Prognostics and Health Management
ENME 808A – Fall 2007

guest lecture

Methods for Making Predictions:

model based methods, data driven methods and hybrid approaches

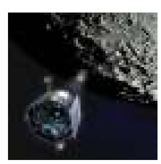
Kai Goebel, Ph.D.
NASA Ames Research Center
Prognostics Center of Excellence
kai.goebel@nasa.gov

Work referenced in these slides was performed in large part while author was with GE
It contains material from public presentations made by the author and papers co-authored as well as material made available from other parties, including contributions by Piero Bonissone, Bill Cheetham, Neil Eklund, Weizhong Yan and others from GE Global Research and George Vachtsevanos and Bhaskar Saha from Georgia Tech
See also list of references on last slide

Some Current NASA Activities in PHM

ROBOTIC SPACE FLIGHT

TacSat 3
Intelligent Power Management
GN&C Health Management
Automated Sensor Calibration



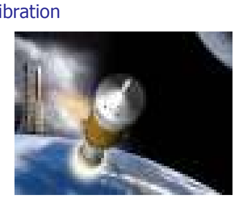
LCROSS
Ground-Based Root Cause Determination; Data Analysis

AERONAUTICS

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

HUMAN SPACE FLIGHT

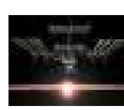
Ground Diagnostics for CLV and Ground Test / Integration Infrastructure



CLV Crew Abort Logic Development

Solid Rocket Motor Failure Detection and Prediction

Space Shuttle Main Engine Abnormal Condition Detection



Space Station Fault Analysis

Data Analysis / Mining for Mission Ops

NASA Ames Prognostics CoE

- Umbrella for prognostic technology development in NASA's health management activities.
- Organization
  - NASA Ames
    - Intelligent Systems division (code TI)
      - Discovery and Systems Health (DaSH) area
        - » Approximately 70 people in five groups
        - » Largest ISHM organization in NASA (nearly 60 engineers and researchers)
        - » Consolidates all ISHM activity at ARC
        - » Broad range of customers across NASA
        - » Mid-TRL (3-7) technology development, maturation, and infusion
        - » PCoE
- Addressing prognostic technology gaps within application areas
  - Aeronautics
  - Exploration science.
- Approximately 20 supporting members.
- Strong affiliations with
  - industry and academia through
    - Grants
    - Space Act Agreements
    - funded IPP, SBIR, STTR, and NRA projects
    - Internships
  - Information exchange with other government organizations (National labs, DoD, …)

Prognostics CoE Activities

- Aeronautics Mission Directorate
  - Aviation Safety Program
    - Integrated Vehicle Health Management (IVHM) program
    - CoE supports technology development for
      - Actuators
        - » EMA
      - Electronic power supplies
        - » Li-Ion batteries
      - Avionics components
        - » Power semiconductors
  - Aging Aircraft and Durability Program
    - Aircraft wiring health management.
      - Model the characteristics of intact and damaged insulation and mechanisms to detect current damage and estimate impact of further deterioration.

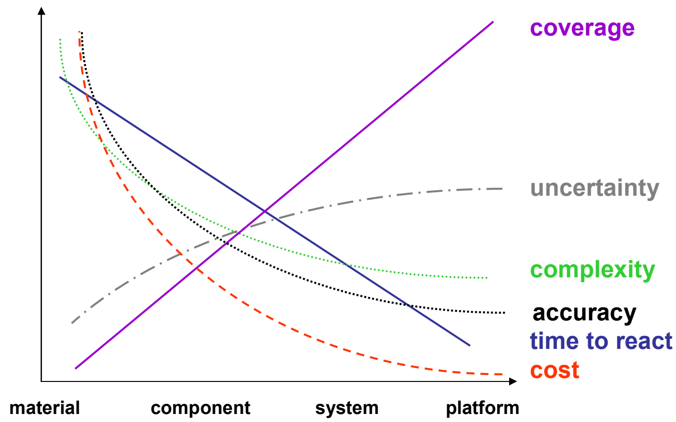
How do you Make Predictions

- Weather
- Stock
- Global Warming
- Other
- Equipment Service Industry
  - New Paradigm: Guarantee Uptime
  - Objective: Make Money
    - Avoid Downtime Cost
      - Direct Penalty to customer
      - Penalty due to safety violations
      - etc.
    - Minimize Maintenance Cost
      - Labor, parts
      - Unscheduled Downtime (see above)
  - Process
    - Monitor equipment
    - Predict failure
    - Decide on action
      - Fix now or later
      - Take logistics into consideration
        - » Shop loading
        - » Parts and resource availability

Prediction Approach

- Stock
  - Using past performance <u>data</u>
  - Is building a stock prediction model easy or hard?
- Weather
  - Using climatographic <u>models</u>
  - What is the expected outcome of using sensor measurements to predict the weather?

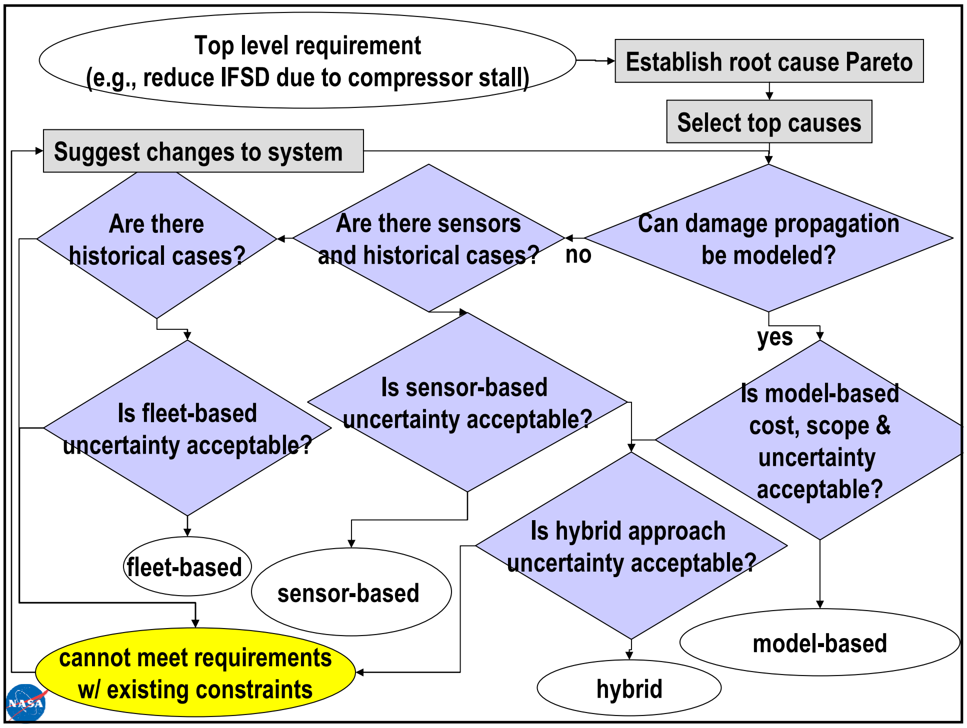
Prognostic Design Tradeoff



- low ← x-axis defined by → high
  - level of abstraction
  - availability of sensor data
  - availability of models

•no one optimal solution for every problem

•selecting an approach



Summary

- Three principal approaches:
  - data driven
  - model based
  - hybrid

- Scale of prognostic effort dramatically influences performance of system – some tradeoffs are required.

- Selection of approach depends on what data and models are available, and is driven primarily by cost and uncertainty constraints.

Data-driven Prediction Techniques

- Sensor-Based Conditioning
  - Multi-Step Adaptive Kalman Filtering
  - Auto-Regressive Moving Average Models
  - Stochastic Auto-Regressive Integrated Moving Average Models (ARIMA)
  - Forecasting by Pattern and Cluster Search
  - Variants Analysis
  - Parameter Estimation Methods
  - Others

Data-driven Prediction Techniques

- Sensor-Based Conditioning (cont'd)
  - AI/CI Techniques
    - Case-Based Reasoning
    - Intelligent Decision-Based Models
    - Min-Max Graphs
    - Support Vector Regression/Relevance Vector Regression
  - Petri Nets
  - Soft Computing Tools
    - Neural Nets
    - Fuzzy Systems
    - Neuro-Fuzzy
    - Random Forests
    - …

Data-driven Prediction Techniques

- Fleet-based Methods
  - Weibull-based lifing
  - Survival analysis
  - …

Model-based Approaches

- Micro-level models (materials-level)
  - e.g.,:
    - Crack growth models
    - Spall growth models

- Macro-level models
  - 1st principle models, e.g.,:
    - Hot Gas Path cycle models

Hybrid Approaches

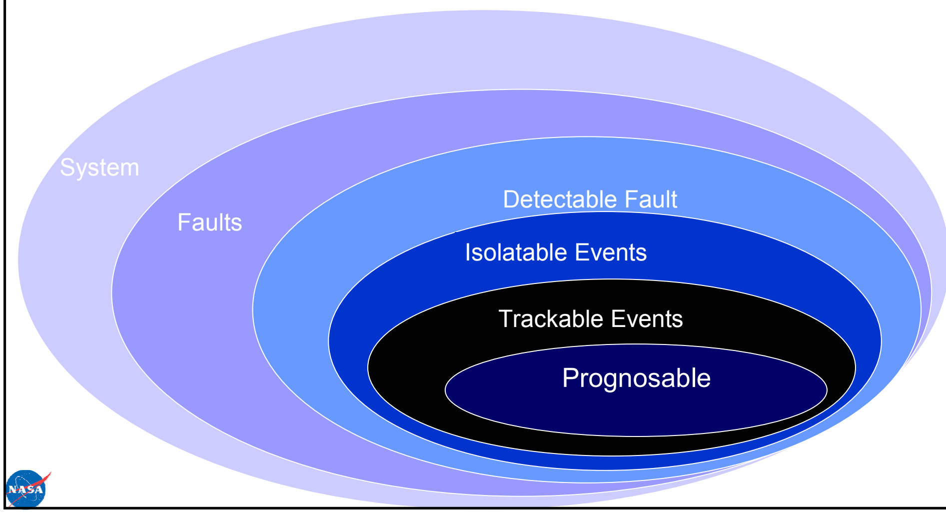
- Pre-Estimate Fusion of Model and Data
- e.g.,:
  - Using thermodynamic engine model and on-wing data for turbine damage propagation calculations
  - Using battery model and impedance measurements for remaining life calculations
- Post-Estimate Fusion of Prognostic Estimates
- e.g.,:
  - Fuse model-based model output and data-driven model output to reduce prediction uncertainty for bearing spall prognostics

Methods for Making Predictions

- Model-Based Methods
  - Micro-level models (material-level)
  - Macro-level models (1st principle-level)
- Data-driven Methods
  - Sensor-Based Conditioning
    - Regression
    - Case-based Reasoning
    - Neural Networks
    - SVM/RVM
    - Statistical Process Control
    - Random Forests
    - others
  - Fleet-based Methods
    - Weibull-based lifing
    - Survival analysis
    - others
- Hybrid Approaches
  - Pre-estimate Fusion of Model and Data
    - Example: Turbine component damage
    - Example: Battery health assessment
  - Post-estimate Fusion of Model and Data
    - Example: Bearing spall

prognostics

- The onion of Fault Detection & Prediction

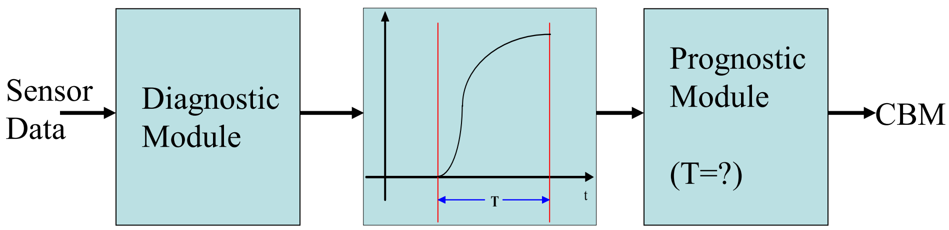


Summary: Requirements

- Basic requirements for prognostics:
  1. Ability to describe system (component, …) health
     a. sophisticated damage propagation model, or
     b. collection of time series sensor measurements, plus a transfer function equating features to health
  2. Definition of failure criterion (i.e., when is health = 0)
  3. Assumptions about future usage (load, cycles, temp, …)
- Additional constraints:
  – Accurate determination of fault mode
    - different fault modes will result in different propagation modes
  – Ability to describe uncertainty of estimates
  – Acceptable risk level

The Prognostic Module

QUESTION: Once an **impending** failure is detected and identified, how can we predict the time left before the failure occurs?



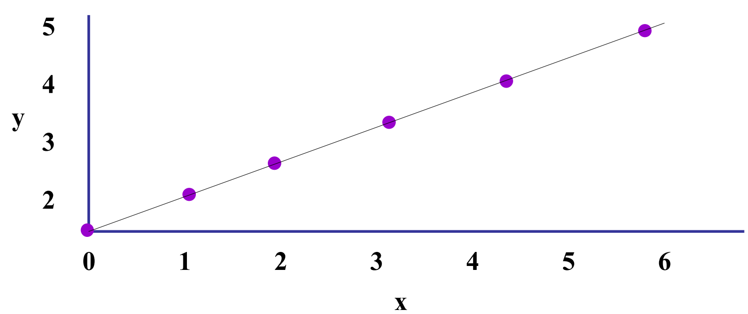
Spring Data

•Experiment	Force(newtons)	Length(inches)
1	1.1	1.5
2	1.9	2.1
3	3.2	2.5
4	4.4	3.3
5	5.9	4.1
6	7.4	4.6
7	9.2	5.0

- What will the length be when the force is 5.0 newtons?
- What will the length be when the force is 10.0 newtons?

