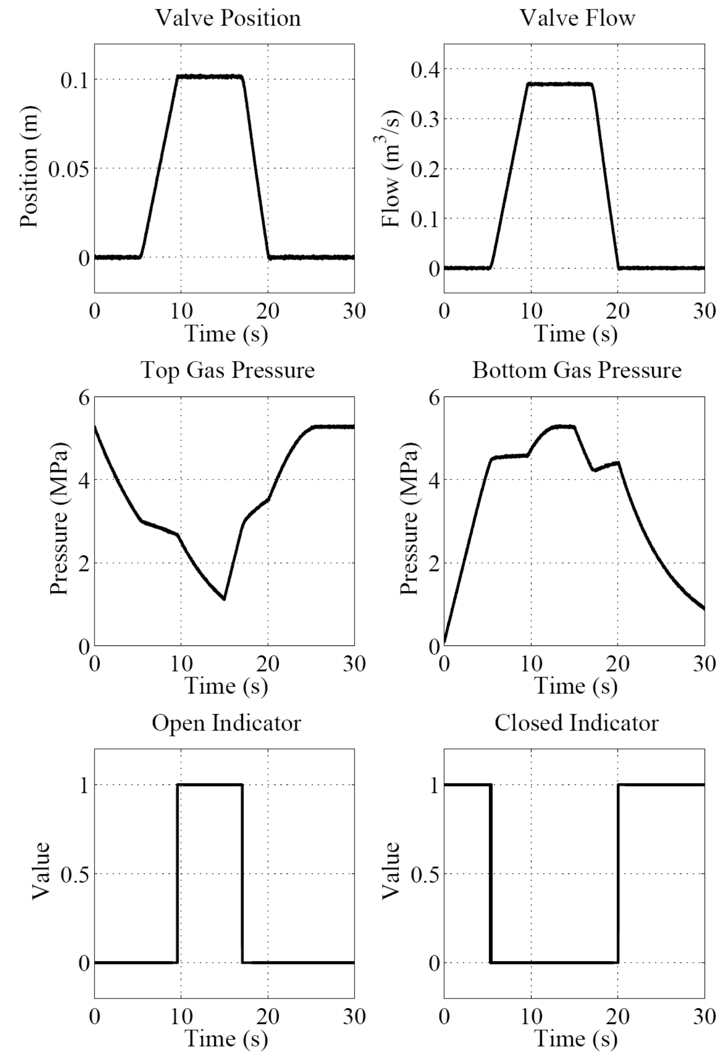
- Fluid flow determined by orifice flow equation:

$$f_v(t) = \frac{x(t)}{L_s} C_v A_v \sqrt{\frac{2}{\rho}\,|p_{fl} - p_{fr}|}\,\mathrm{sign}(p_{fl} - p_{fr})$$

Pneumatic Valve Modeling

- Possible sensors include

$$\mathbf{y}(t) = \begin{bmatrix} x(t) \\ p_t(t) \\ p_b(t) \\ f_v(t) \\ open(t) \\ closed(t) \end{bmatrix} \begin{array}{l} \text{Valve position} \\ \text{Gas pressure (top)} \\ \text{Gas pressure (bottom)} \\ \text{Fluid flow} \\ \text{Open indicator} \\ \text{Closed Indicator} \end{array}$$