Fitting Data

- Statistical method of fitting data to an equations

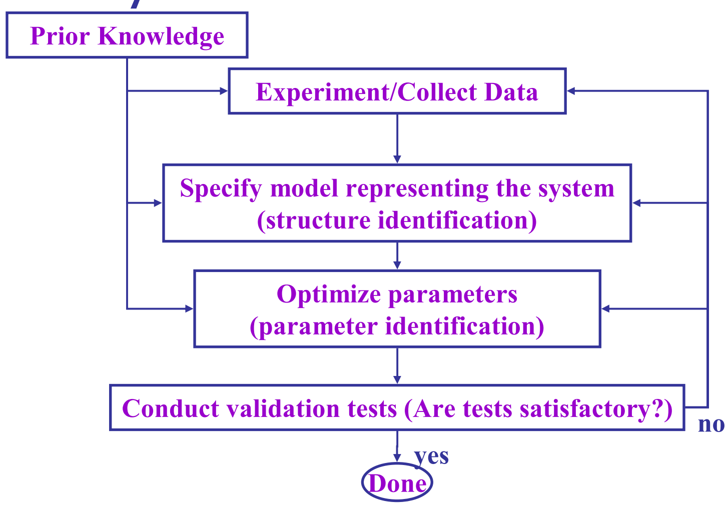


y = x - 3

Establishing a Model

- The problem of determining a mathematical model for an unknown system by observing its input-output data pairs is generally referred to as <u>system identification</u>
- The purposes of system identification are
  - to predict a system's behavior,
  - to explain the interactions and relationships between inputs and outputs, and
  - to design a controller or simulation of the system

System Identification Process



– Structure and parameter identification may need to be done repeatedly

11/29/2007

Structure Identification

- Determine the class of models within which the search for the most suitable model is conducted. $y = f(u;\theta)$ where u is the input vector and θ is the parameter vector.
- Example: Linear $y = \theta_0 + \theta_1 u_1$,
  Second-order polynomial $y = \theta_0 + \theta_1 u_1 + \theta_2 u_1^2$
- If possible take advantage of domain knowledge
- Else use automated structure ID techniques

11/29/2007

Spring Example

•Structure Identification can be done using domain knowledge.

•The change in length of a spring is proportional to the force applied.

- Hooke's law

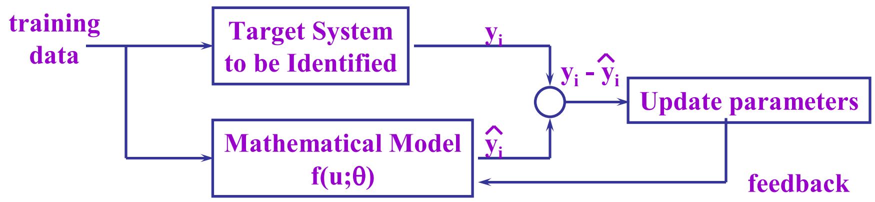
$$\text{length} = k_0 + k_1 \text{ force}$$

11/29/2007

Hooke's Law

- In physics, Hooke's law of elasticity is an approximation which states that the amount by which a material body is deformed (the strain) is linearly related to the force causing the deformation (the stress).
  - In most solids (and in most isolated molecules) atoms are in a state of stable equilibrium.
- For systems that obey Hooke's law, the elongation produced is proportional to the load:
  - $F=-kx$
- where
  - x is the distance the spring is elongated by,
  - F is the restoring force exerted by the spring, and
  - k is the **spring constant** or **force constant** of the spring.
- When this holds, we say that the spring is a **linear spring**.

Parameter Identification



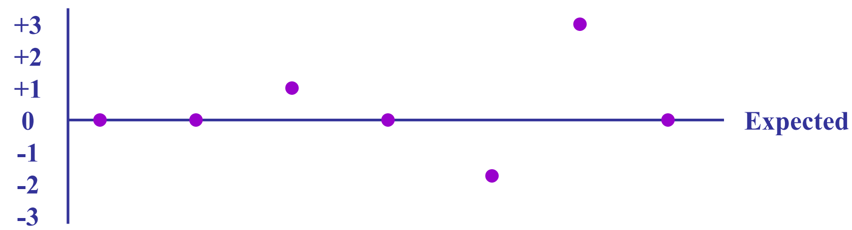
- Training data is used for both system and model.
- Apply optimization technique to tune parameters
  - Difference between Target Systems output, y_i, and Mathematical Model output, $\hat{y}_i$, is used to update parameter vector, θ.
  - Example: Linear model where
    $\theta_0 = -3$
    $\theta_1 = 1$

Least-Squares Parameter Optimization

- Here restricted to:
- linear models $y = u\theta$
  models that have linear parameters
  $y = \theta_0 + \theta_1 u_1$
- static (memory-less) systems
  - output depends on current inputs only.
  - output does not depend on history.

Least-Squares: Error

- Least squared heavily penalizes error for data points that are far from the expected value



Error	0	0	+1	0	-2	+3	0
Error^2	0	0	1	0	4	9	0

Sum of Squared Error = (0 + 0 + 1 + 0 + 4 + 9 + 0) = 14

Least-Squares: Matrix method

- To uniquely identify the unknown vector θ it is necessary that
  - m (# of training items) >= n (# of parameters)
  - training data is $\{(u_i;y_i),\ i = 1, \ldots, m\}$
- If m = n , then we can solve for θ in $y=A\theta$ using

$$\theta = A^{-1} y$$

If m = n = 2 where $f_1(u_i) = u_i^0$ and $f_2(u_i) = u_i^1$ then

$$A = \begin{bmatrix} f_1(u_1) & f_2(u_1) \\ f_1(u_2) & f_2(u_2) \end{bmatrix} = \begin{bmatrix} 1 & u_1 \\ 1 & u_2 \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Least-Squares: Matrix method (2)

- Example: Find a line from two points (1.1, 1.5), (1.9, 2.1)

$$A = \begin{bmatrix} 1 & 1.1 \\ 1 & 1.9 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \qquad y = \begin{bmatrix} 1.5 \\ 2.1 \end{bmatrix}$$

$$A^{-1} = \begin{bmatrix} 2.375 & -1.375 \\ -1.25 & 1.25 \end{bmatrix}$$

$$\theta = A^{-1} y = \begin{bmatrix} 2.375 & -1.375 \\ -1.25 & 1.25 \end{bmatrix} \begin{bmatrix} 1.5 \\ 2.1 \end{bmatrix} = \begin{bmatrix} .675 \\ .75 \end{bmatrix}$$

y = .675 + .75x

Least-Squares for m > n

- When m > n there are more data pairs than fitting parameters.
- An exact solution, satisfying all m equations, is not always possible.
- In order to handle this we need to incorporate an error vector.

$$A\theta + e = y$$

$$\begin{bmatrix} f_1(u_1) & \cdots & f_n(u_1) \\ \vdots & \vdots & \vdots \\ f_1(u_m) & \cdots & f_n(u_m) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} + \begin{bmatrix} e_1 \\ \vdots \\ e_m \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

Least-Squares: m > n (2)

- Best set of parameters θ is the one that minimizes the sum of the squared values of e.
- Error is minimized when

$$\theta = (A^T A)^{-1} A^T y$$

Spring Example

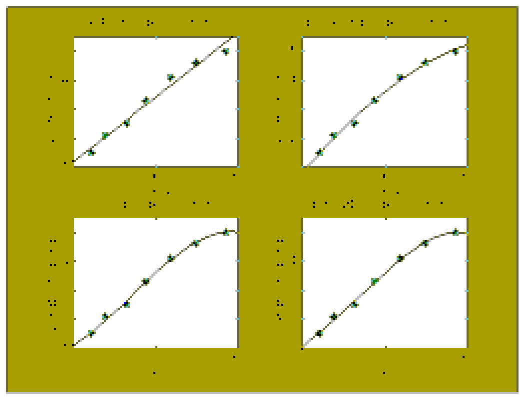
$$A\theta + e = y$$

$$\begin{bmatrix} 1 & 1.1 \\ 1 & 1.9 \\ 1 & 3.2 \\ 1 & 4.4 \\ 1 & 5.9 \\ 1 & 7.4 \\ 1 & 9.2 \end{bmatrix} \begin{bmatrix} k_0 \\ k_1 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 2.1 \\ 2.5 \\ 3.3 \\ 4.1 \\ 4.6 \\ 5.0 \end{bmatrix}$$

$$\begin{bmatrix} k_0 \\ k_1 \end{bmatrix} = (A^T A)^{-1} A^T y = \begin{bmatrix} 1.20 \\ 0.44 \end{bmatrix}$$

y = 1.2 + .44x

Example: Spring data plot

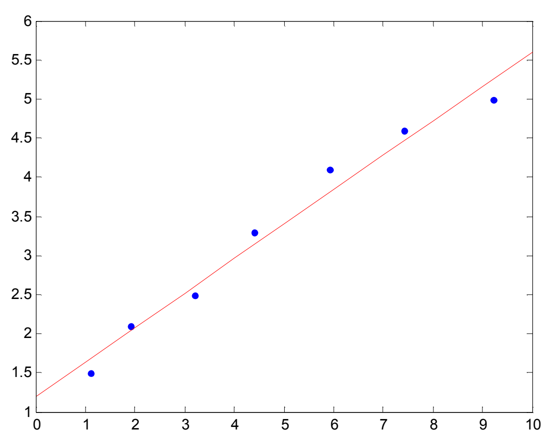


•When force = 5, length = 1.2 + 5 * .44 = 3.4

11/29/2007

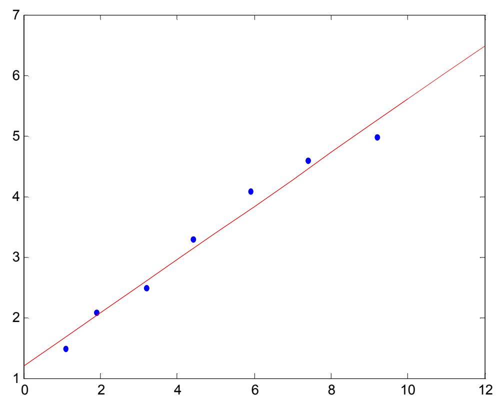
35

Prediction (1)



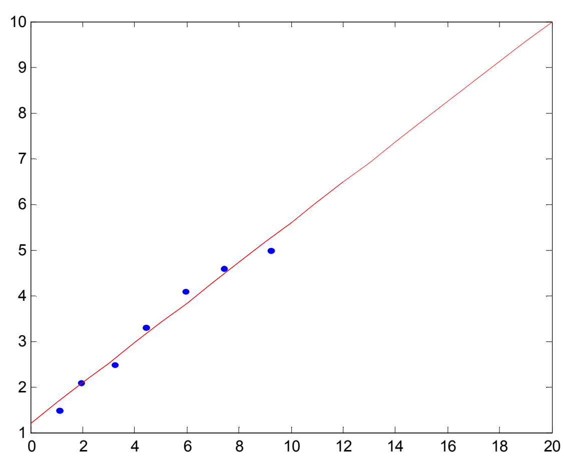
•When force is 10, length = 1.2 + 10 * .44 = 5.6

Prediction (2)



•When force is 12, length = 1.2 + 12 * .44 = 6.48

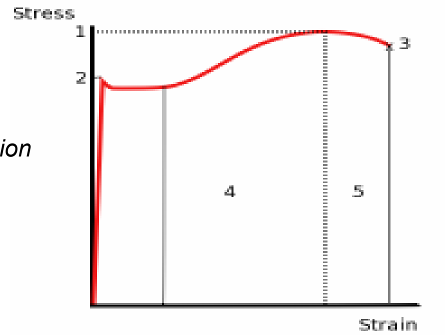
Prediction (3)



•When force is 20, length = 1.2 + 20 * .44 = ?

But

1. *Ultimate strength*
2. *Yield strength*
3. *Rupture*
4. *Strain hardening region*
5. *Necking region.*



Stress-strain curve for low-carbon steel.

Hooke's law is only valid for the portion of the curve between the origin and the yield point.

Non-linear models

- Damage progressions often appears to have exponentially growing characteristics
  - Arrhenius law provides physical interpretation for temperature dependence of a chemical reaction rate

    $$\tau = A \cdot e^{-\left(\frac{Ea}{kT}\right)}$$

  - where
    - τ : Life span
    - *Ea* : Activation energy (eV)
    - *T* : Absolute Temperature (oK)
    - *A* : Constant
    - *k* : Boltzmann's constant
  - Assuming that we know the life span expectancy of a component at a given temperature T_{nom}, we can estimate the acceleration factor for health at a higher temperature T_{stress} using the above formula.

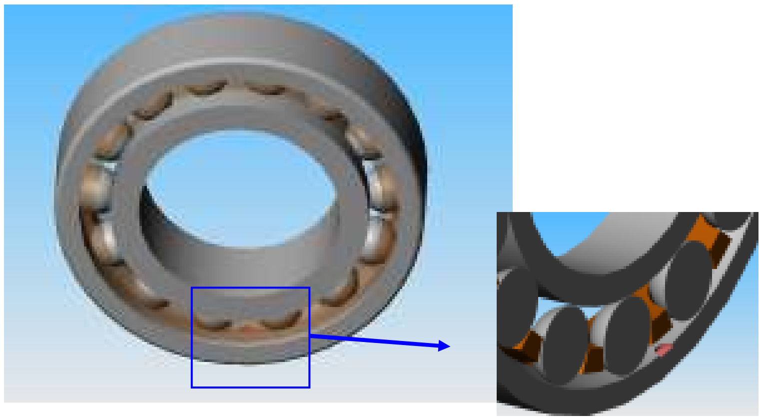
    $$AF(T_{nom}, T_{stress}) = e^{-\left(\frac{Ea}{k_B}\left(\frac{1}{T_{nom}} - \frac{1}{T_{stress}}\right)\right)}$$

  - Other acceleration factors: $AF(P) = f(P_{nom}, P_{stress}, a)$
    - where
      - *P*=stressor (humidity, vibration, etc.)
      - *a* = stress-specific exponent
  - Damage mechanics, Lemaitre 1992, etc.:

    $$\frac{dD}{dN} = f(\sigma, \varepsilon, D)$$

    $$\frac{dD}{dN} = \frac{C_{tr}\varepsilon_P^{\gamma+1}}{\Gamma(\gamma+1)(1-D)^2}$$

Bearings



• designed for "infinite" life
• fail due to hard particle contamination (random occurrence)

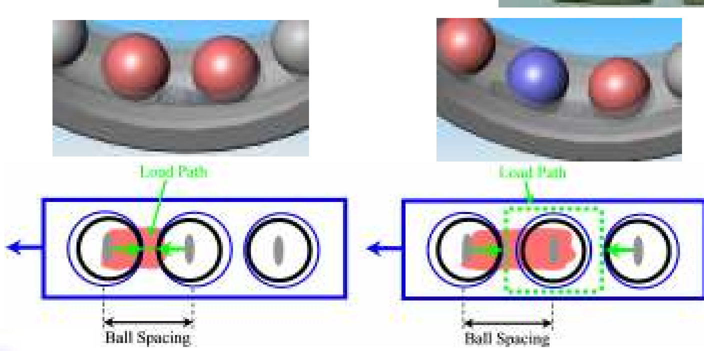
Failure Criterion

Destructive failure occurs when cage fails:

-In low speed bearings, spalls can cover entire race

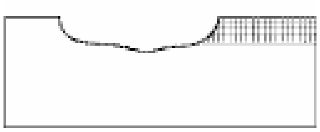
-In high speed bearings, cage fails when spall length >= circumferential ball spacing

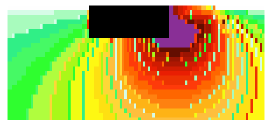
Qualitative Analysis: Stress on cage crossbars and rails increases dramatically when spall length > ball spacing:



Materials-based Spall Propagation Model

damage accumulation:



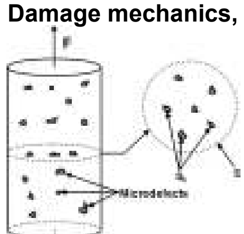


$$\frac{dD}{dN} = f(\sigma, \varepsilon) \qquad \frac{dD}{dN} = f(\sigma, \varepsilon, D)$$

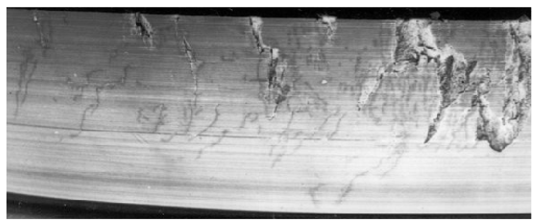
Can't measure damage directly in this application, need to select *f* that yields correct propagation rate for test cases

Damage mechanics, Lemaitre 1992, etc.:

$$\frac{dD}{dN} = \frac{C_{tr}\varepsilon_P^{\gamma+1}}{\Gamma(\gamma+1)(1-D)^2}$$



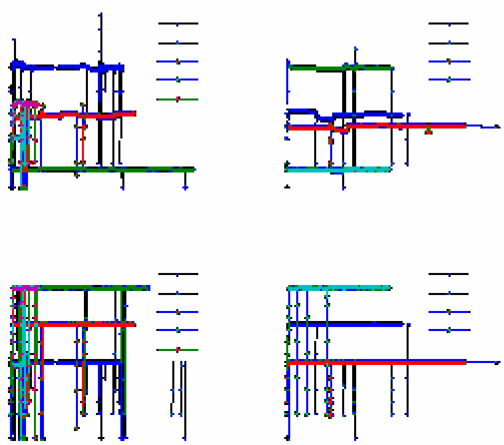
$$D = \frac{S_D}{S} \qquad \tilde{\sigma} = \frac{F}{S(1-D)} = \frac{\sigma}{(1-D)}$$



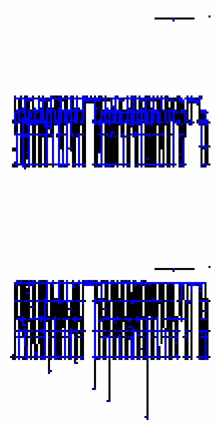
Images courtesy of Sentient Corporation

Experimental Runs

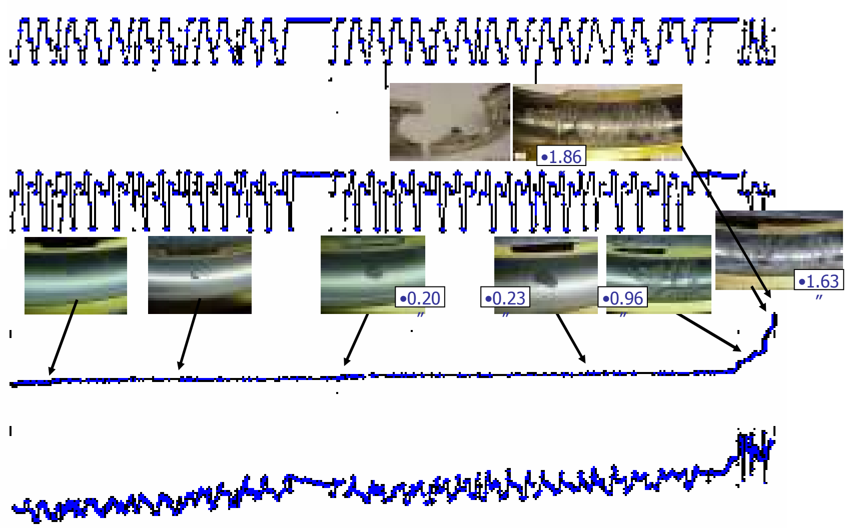
Steady-State Runs



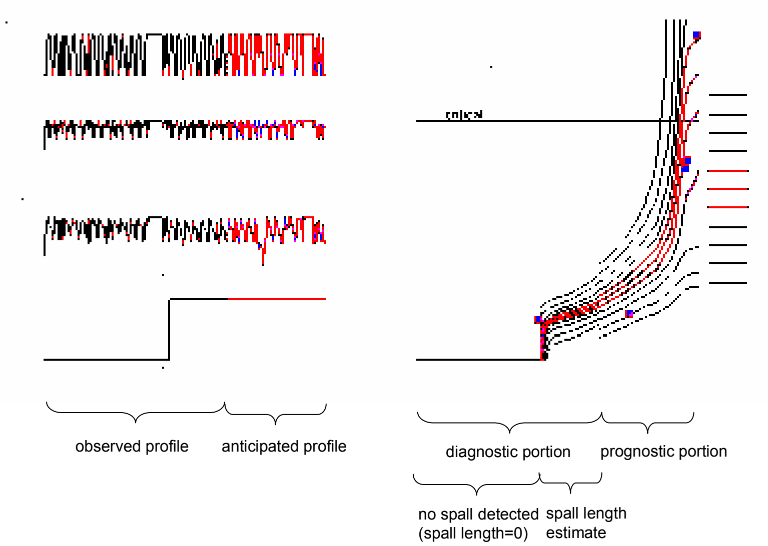
Test Runs
(variable load/speed)



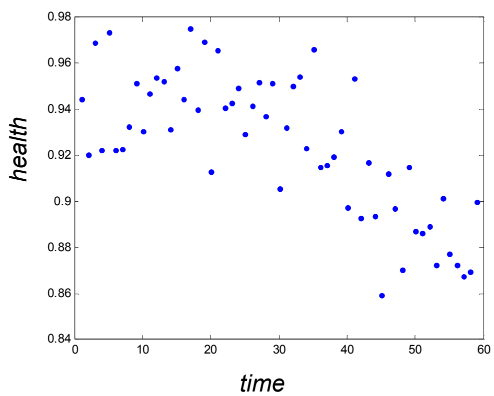
Damage Progression vs. run time hrs



Prediction Results



Non-linear Regression



Nonlinear fit may be appropriate here

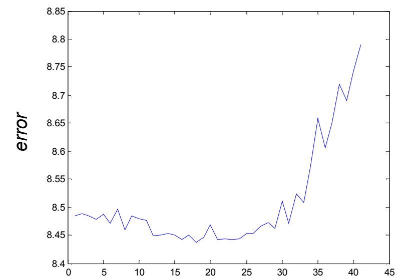
Damage Progression

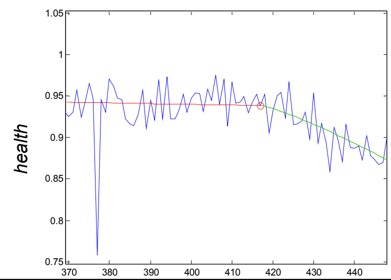
- Many (not all) damage progressions appears to have exponentially growing characteristics
- But: pure data-driven approaches have their limits
  - What is the offset (i.e., what prior damage exists)?
  - What is the damage limit?
  - What is the (exponential) model structure?
  - What are the model parameters?

Non-linear Damage Progression

- Preprocessing:
  - Find elbow point
    - Iterate through suspected elbow region
    - Perform linear regression for left side and exponential regression on right side.
    - Calculate sum of errors
    - Minimum error establishes elbow point
  - Structure ID:
    - Experience shows that this damage progression increases exponentially.

$$y = y_0 + a - a \cdot e^{b \cdot t^c}$$



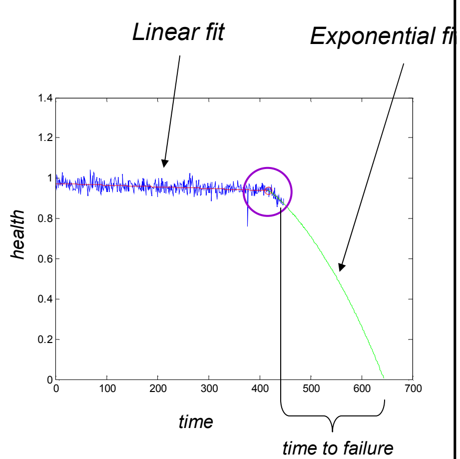


Parameter ID and Prediction

- Exponential regression optimizes over parameters a, b & c
  - Use constrained nonlinear multivariable function optimization

$$\min_{a,b,c} f(a,b,c,t)$$

  - Here
    - y_0 = 0.9384
    - a = 1.1957
    - b = 8.8143e-004
    - c = 1.1953
  - Then extrapolate resulting function to intersection with health=0
- Issues?



Case-Based Reasoning for Forecasting

- Basic Hypothesis:
  - Examples of cases that went to failure have a trail of characteristic features that describes them at each time step plus an associated remaining life.
  - The comparison of a new example to old existing examples allows estimation of remaining life

What is CBR?

- A case-based reasoner solves new problems by using or adapting solutions that were used to solve old problems
- offers a reasoning paradigm that is similar to the way many people routinely solve problems

52

What Is CBR?

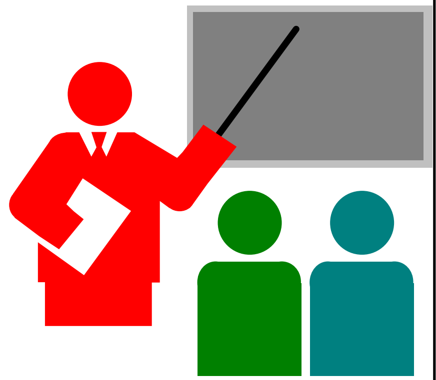
What is 12 x 12?

144

What is 13 x 12?

12 x 12 + 12

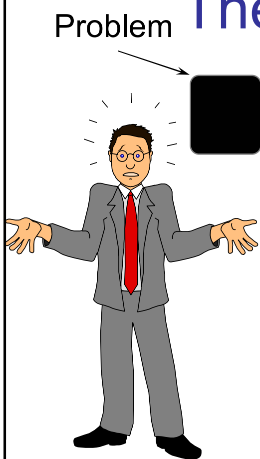
156



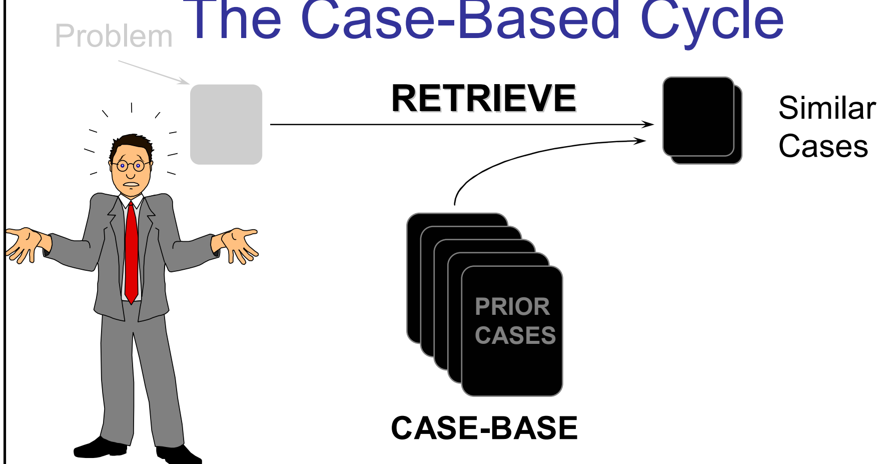
Who uses CBR?