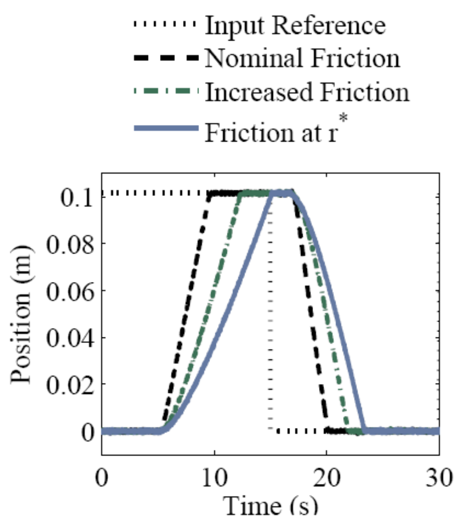


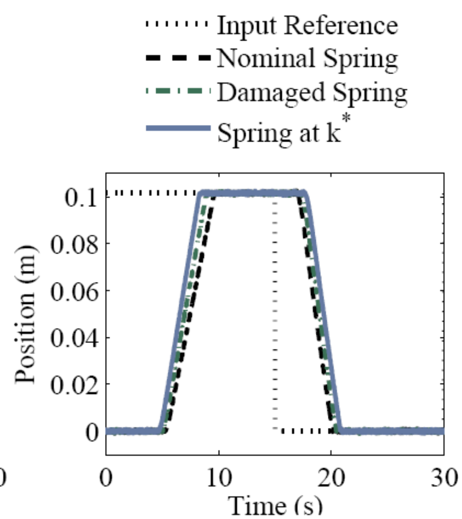
where,

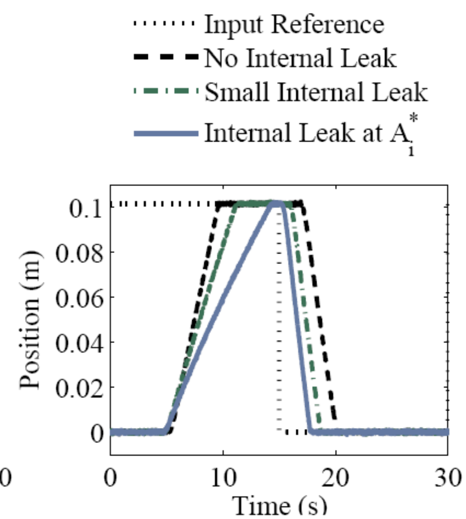
$$open(t) = \begin{cases} 1, & \text{if } x(t) \geq L_s \\ 0, & \text{otherwise} \end{cases}$$

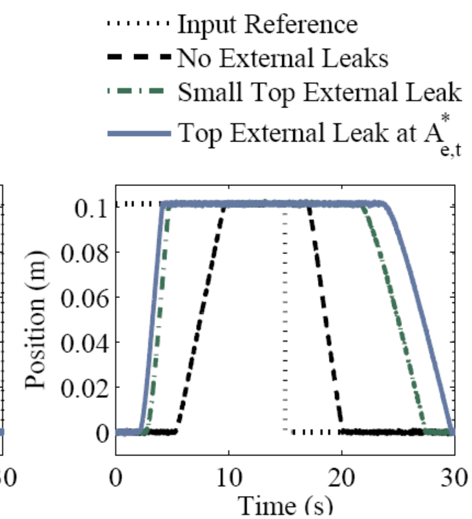
$$closed(t) = \begin{cases} 1, & \text{if } x(t) \leq 0 \\ 0, & \text{otherwise} \end{cases}$$

Modeling Damage









Increase in friction

- Based on sliding wear equation
- Describes how friction coefficient changes as function of friction force, piston velocity, and wear coefficient

$$\dot{r}(t) = w_r |F_f(t) v(t)|$$

Degradation of spring

- Assume form similar to sliding wear equation
- Describes how spring constant changes as function of spring force, piston velocity, and wear coefficient

$$\dot{k}(t) = -w_k |F_s(t) v(t)|$$

Growth of internal leak

- Based on sliding wear equation
- Describes how leak size changes as function of friction force, piston velocity, and wear coefficient

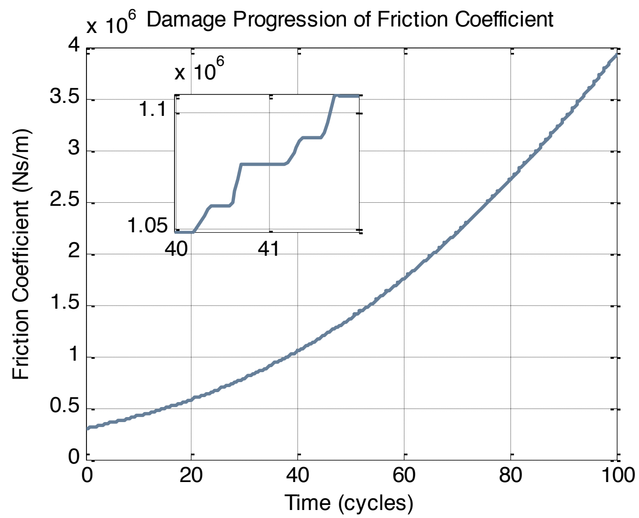
$$\dot{A}_i(t) = w_i |F_f(t) v(t)|$$

Growth of external leak

- Based on environmental factors such as corrosion
- Assume a linear change in absence of known model

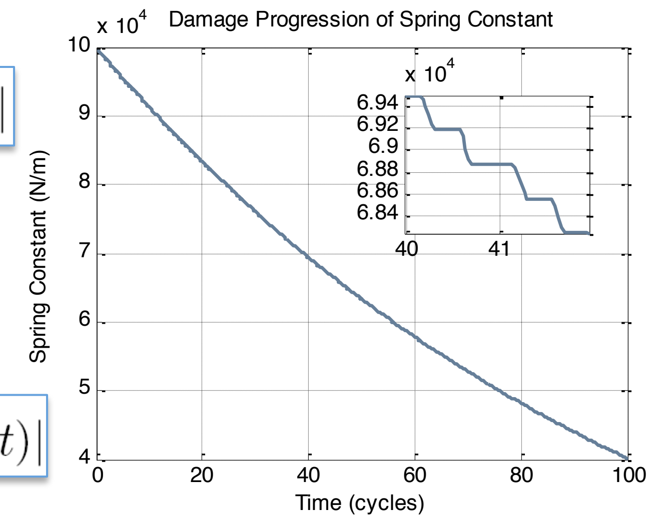
$$\dot{A}_e(t) = w_e$$

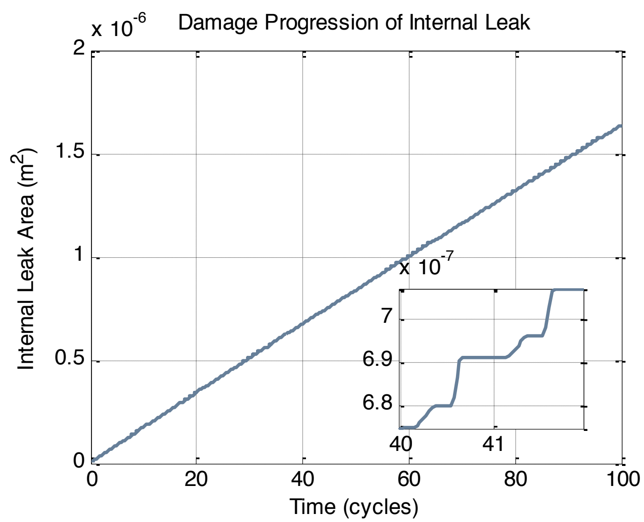
Damage Progression



$$\dot{r}(t) = w_r |F_f(t)v(t)|$$

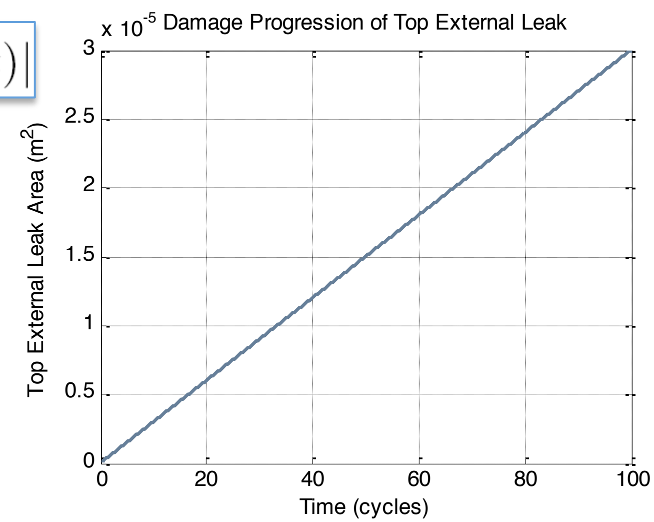
$$\dot{k}(t) = -w_k |F_s(t)v(t)|$$





$$\dot{A}_i(t) = w_i |F_f(t)v(t)|$$

$$\dot{A}_e(t) = w_e$$



Damage Estimation

- Wear parameters are unknown, and must be estimated along with system state

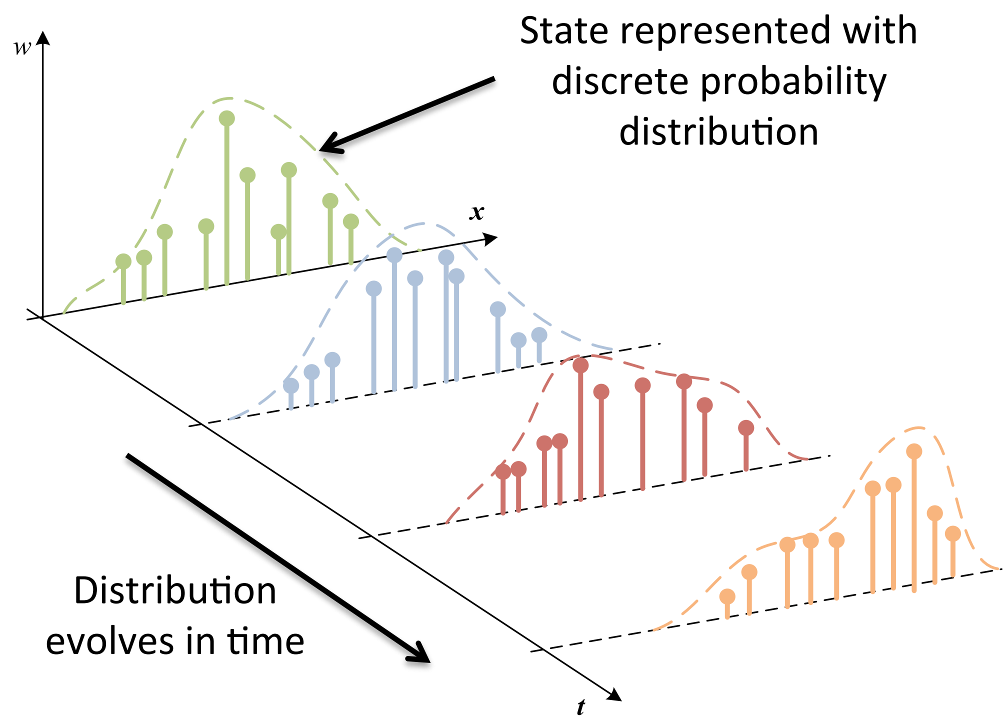
Augment system state with unknown parameters and use state observer