- Lawyers
  - find previous ruling that applies to case
  - show that it applies to current case
- Real Estate Appraiser
  - find similar comparable houses
  - estimate value of target based on value of comparable
- Diagnosticians
  - Collect labeled data of normal data and faults ("cases")
  - Assess whether new data is similar to old cases
- Prognosticators
  - Collect time series of data that went to failure (label is remaining life)
  - Assess whether new data is similar to old case

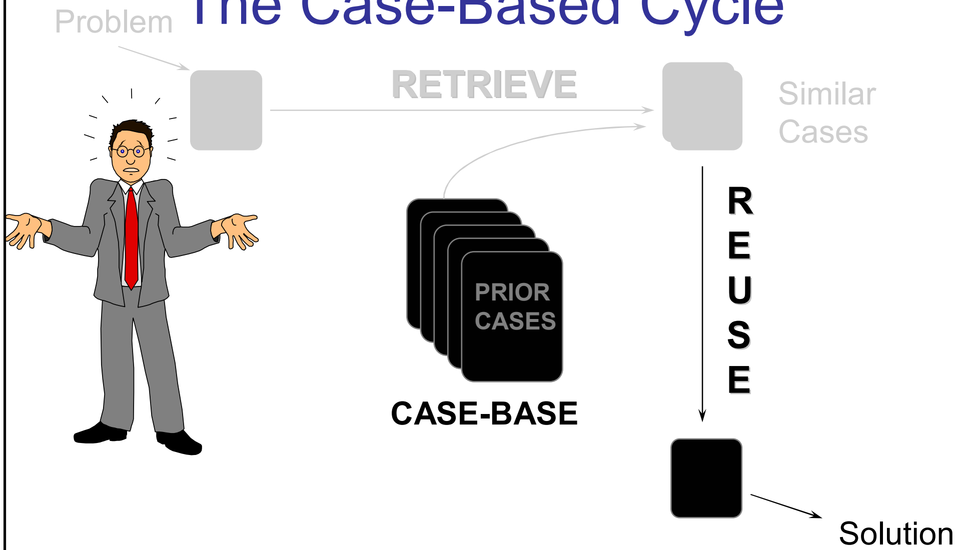
The Case-Based Cycle



The Case-Based Cycle

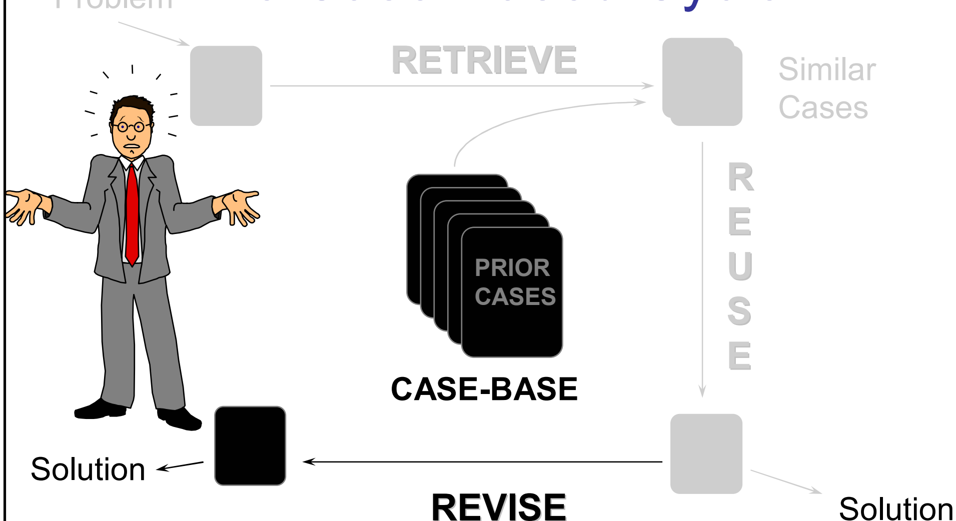


The Case-Based Cycle



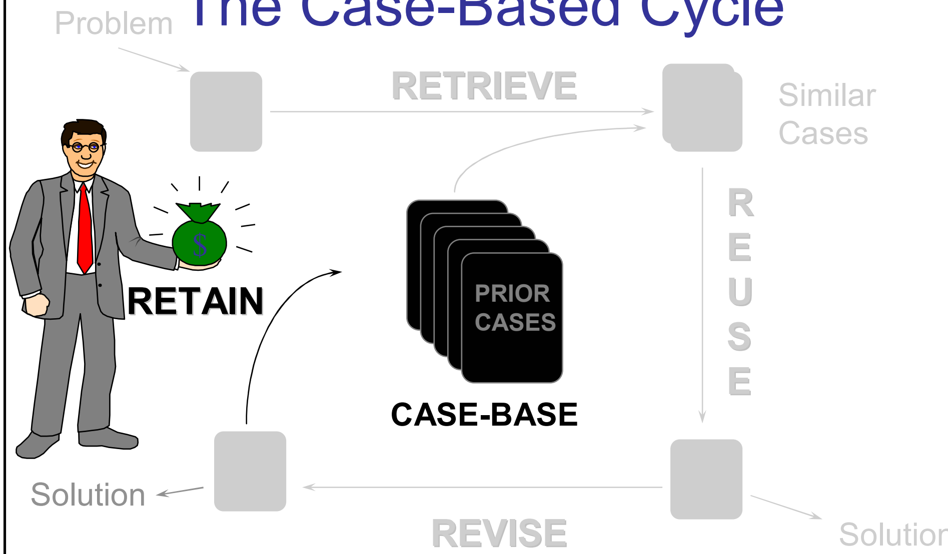
57

The Case-Based Cycle



58

The Case-Based Cycle



59

What is a Case?

- several features describing a problem (or a faulty equipment)
- plus an outcome or a solution (or remaining life)
- cases can be very rich
  - text, numbers, symbols, plans, multimedia
- cases are not usually distilled knowledge
- cases are records of real events
- and are excellent for justifying decisions

60

How Does Retrieval Work?

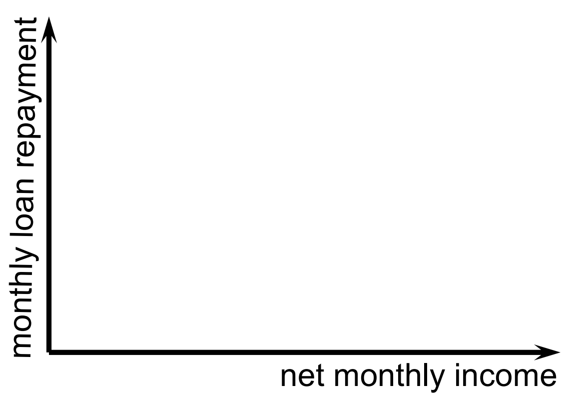
- imagine a decision with two factors that influence it
- should you grant a person a loan?
  - ☆ net monthly income
  - 🕐 monthly loan repayment

  (more factors in reality)

61

How Does CBR Work?

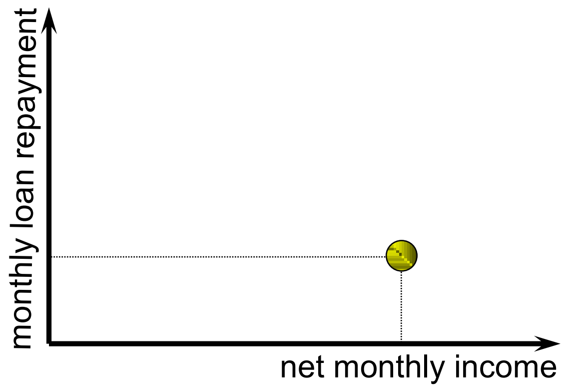
- these factors can be used as axes for a graph



62

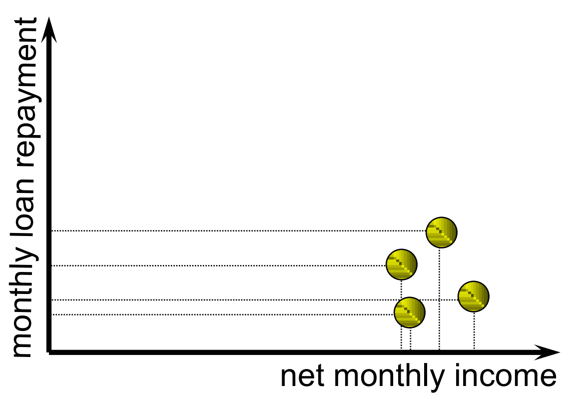
How Does CBR Work?

- a previous loan can be plotted against these axes



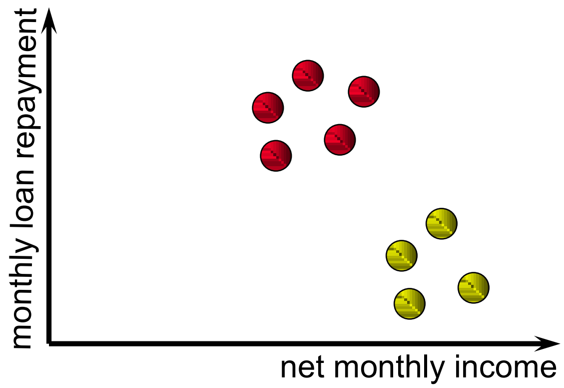
How Does CBR Work?

- and more good loans



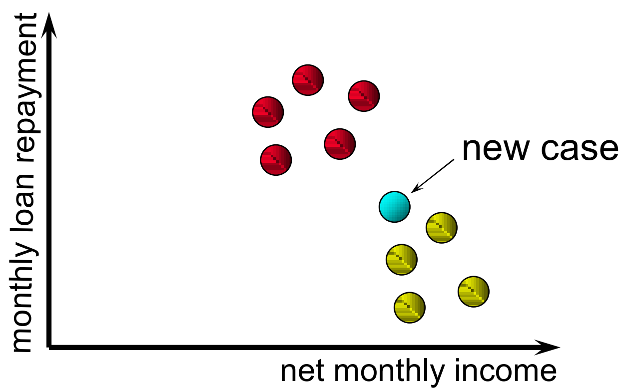
How Does CBR Work?

- plus some bad loans



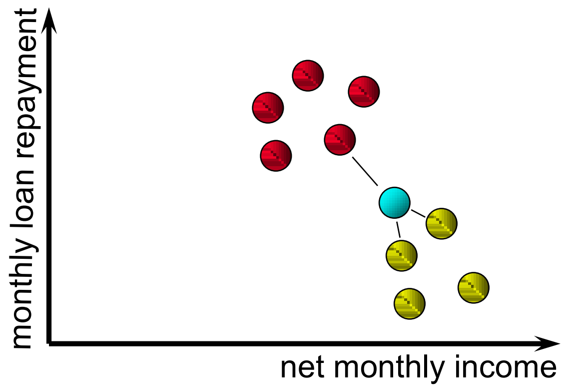
How Does CBR Work?

- a new loan prospect can be plotted on the graph



How Does CBR Work?

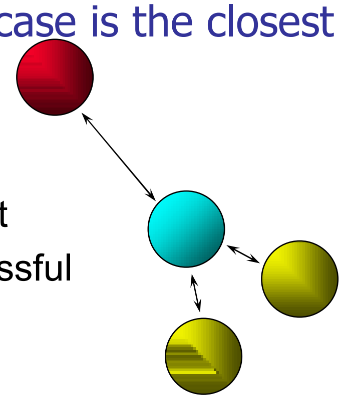
- and the distance to its nearest neighbors calculated



67

How Does CBR Work?

- the best matching past case is the closest



- this suggests a precedent
- the loan should be successful

68

knn Algorithm

Looks for k nearest neighbors (knn) to classify data

Assigns class based on majority among knn

- Compute distance from data point to labeled samples
- If knn have not been found yet then include data point
- Else, if a labeled sample is closer to the data point than any other knn then replace the farthest with the new one
- Deal with ties
- Repeat for the next labeled sample
- When done, perform majority vote on collected cases to determine class assignment

Fuzzy knn Algorithm

- Compute distance from data point to labeled samples
- If knn have not been found yet then include data point
- Else, if a labeled sample is closer to the data point than any other knn then replace the farthest with the new one
- Compute membership

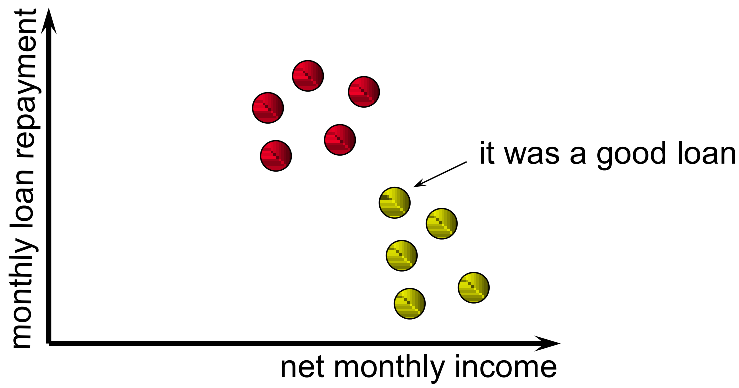
  (inverse of distances from nn and their class memberships)

$$u_i(x) = \frac{\sum_{j=1}^{k} u_{ij} \left(\frac{1}{\|x - x_j\|^{\frac{2}{m-1}}} \right)}{\sum_{j=1}^{k} \left(\frac{1}{\|x - x_j\|^{\frac{2}{m-1}}} \right)}$$

- Repeat for the next labeled sample

How Does CBR Work?

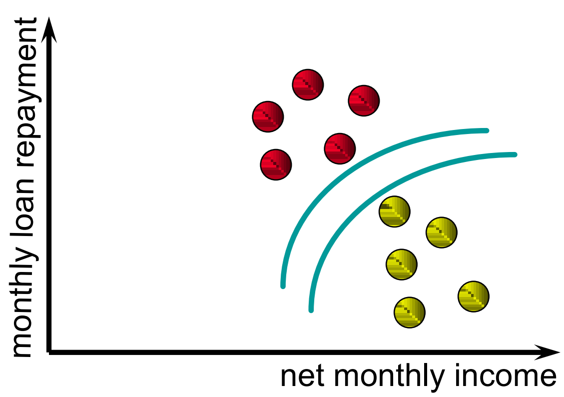
- over time the prediction can be validated



71

How Does CBR Work?

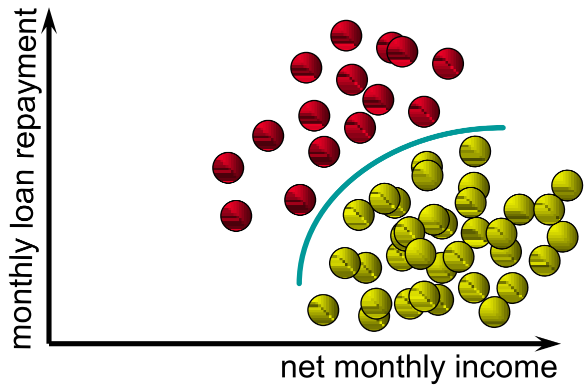
- the system is learning to differentiate good and bad loans better



72

How Does CBR Work?

- as more cases are acquired its performance improves



73

Retrieval Issues

- Do all indexed features have the same weight?
- Is similarity linearly proportional to distance between case and new problem?
- What distance measure should be used (city block, line of sight, …)
- Uniformity of solution space

74

When to Apply CBR?

- when a domain model is difficult or impossible to elicit
- when the system will require constant maintenance
- when records of previously successful solutions exist

75

Case-Base Issues

- Number of cases needed
- Locally dense areas in CB vs. sparse areas
- Removing overlapping cases
- How to efficiently search
  - create abstractions from cases
  - multiple case bases
- Features used for indexing
- Weighting the features
- How to deal with evolving systems

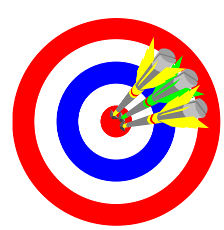
76

Disadvantages of CBR

- Can take large processing time to find similar cases in case-base
- Can take large storage space for all the cases
- Cases may need to be created by hand
- Adaptation may be difficult

CBR Tradeoff

- if you require the *best* solution or the *optimum* solution - CBR may not be for you
- CBR systems generally give *good* or *reasonable* solutions
- this is because the retrieved case often requires adaptation



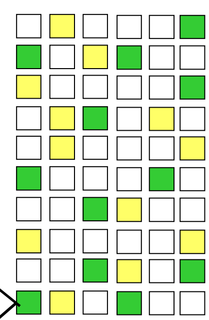
Application to Locomotive Failure

- X = Age of locomotive, # of recommendations, miles/day, avg. megawatt hours, and many more.
- Y = Time to failure for that locomotive.
- We analyzed the performance of the evolutionary approach over two years of operation and maintenance data for a fleet of 1100 locomotives.
- The evolutionary algorithm automatically picked the best combination of variables from the X space that best allowed us to predict the time to failure.
- With passage of time, the genetic algorithm also automated the task of keeping the model trained without human intervention.

X = Asset History

Y = Future Reliability

Loco Number:	*5700*
Design and Configuration	
Type:	*AC4400*
Electrical System:	*Bosch*
…	
Utilization Information	
Age:	*2.9 years*
Mileage:	*247,567 mi.*
Average miles/day:	*299*
Maintenance Information	
Time elapsed since last repair:	*10 days*
Median time between repairs	*60 days*
Median time from repair to next recommendation (Rx)	*52 days*



Top 20% reliability

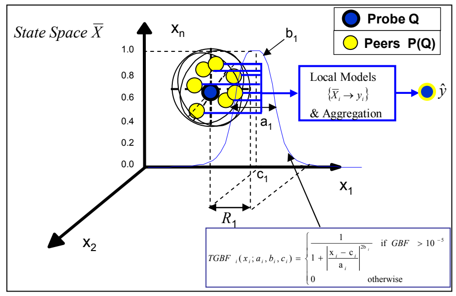
Bottom 20% reliability

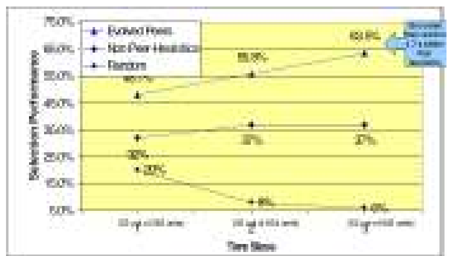
We focus on selection of top 20%

Bonissone, P.P.; Varma, A.; Aggour, K.; A fuzzy instance-based model for predicting expected life: a locomotive application. IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2005. CIMSA. 2005 20-22 July 2005. pp.:20 - 25

Reasoning process

Fuzzy Case-Based Model (FCBM)





1) Retrieval of similar cases from the Data Base

2) Evaluation of similarity measure between the probe and the retrieved cases

3) Local reasoning models give a decision (Y) for each of the cases

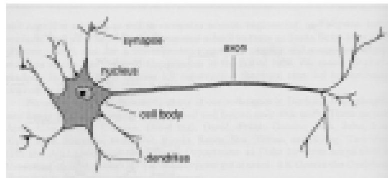
4) Outputs Aggregation (weighted by their similarities) as the final decision (Y) for the probe Q

Bonissone, P.P.; et al., 2005

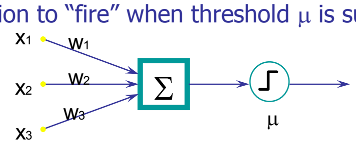
Note:
This process is actually more correctly referred to as "Instance-Based Modeling"

Neural Networks for Prognostics

Neurons



- McCulloch & Pitts (1943)
  - simple model of neuron as a binary threshold unit
  - uses step function to “fire” when threshold μ is surpassed



  - Real Neurons:
    - use not even approximately threshold devices
    - probably use a non-linear aggregation method
    - produce a sequence of pulses (not a single output level)
    - do not have the same fixed delay (t-> t+1)
    - are not updated synchronously
    - amount of transmitter substance varies unpredictably
  - But: Artificial Neurons can do useful things (e.g., for pattern recognition)

Neural Nets: Categorization

- Supervised Learning
  - Multilayer perceptrons
  - Radial basis function networks
  - Modular neural networks
  - LVQ (Learning Vector Quantization)
- Reinforcement Learning
  - Temporal Difference Learning
  - Q-Learning
- Unsupervised Learning
  - Competitive learning networks
  - Kohonen self-organizing networks
  - ART (adaptive resonant theory)

Supervised Neural Networks

- Requirement:
  - known input-output relations

Perceptrons

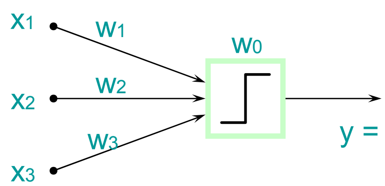
- Rosenblatt: 1950s
- Input patterns represented is binary
- Single layer network can be trained easily
- Output o is computed by

$$o = f\left(\sum_{i=1}^{n} w_i x_i - w_0\right)$$

- where
  - w_i is a (modifiable) weight
  - x_i is the input signal
  - w_0 is some threshold (weight of constant input)
  - $f(\cdot)$ is the activation function $f(x) = \mathrm{sgn}(x) = \begin{cases} 1 \text{ if x} > 0 \\ 0 \text{ otherwise} \end{cases}$

Single-Layer Perceptrons

- Network architecture

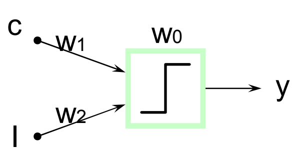


$y = \text{signum}(\Sigma w_i x_i + w_0)$

Single-Layer Perceptron

Example: Fish classification

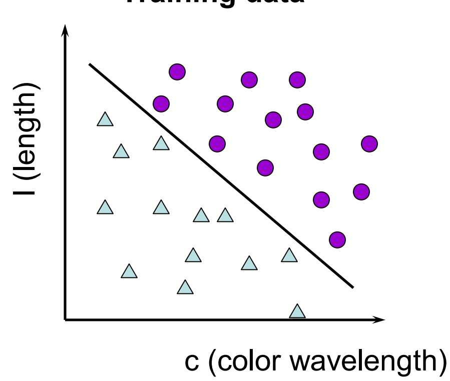
Network Arch.



$y = \text{signum}(cw_1 + lw_2 + w_0)$

$= \begin{cases} -1 & \text{if Bass} \\ 1 & \text{if Salmon} \end{cases}$

Training data



Perceptron Learning

- Learning:
  - select an input vector
  - if the response is incorrect, modify all weights

$$\Delta w_i = \eta\, t_i x_i$$

where

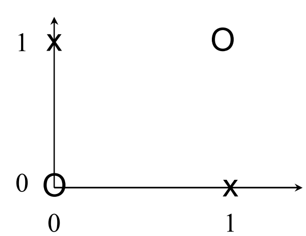
t_i is a target output

η is the learning rate

XOR

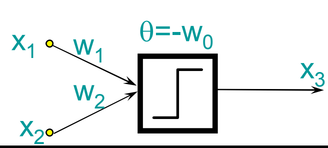
- Minsky and Papert reported a severe shortcoming of single layer perceptrons, the XOR problem…

x1	x2	output
0	0	0
0	1	1
1	0	1
1	1	0



not linearly separable

$$0w_1 + 0w_2 + w_0 \le 0 \Leftrightarrow w_0 \le 0$$
$$0w_1 + 1w_2 + w_0 > 0 \Leftrightarrow w_2 > -w_0$$
$$1w_1 + 0w_2 + w_0 > 0 \Leftrightarrow w_1 > -w_0$$
$$1w_1 + 1w_2 + w_0 \le 0 \Leftrightarrow w_1 + w_2 \le -w_0$$

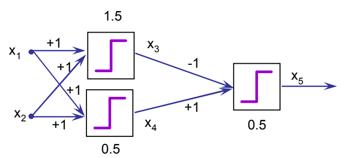


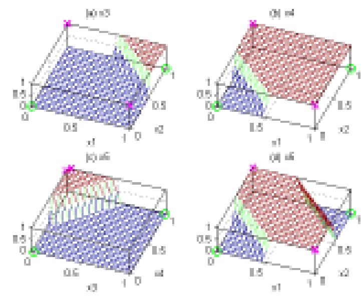
Enter the Dark Ages of NNs

- …which (together with a lack or proper training techniques for multi-layer perceptrons) all but killed interest in neural nets in the 70s and early 80s.

Two-Layer Perceptron: XOR

- Multi-layer architectures can deal with XOR
  - Determining weights is painful





- Idea
  - find first derivative of the error wrt the weights,
  - then move the weights a small amount towards a smaller error
  - Can one take the first derivative for the architecture above?
- what alternative learning method could one employ?

Multi-Layer Perceptrons

- Recall the output

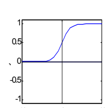
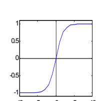
$$o = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right)$$

- and the squared error measure

$E_p = \left(t_p - o_p\right)^2$ which is amended to $\quad {}^pE_k = \sum_{k=1}^{n}\left({}^pt_k - {}^po_k\right)^2$

- and the activation function

$f(x) = \dfrac{1}{1+e^{-x}}$ or $f(x) = \dfrac{1-e^{-x}}{1+e^{-x}} = \tanh\left(\dfrac{x}{2}\right)$ or $f(x) = x$

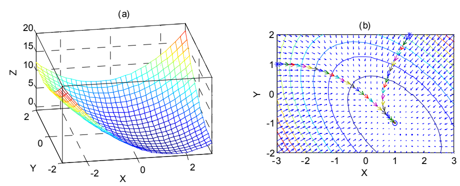
 "squashing functions" 

- then the learning rule for each node can be derived using the chain rule...