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ v(t) \\ m_t(t) \\ m_b(t) \\ r(t) \\ k(t) \\ A_i(t) \\ A_{e,t}(t) \\ A_{e,b}(t) \end{bmatrix} \begin{array}{l} \text{Position} \\ \text{Velocity} \\ \text{Gas mass above piston} \\ \text{Gas mass below piston} \\ \text{Friction coefficient} \\ \text{Spring rate} \\ \text{Internal leak area} \\ \text{External leak area (top)} \\ \text{External leak area (bottom)} \end{array}$$

$$\boldsymbol{\theta}(t) = \begin{bmatrix} w_r(t) \\ w_k(t) \\ w_i(t) \\ w_{e,t}(t) \\ w_{e,b}(t) \end{bmatrix} \begin{array}{l} \text{Friction wear} \\ \text{Spring wear} \\ \text{Internal leak wear} \\ \text{External leak wear (top)} \\ \text{External leak wear (bottom)} \end{array}$$

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



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:

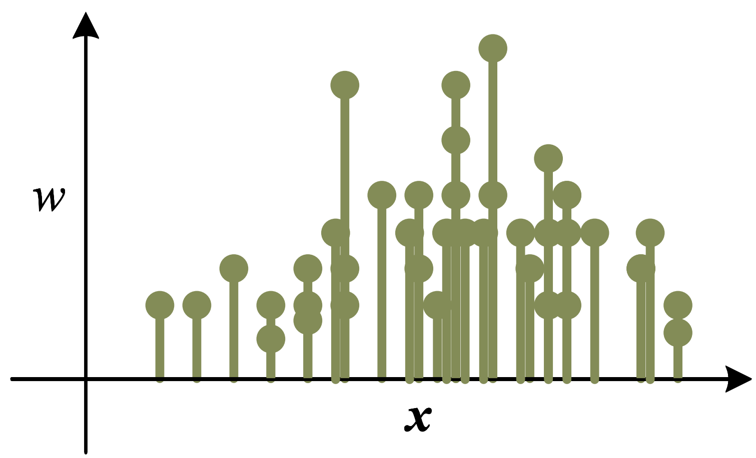
$$\theta_k = \theta_{k-1} + \xi_{k-1}$$

  - 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

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

Sampling Importance Resampling

- 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



Prediction

- Particle filter computes

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

- Prediction n steps ahead approximated as

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

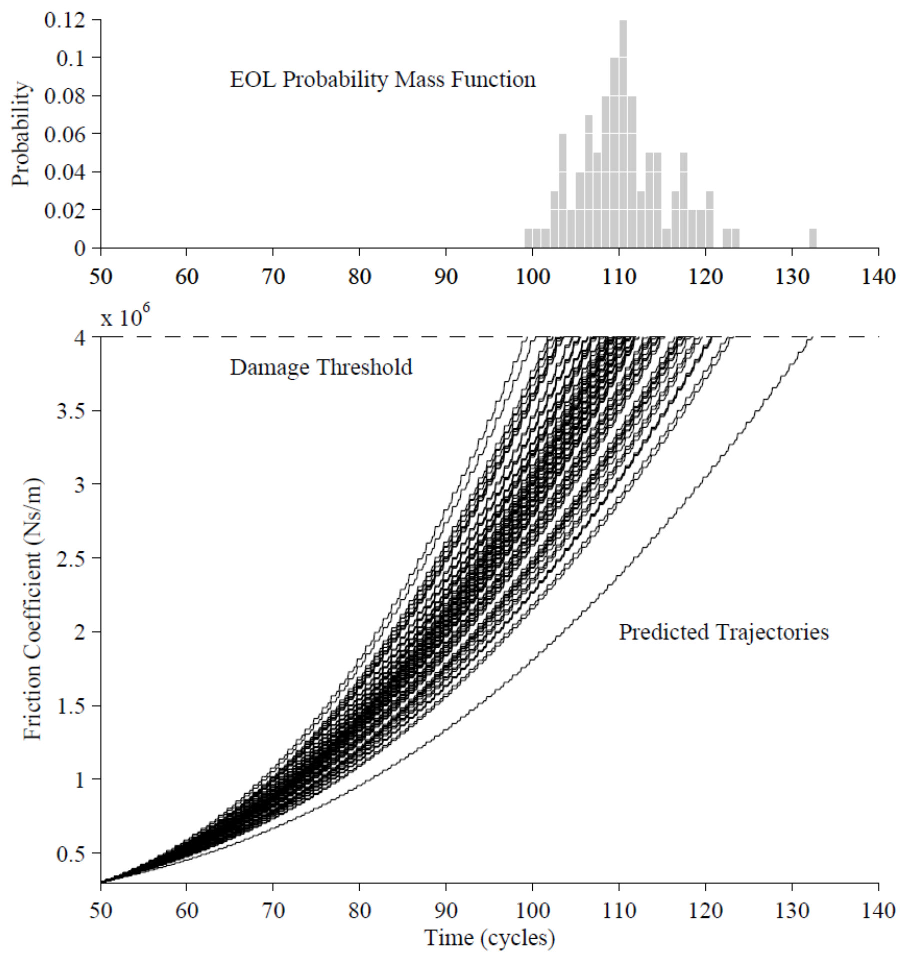
- Similarly, EOL prediction approximated as

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

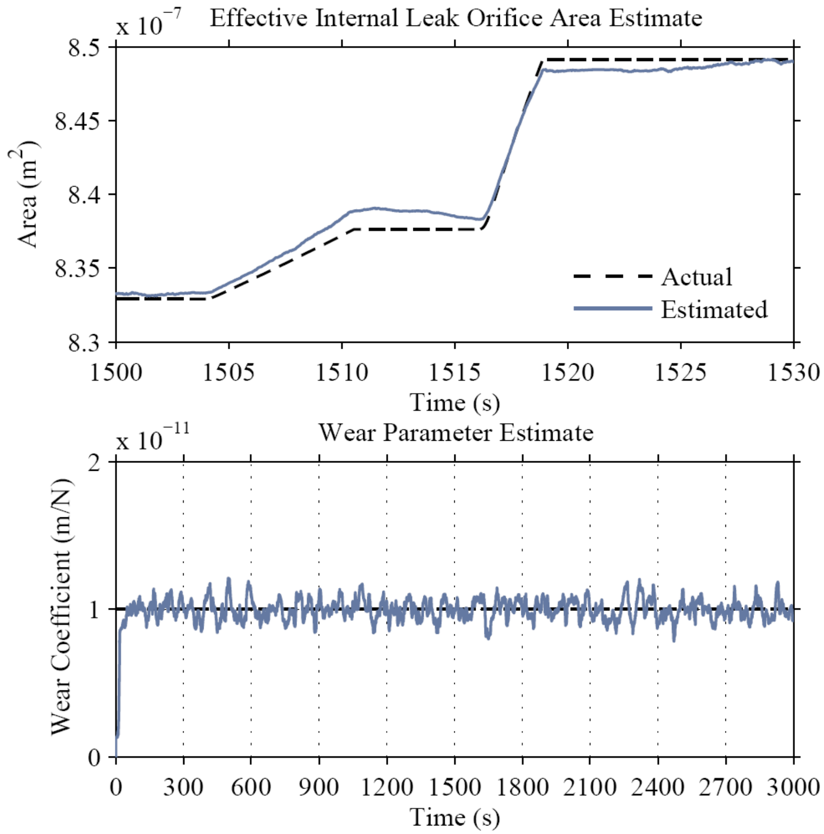
- General idea
  - Propagate each particle forward until EOL reached (condition on EOL evaluates to true)
  - Use particle weights for EOL weights