Backpropagation

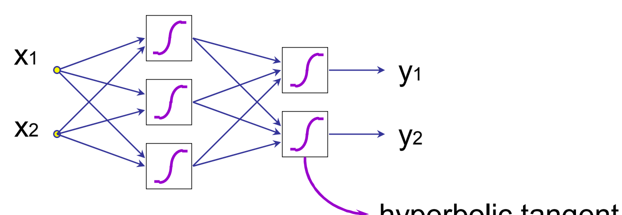
- make incremental change in the direction dE/dw to decrease the error. $\frac{\partial E}{\partial parameters}$
- The learning rule for each node can be derived using the chain rule...
- …to propagate the error back through a multi-layer perceptron.

$$\Delta w_{ki} = -\eta \sum_{p} \frac{\partial E_p}{\partial w_{ki}}$$



Multilayer Perceptrons (MLPs)

Network architecture



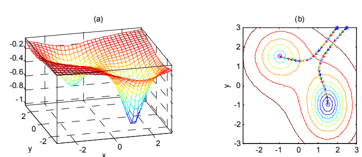
Learning rule:

- **Steepest descent (Backprop)**
- **Conjugate gradient method**
- **All optim. methods using first derivative**
- **Derivative-free optimization**

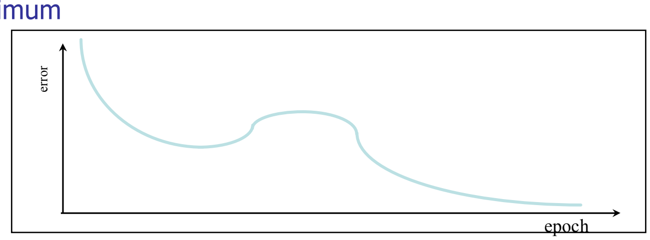
Backprop Procedure

1. Initialize weights to small random values
2. Choose a pattern and apply it to input layer
3. Propagate the signal forward through the network
4. Compute the deltas for the output layer
5. Compute the deltas for the preceding layers by propagating the error backwards
6. Update all weights
7. Go back to step 2 and repeat for next pattern
8. Repeat until error rate is acceptable

Local Minima

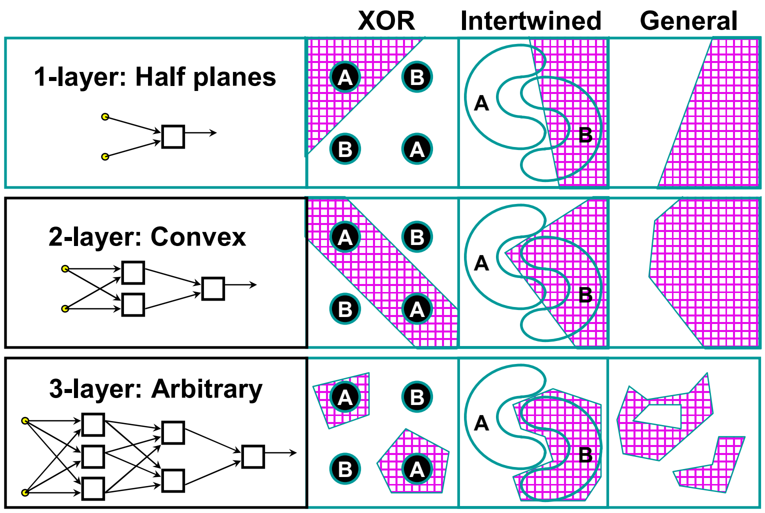


- There is no guarantee that the algorithm converges to a global minimum



- check with different initial conditions (different weights, etc.)
- perturb the system (data) with noise to improve result

MLP Decision Boundaries



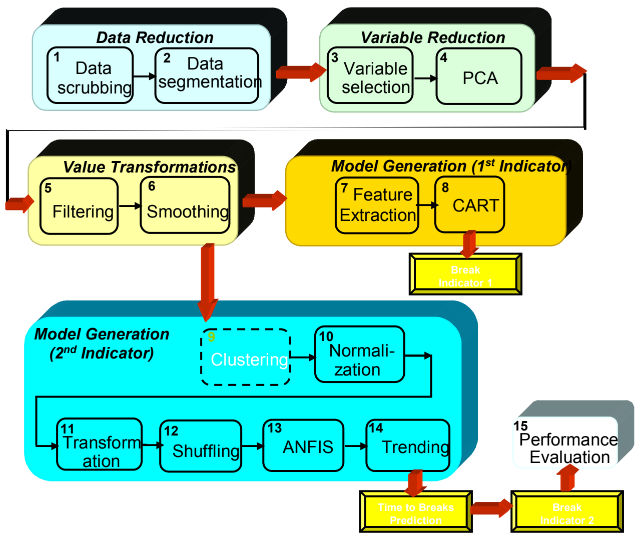
11/29/2007

99

Prognostics Example

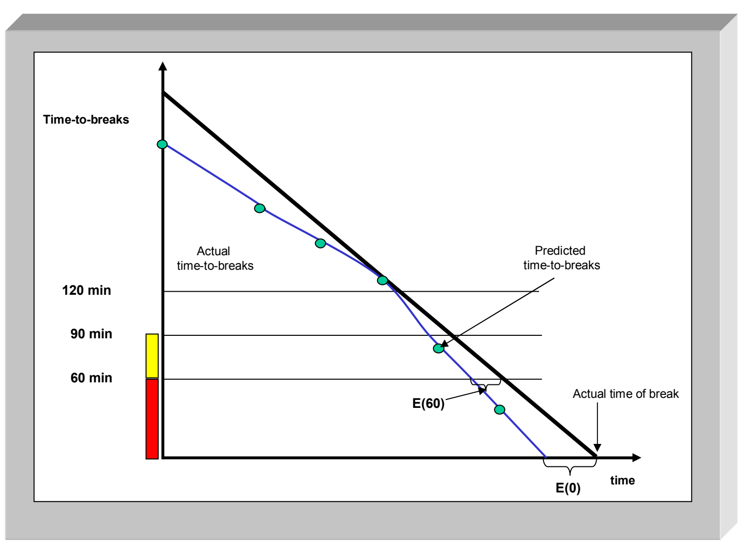
- Problem:
  - Paper Making Machine standstill very costly
  - If the user knew about an impending breakage, control parameters could be changed to avert the breakage
  - many indicators
  - poor indicators
  - high noise
- Solution
  - Prognostics system using a combination of techniques
    - data scrubbing
    - variable reduction
    - variable transformation
    - model generation
    - trending
    - performance evaluation

Breakage Prediction Process

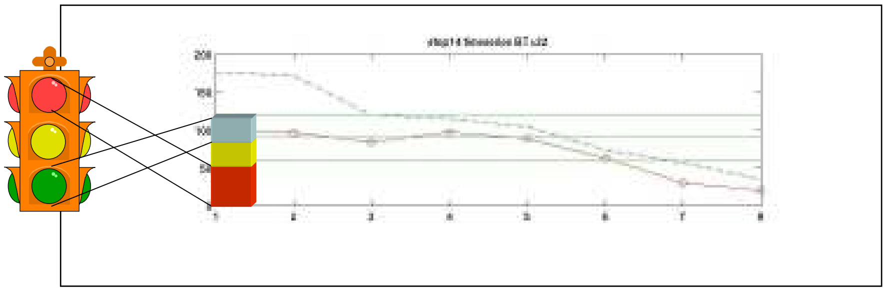


US5,942,689: System and method for predicting a web break in a paper machine, Bonissone, P., Chen, Y., Khedkar, P., 1999.
Pat. Pend.: Method for Predicting Time to Break Wet-End in Paper Mills Using Principal Component Analysis and Classification Trees", P. Bonissone, Y. Chen, filed 9/15/1999.
Pat. Pend.: Method for Predicting Time to Break Wet-End in Paper Mills Using Principal Component Analysis and Neuro Fuzzy Systems and Trending Analysis", P. Bonissone, Y. Chen, filed 9/15/1999.
Pat. Pend.: System and method for improving Decision Trees by Bagging for Wet-End Web Breakage Prediction in Paper Mills", P. Bonissone, Y. Chen, 1999.

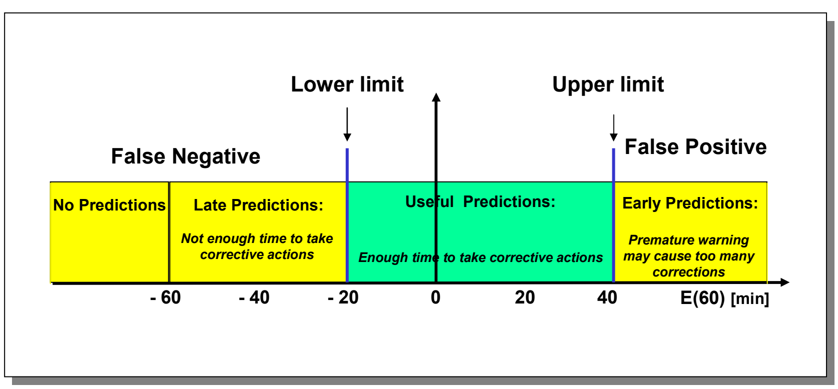
Graphical Description of the Prediction Error at time=60, i.e., E(60)



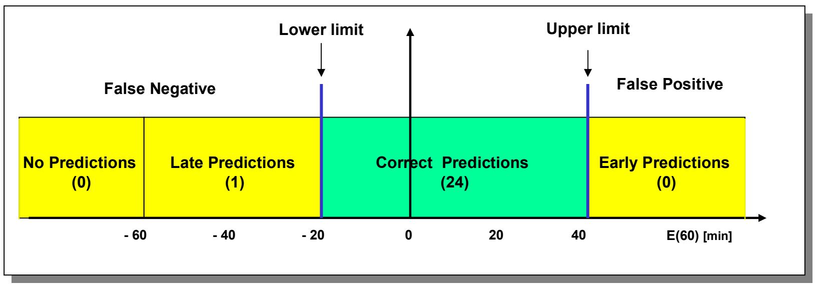
Trending Analysis



Definition of Useful Prediction: Limits for E(60)



Summary of E(60) Analysis



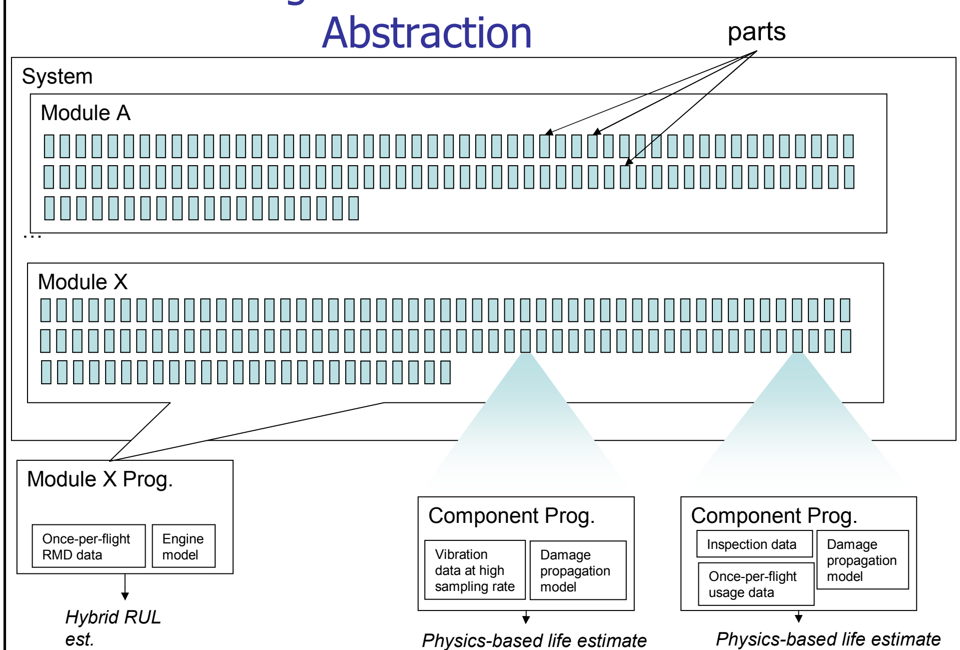
Hybrid Approaches

- Pre-Estimate Fusion of Model and Data
- e.g.,:
  - Using thermodynamic engine model and on-wing data for turbine damage propagation calculations
  - Using battery model and impedance measurements for remaining life calculations
- Post-Estimate Fusion of Prognostic Estimates
- e.g.,:
  - Fuse model-based model output and data-driven model output to reduce prediction uncertainty for bearing spall prognostics

Pre-Estimate Hybrid Fusion

- Problem constraints
  - Damage propagation model not available/too expensive
  - Failure data not available
  - Engine thermo-dynamic model available
  - In-flight snapshot data available
- Challenge
  - Can one predict end-of-life of modules within engine due to component level faults?

Prognostics: Levels of Abstraction

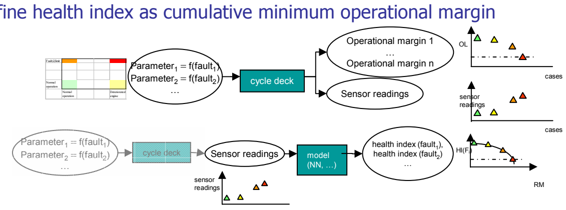


Hybrid Prediction

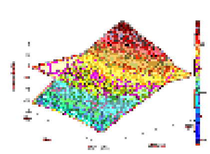
- What is hybrid here
  - Use thermo-dynamic model to characterize module health
    - Challenge: what is end of life
    - Use operational margins as clue
  - Use sensor data and feed through model to get current health state
  - Then use updated nonlinear extrapolation to propagate damage and calculate time remaining
- Byproduct
  - No detailed materials knowledge required
  - Allows segregation of worsening faults vs. stable faults
  - Health estimate of component
  - Deterioration estimate
  - Root cause information

Component-level Prognostics: Approach

1. Off-line: Learn impact of faults
   - Model Faults with different magnitude in cycle deck
   - Understand impact of faults
     - Expected sensor output
     - Changes of performance characteristics
   - Define health index as cumulative minimum operational margin



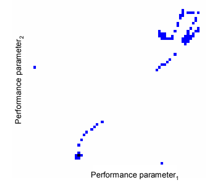
   - Generate health index map for different components

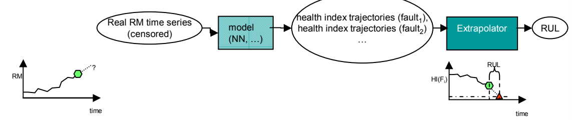


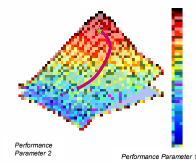
Component-level Prognostics Approach

2. Online: Evolve component health

- Estimate deterioration
- Find performance parameters that optimally match sensor data
- Calculate health index
- Propagate health index forward by evolving fault through non-linear regression

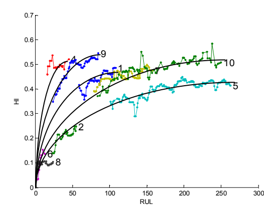




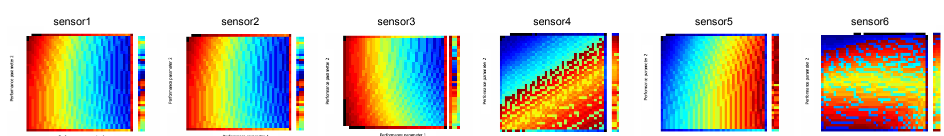


- RUL is intersection of health index at risk level cut-off

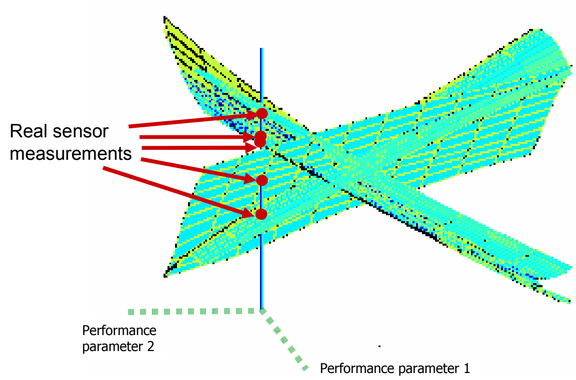
$$RUL = \hat{T}\left(HI = 0 \middle| \sigma = \sigma_{acceptable}\right) - T\left(now \middle| T_{now} \geq \hat{T}_{fault}\right)$$



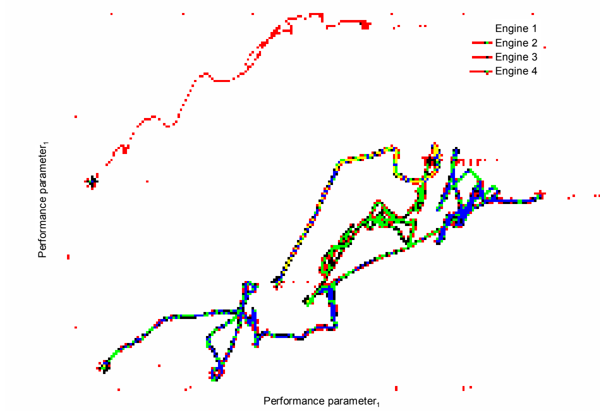
State Assessment



- Generate response surfaces for module faults
- Normalize surfaces
- Find performance parameter pair that best explains sensor data **min(Σwi*(Dist_i)^2), i Є {sensor1, sensor2, sensor3, sensor4, sensor5, sensor6}**



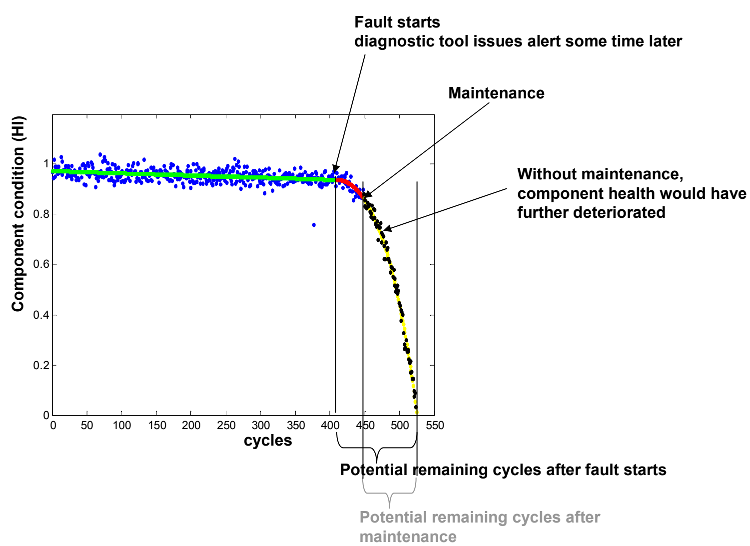
Real Example – Actuator failures



- This fault mode has consistent direction in performance parameter space
  - important diagnostic evidence for the fault mode
- Other fault modes are manifested by trajectories with different slope
- Rate of change is important prognostic information
- No additional specific materials/geometry/thermodynamics knowledge is used

Example Case (Real Data)

- Tracking component condition (health)



Hybrid Approaches

- Pre-Estimate Fusion of Model and Data
- e.g.,:
  - Using thermodynamic engine model and on-wing data for turbine damage propagation calculations
  - Using battery model and impedance measurements for remaining life calculations
- Post-Estimate Fusion of Prognostic Estimates
- e.g.,:
  - Fuse model-based model output and data-driven model output to reduce prediction uncertainty for bearing spall prognostics

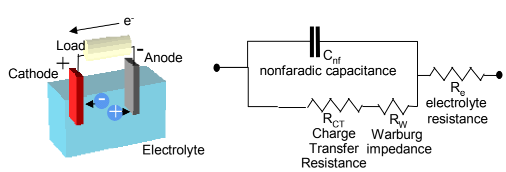
Pre-Estimate Hybrid Fusion

- Problem constraints
  - Lack of in-depth understanding of electro-chemical system
  - Low-fidelity model
  - Long-term aging data available
- Challenge
  - Can one predict end-of-life of system
    - Conditions are different than those seen in prior experience
    - Significant uncertainties in model, sensor data, operating conditions
    - Quantify uncertainties

Hybrid Prediction

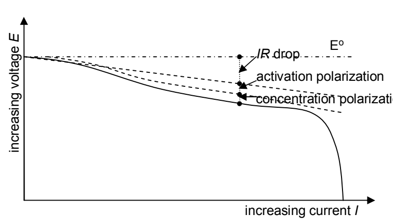
- What is hybrid here
  - Use lumped parameter model to characterize battery health
  - Use sensor data and feed through model to get current health state
  - Use particle filter approach integrating future expected conditions to propagate damage and calculate time remaining
- Byproduct
  - Uncertainty estimates
  - No detailed electro-chemical knowledge required

Battery Basics

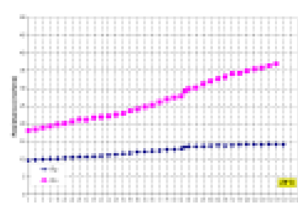


- Impedance Z consists of:
  - resistance R, and
  - reactance X
- (Z=R+*j*X)

Lumped parameter model used to predict response



Polarization Curve



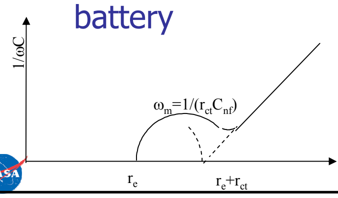
Internal resistance increases as battery decays

Measurement Scheme

Electrochemical Impedance Spectroscopy (EIS)

- Requires oscillator to carry out frequency sweep
- Plot capacitive vs. resistive component of cell

  - Interdependence of the components yields a semicircle
  - Linear portion of curve corresponds to diffusion given by Warburg impedance
- Response is different in presence of passivation and corrosion, providing a diagnostic for the health state of battery



Electrolyte weakening

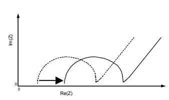
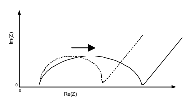
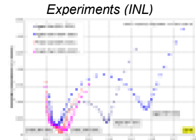


Plate Sulfation



Experiments (INL)

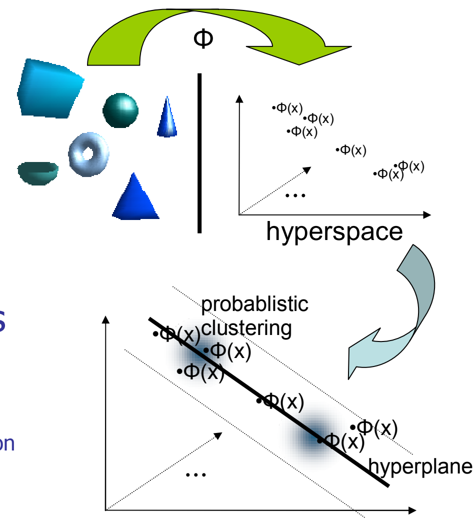


Bayesian Approach

- Relevance Vector Machine
  - State of the art in nonlinear probabilistic regression
  - Faster than SVM
- Particle Filter
  - State of the art for nonlinear non-Gaussian state estimation
  - Slower than Kalman Filter
  - Uses model

Relevance Vector Machine

- Detect and exploit complex patterns in data
  - represent complex patterns
  - exclude spurious patterns (overfitting)
- Implementation steps
  - embed data into higher dimensional space using kernels
  - cluster data in probabilistic fashion
  - detect linear relations in hyperspace (hyperplane)



Basic Idea

- Two Separate Modules:

Kernel Function → takes care of the embedding

Learning Module → performs the learning in the embedding space

Kernels

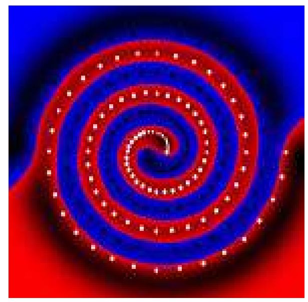
Kernels: A function that returns the value of the dot product between the images of the two arguments

$$K(x1,x2)= \langle \varphi(x1),\varphi(x2) \rangle$$

Simple examples of kernels:

$$K(x,z) = \langle x,z \rangle^d$$

$$K(x,z) = e^{-||x-z||^2/2\sigma}$$



Source: M. J. Tipping

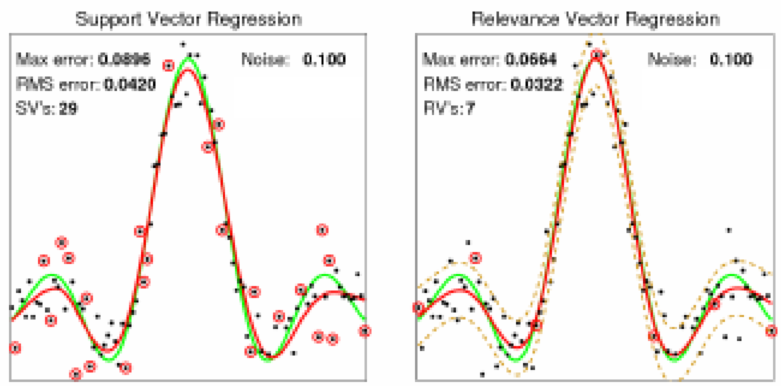
RVM: An Extension of SVM

- RVM is a sparse Bayesian model using the same kernel basis as SVM

$$y(\mathbf{x};\mathbf{w}) = \sum_{n=1}^{N} w_n K(\mathbf{x},\mathbf{x}_n) + w_0$$

- Advantages
  - Nuisance parameters can be integrated out
  - Posterior probabilities generated
  - Not limited to Mercer kernels

Comparative Performance



Source: M. J. Tipping

Comparative Performance

Regression

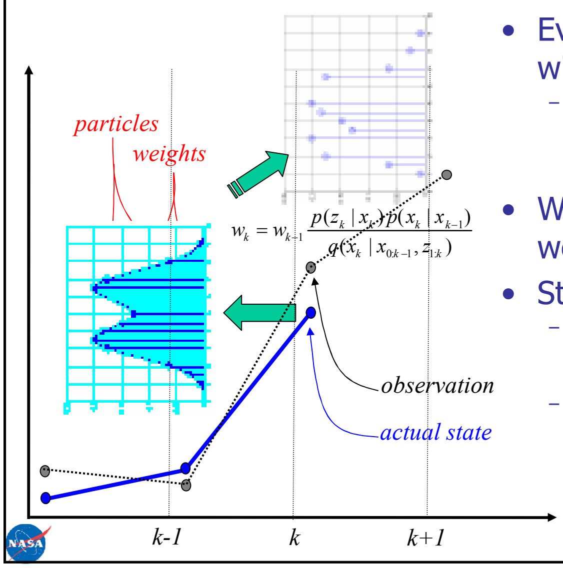
			errors		vectors	
Data set	N	d	SVM	RVM	SVM	RVM
Sinc (Gaussian noise)	100	1	0.378	0.326	45.2	6.7
Sinc (Uniform noise)	100	1	0.215	0.187	44.3	7.0
Friedman #1	240	10	2.92	2.80	116.6	59.4
Friedman #2	240	4	4140	3505	110.3	6.9
Friedman #3	240	4	0.0202	0.0164	106.5	11.5
Boston Housing	481	13	8.04	7.46	142.8	39.0
Normalized Mean			1.00	0.86	1.00	0.15

Classification

			errors		vectors	
Data set	N	d	SVM	RVM	SVM	RVM
Pima Diabetes	200	8	20.1%	19.6%	109	4
U.S.P.S.	7291	256	4.4%	5.1%	2540	316
Banana	400	2	10.9%	10.8%	135.2	11.4
Breast Cancer	200	9	26.9%	29.9%	116.7	6.3
Titanic	150	3	22.1%	23.0%	93.7	65.3
Waveform	400	21	10.3%	10.9%	146.4	14.6
German	700	20	22.6%	22.2%	411.2	12.5
Image	1300	18	3.0%	3.9%	166.6	34.6
Normalized Mean			1.00	1.08	1.00	0.17

Source: M. J. Tipping

Particle Filters



- Every particle is associated with a weight
  - Particles, together with their weights, represent a sampled version of the PDF.