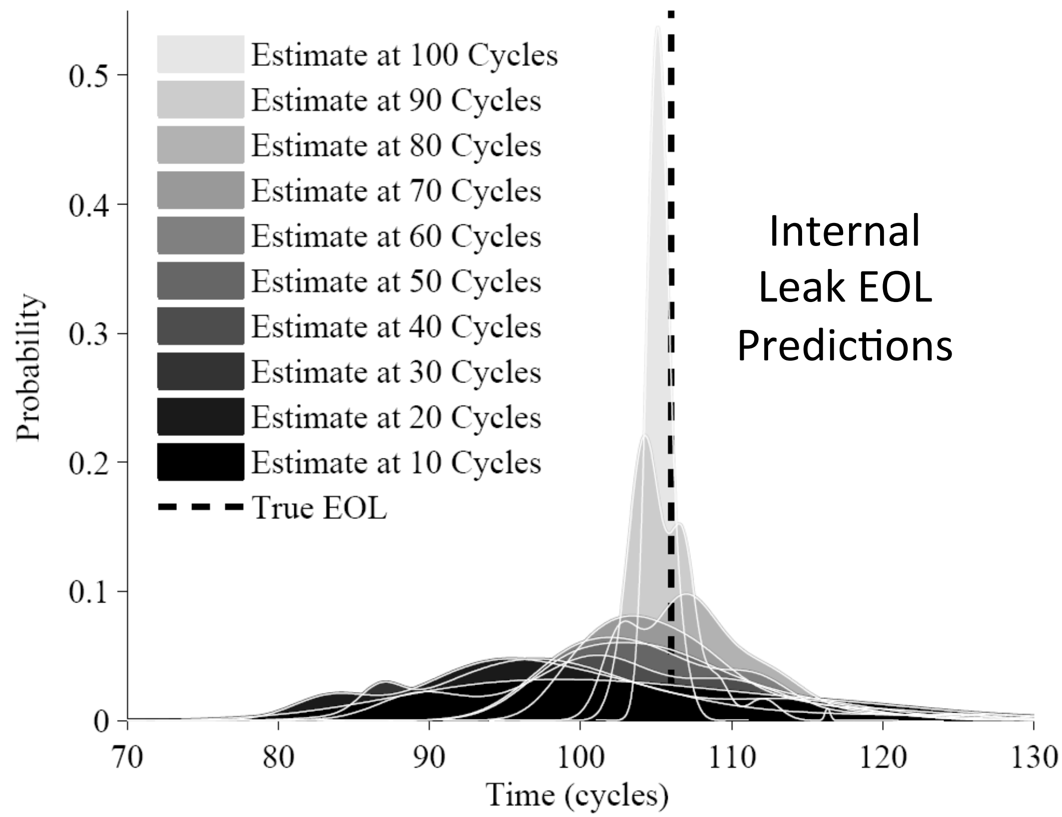
It'll Break at this Time:

- Friction progression EOL prediction



Validation of Methodology





Internal Leak EOL Predictions

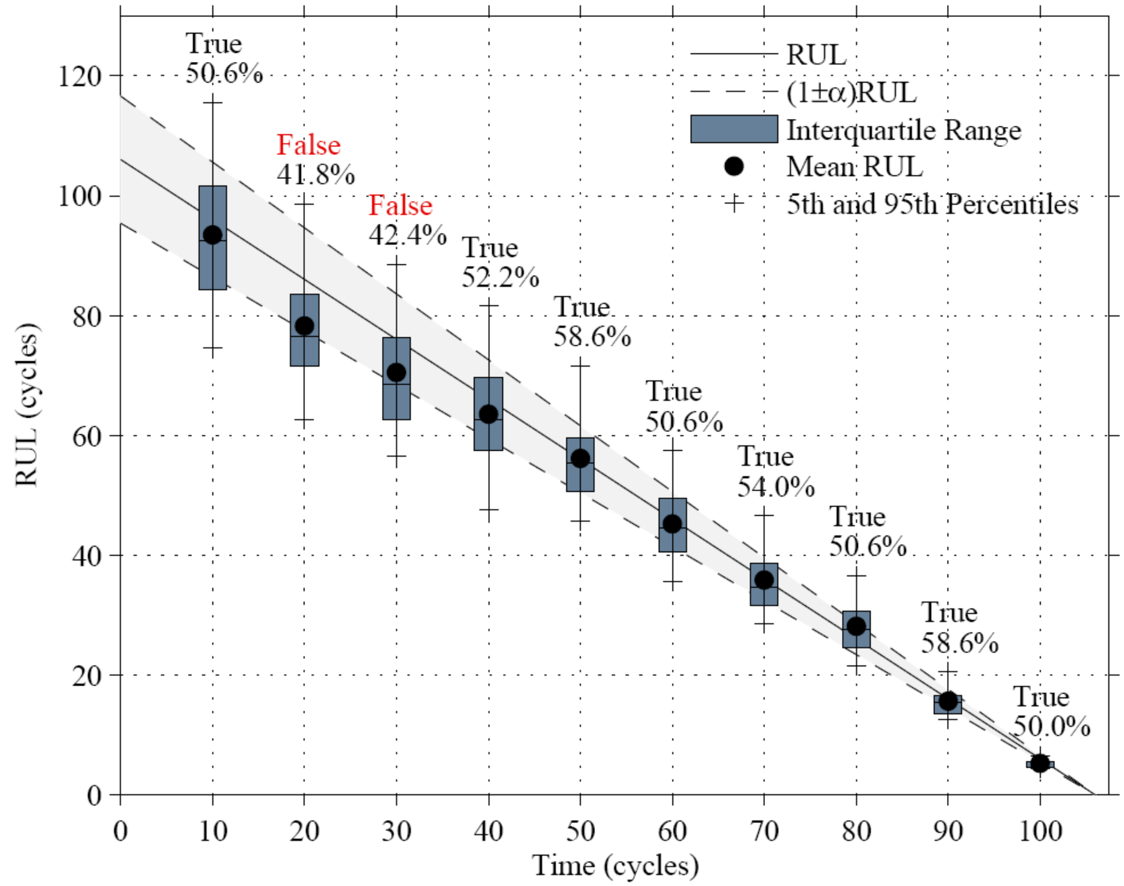
Estimate of wear parameter converges after a few cycles, after this, leak area can be tracked well.