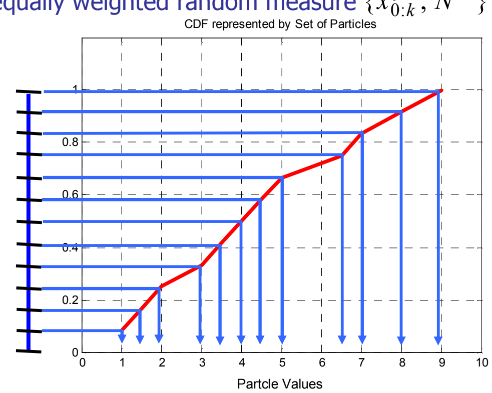
    $S_k = \{\langle \mathbf{x}_k^{(i)}, w_k^{(i)} \rangle \mid i = 1, \ldots, n\}$
- We study the propagation of weights in time
- Steps:
  - Predict the "a priori" PDF parameters, using the model

    $p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}$
  - Update parameters, given the new observation

    $p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})}$

The Particle Filter Framework

- Sequential Importance Sampling (SIS) to avoid degeneracy of weights
  - Maps the previously weighted random measure $\{x_{0:k}^i, \tilde{w}_k(x_{0:k}^i)\}$ onto a new equally weighted random measure $\{x_{0:k}^j, N^{-1}\}$



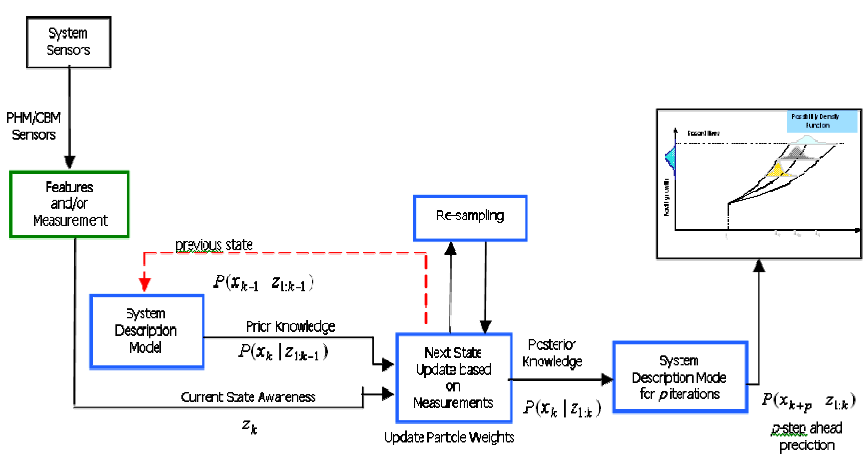
Prognosis

- P-step ahead prediction defined by using the model update procedure in a recursive manner

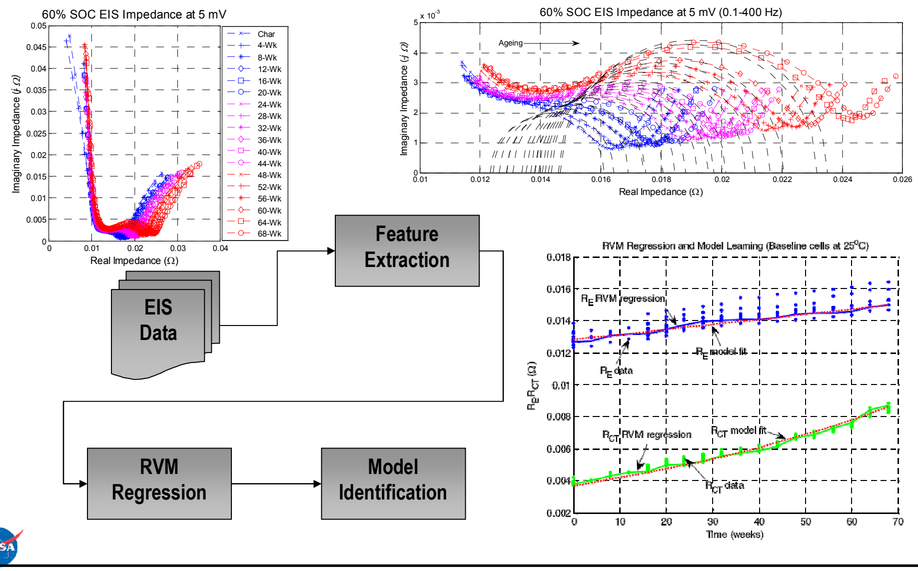
$$p(x_{k+p} \mid z_{0:k}) = \int p(x_k \mid z_{0:k}) \prod_{j=k+1}^{k+p} p(x_j \mid x_{j-1}) dx_k$$

$$= \sum_{i=1}^{N} \tilde{w}_k^{(i)} p(x_{k+1} \mid x_k^{(i)}) \prod_{j=k+2}^{k+p} p(x_j \mid x_{j-1})$$

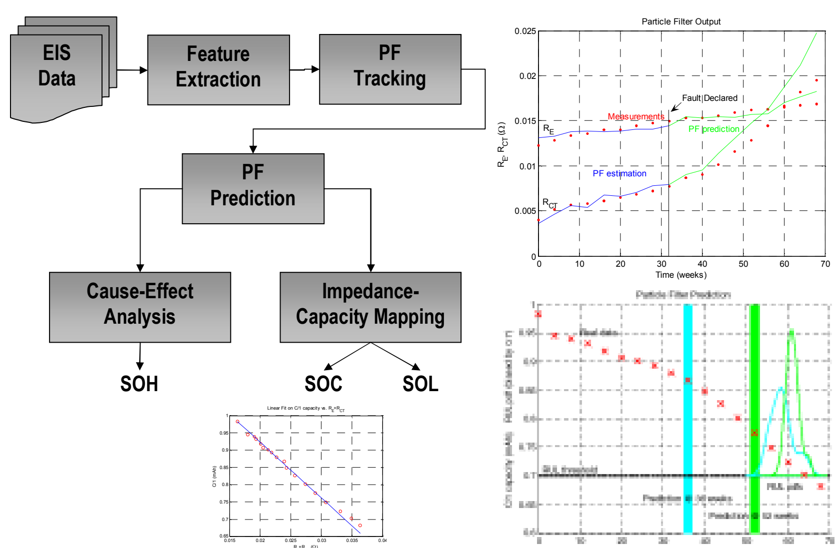
The Particle Filter Framework



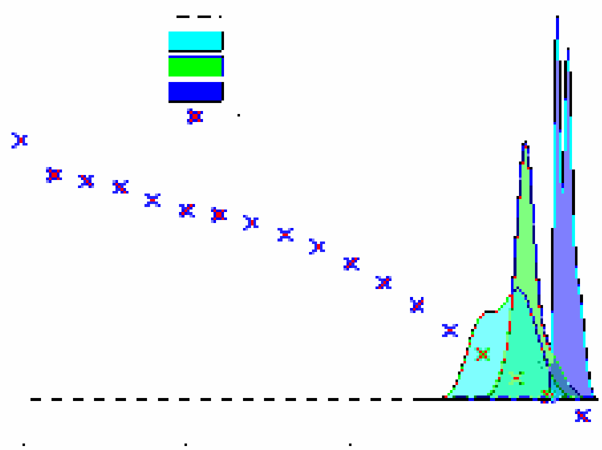
Model Development



Prognosis Framework



Remaining Useful Life



Summary and Final Remarks

- Choice of prognostic technique must be based on constraints
  - Model-based: need understanding of damage propagation
  - Data-driven: need data time series to failure and failure criterion
- Techniques discussed today
  - Examples for physics-based models
  - Particle filters
  - Data-driven techniques
    - Regression (linear, nonlinear)
    - Case-based reasoning
    - Neural networks
    - RVM
  - Hybrid techniques
    - Pre-estimate fusion
- Issues requiring more research
  - Perception and state estimation
    - Sensors are still the weak link
  - Learning and adaptive systems
    - Space is the "final frontier" for ISHM
  - Software complexity
    - V&V, certification
  - Life estimation and physics of failure
    - Prognostics is still in its infancy
      - Model building
      - Uncertainty management
      - Performance assessment

References

K. Goebel and N. Eklund	Prognostic Fusion for Uncertainty Reduction	Proceedings of Infotech @ AIAA		2007
F. Xue, K. Goebel, P. Bonissone, W. Yan	An Instance-Based Method for Remaining Useful Life Estimation for Aircraft Engines	Proceedings of MFPT 2007		2007
H. Qiu, N. Eklund, W. Yan, P. Bonissone, F. Xue, K. Goebel	Estimating deterioration level of aircraft engines	Proceedings of ASME Turbo Expo 2007		2007
X. Hu, N. Eklund, K. Goebel, and W. Cheetham	Hybrid Change Detection for Aircraft Engine Fault Diagnostics	Proceedings of 2007 IEEE Aerospace Conference		2007
K. Goebel, H. Qiu, N. Eklund, W. Yan	Modeling Propagation of Gas Path Damage	Proceedings of 2007 IEEE Aerospace Conference		2007
P. Bonissone, K. Goebel, I. Naresh	Knowledge and Time: Selected Case Studies in Prognostics and Health Management (PHM)	Proceedings of IPMU '06		2006
N. Eklund and K. Goebel	Using Meta-Features to Boost the Performance of Classifier Fusion Schemes for Time Series Data	Proceedings of International Joint Conference on Neural Networks, 2006. IJCNN '06.	pp. 3223 - 3230	2006
N. Iyer, K. Goebel, and P. Bonissone	Framework for Post-Prognostic Decision Support	Proceedings of 2006 IEEE Aerospace Conference	11.0903	2006
K. Goebel, N. Eklund, and P. Bonanni	Fusing Competing Prediction Algorithms for Prognostics	Proceedings of 2006 IEEE Aerospace Conference	11.1004	2006
N. Eklund and K. Goebel	Using Neural Networks and the Rank Permutation Transformation to Detect Abnormal Conditions in Aircraft Engines	Proceedings of the 2005 IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications, SMCia/05	pp. 1-5	2005
K. Goebel and P. Bonissone	Prognostic Information Fusion for Constant Load Systems	Proceedings of the 7th Annual Conference on Information Fusion, Fusion 2005, Vol. 2.	pp. 1247 - 1255	2005
K. Goebel, P. Bonanni, N. Eklund	Towards an Integrated Reasoner for Bearings Prognostics	Proceedings of 2005 IEEE Aerospace Conference	pp. 1 - 11	2005
W. Yan, K. Goebel, and C. J. Li	Flight Regime Mapping for Aircraft Engine Fault Diagnosis	Proceedings of the 58th, Meeting of the Society of Mechanical Failures Prevention Technology, Virginia Beach, VA, April 26-30, 2004, Eds.: H. C. Pusey, S. C. Pusey and W. R. Hobbs, MFPT, Winchester, VA	pp. 153-164	2004
K. Goebel, N. Eklund, and B. Brunell	Rapid Detection of Faults for Safety Critical Aircraft Operation	2004 IEEE Aerospace Conference Proceedings, vol. 5	pp. 3372 - 3383	2004
M. Krok and K. Goebel	Prognostics for advanced compressor health monitoring	Proceedings of SPIE, System Diagnosis and Prognosis: Security and Condition Monitoring Issues III, Vol. #5107	pp.1-12	2003
P. Bonissone and K. Goebel	When will it break? A Hybrid Soft Computing Model to Predict Time-to-break Margins in Paper Machines	Proceedings of SPIE 47th Annual Meeting, International Symposium on Optical Science and Technology, Vol. #4787	pp. 53-64	2002
P. Bonissone, K. Goebel, and Y. Chen	Predicting Wet-End Web Breakage in Paper Mills	Working Notes of the 2002 AAAI symposium: Information Refinement and Revision for Decision Making: Modeling for Diagnostics, Prognostics, and Prediction, Technical Report SS-02-03, AAAI Press, Menlo Park, CA	pp. 84-92	2002

- last slide

Internship Opportunities

- Perform research in prognostics at NASA Ames for a period from 3-12 months
- Contact kai.goebel@nasa.gov