EOL predictions all contain true EOL, and get more accurate and precise as EOL is approached.

α-λ Performance

- Plot summarizes performance of internal leak prognosis
- Over 50% of probability mass concentrated within the bounds at all prediction points except at 20 and 30 cycles
  - Mean RULs are within the bounds at these points
- For α=0.122, metric is satisfied at all points

α=0.1, β=0.5



Some Current and Past Activities in ISHM

ROCKET ENGINE TEST STAND

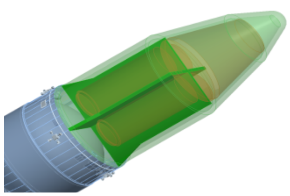
ROBOTIC SPACE FLIGHT

LCROSS
Ground-Based Root Cause Determination; Data Analysis

AERONAUTICS

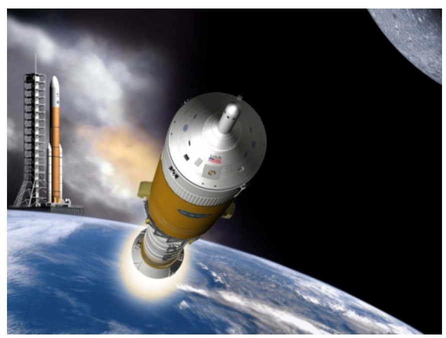
IVHM
On-board and off-board Diagnostics, Prognostics, Logistics

HUMAN SPACE FLIGHT



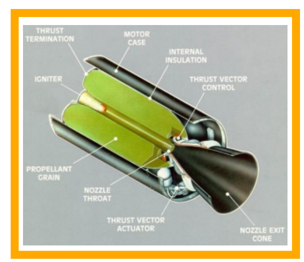
Composites Shroud

Ground Diagnostics for CLV and Ground Test / Integration Infrastructure



CLV Crew Abort Logic Development

Space Station Fault Analysis



Solid Rocket Motor Failure Detection and Prediction

Space Shuttle Main Engine Abnormal Condition Detection

Data Analysis / Mining for Mission Ops

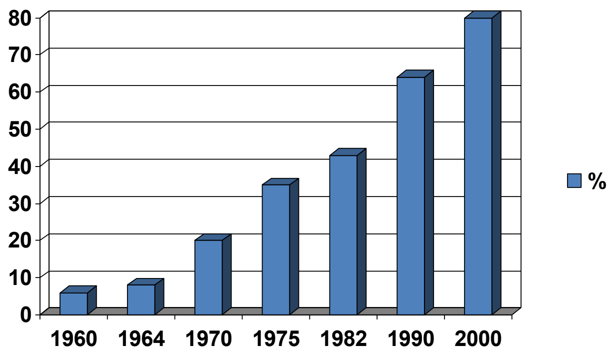
S&T Challenge

Learning and Adaptation

- Our knowledge of the space environment decreases drastically as we explore beyond the earth's atmosphere
  - Practical limits to how much "a priori" knowledge can be stored on board
- Beyond earth orbit, autonomy is a critical enabler for exploration (with or without a crew)
- Science return from robotic spacecraft can be significantly increased if these spacecraft can learn from their environment and adapt
  - Serendipitous science
  - Novelty detection
  - Automated discovery
  - Accurate response to unforeseen failures

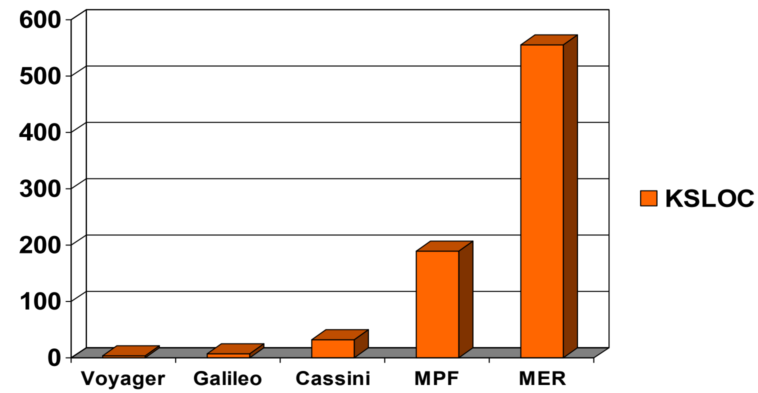
S&T Challenge

Software Complexity



% Functionality in Software in Military Aircraft

Source Lines of Code in NASA Robotic Spacecraft



- Traditional flight software certification requires exhaustive testing:
  - Of all nominal execution traces (all possible branches) of the software
  - In response to all input commands and allowable sensor values
  - Of known failure modes
- Simply not possible for health management systems of reasonable complexity
  - More R&D needed in automated verification and validation
  - Need methods and tools to V&V adaptive systems
  - Model-based software development (autocoding) to reduce cost of development and testing
  - Flight certification methods need to accommodate the unique needs of health management systems.

S&T Challenge

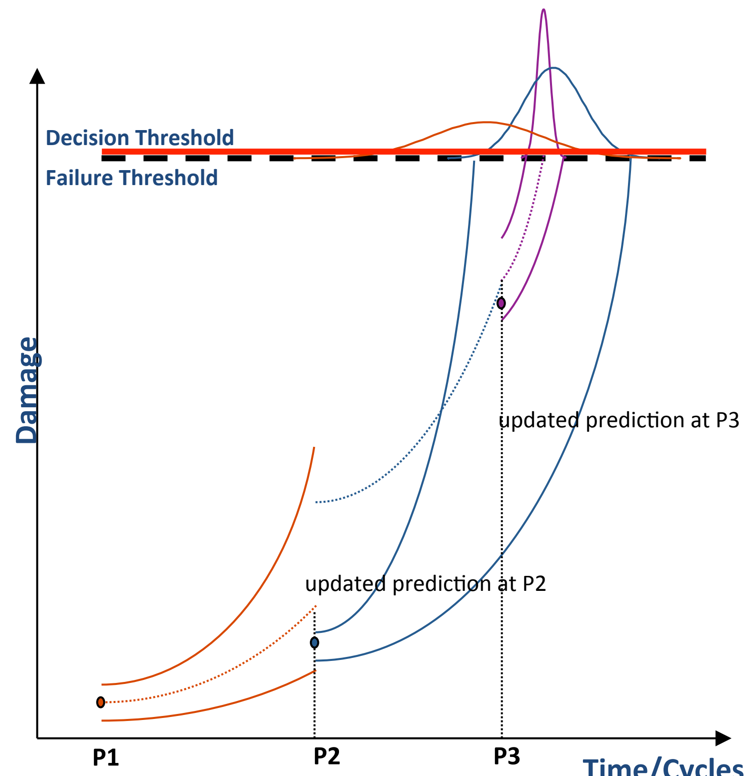
Decision Making

- PHM information is only a means to an end
- Integrate Diagnostic and Prognostic information with
  - Logistics
  - Fleet management/mission management objectives
  - Operations
  - …
- Path forward
  - Automated reconfiguration
  - Decision process is multi-objective dynamic optimization

S&T Challenge

Uncertainty Management

- Quantification of uncertainty is a key in being able to realize value of remaining life estimates
- Methods to quantify and manage uncertainty lacking
- Standardized metrics to express uncertainty in PHM context lacking



Closing Thoughts

- Integrated System Health Management is a systematic engineering discipline where health management principles are applied to systems
  - Seen more and more as an enabler for aerospace applications
  - Prognostics is a relatively new technology that promises to predict time-to-failure
- Ongoing activities at NASA cover range of ISHM areas
- Challenges in S&T
  - Learning and adaptive systems
    - Space is the "final frontier" for ISHM
  - Software complexity
    - V&V, certification
  - Uncertainty Management
    - Credible methods to manage uncertainty
  - Decision Making
    - Tie-in to logistics; reconfiguration
- Implementation will be one step at a time
  - Finding the right applications is crucial
  - Ground -> Aircraft -> Robotic craft -> Human space flight
  - Increasing level of comfort and confidence over time
  - Proving benefit over cost
  - Taming software complexity

- last slide