When Will it Break?

Prognostics and Health Management at NASA

Kai Goebel, Ph.D.
kai.goebel@nasa.gov
Prognostics Center of Excellence
Discovery and Systems Health (DaSH)
Intelligent Systems Division
NASA Ames Research Center, California USA

October 26, 2016
FPNI

Acknowledgement:
The slides represent work done by PCoE members who have contributed to this presentation. They include Abhinav Saxena, Matt Daigle, Edward Balaban, George Gorospe, Chris Teubert, Sriram Narasimhan, Indranil Roychoudhury, Susan Frost, Chetan Kulkarni, Shankar Sankararaman, Jose Celaya.
All details presented here are in the public domain and used for information purposes only.

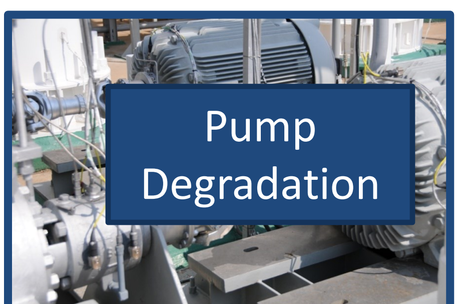
http://prognostics.nasa.gov

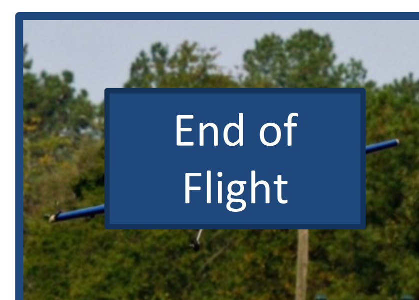
Prognostics CoE

National Aeronautics and Space Administration
NASA

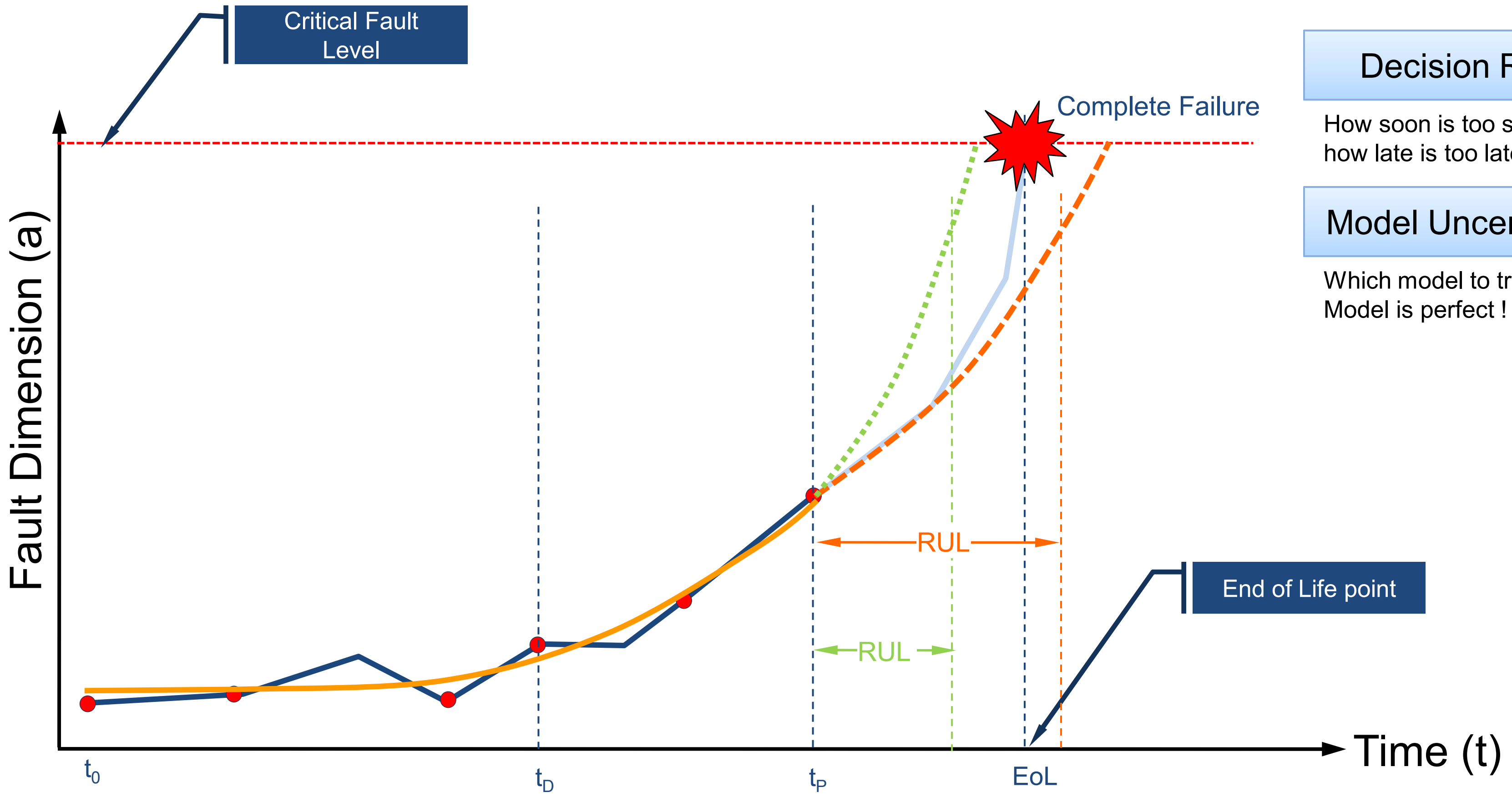
Prognostics Definition

- Prognostics = A prediction of the occurrence of some <u>event of interest</u> to the system
- This event could be
  - Component failure
  - Violation of functional or performance specifications
  - Accomplishment of some system function
  - End of a mission
  - … anything of importance you want to predict, because that knowledge is useful to a decision
- What this event represents does not matter to the framework





Prognostics Framework



Decision Risk

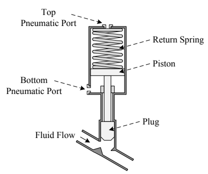
How soon is too soon and how late is too late?

Model Uncertainty

Which model to trust? No Model is perfect !

The Science of Predicting RUL

- RUL: Remaining Useful Life
  - Model underlying physics of a component/subsystem

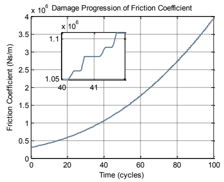


$$f_t(t) = f_g(p_t(t), u_t(t))$$
$$f_b(t) = f_g(p_b(t), u_b(t))$$

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

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

  - Model physics of damage propagation mechanisms



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

  - Determine criteria for End-of-Life Threshold

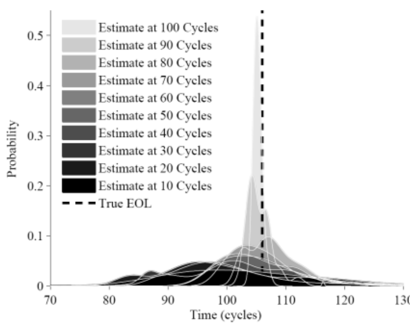
$$EOL(t_P) \triangleq \arg\min_{t \geq t_P} C_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$$

  - Develop algorithms to propagate damage into future

Algorithm 2 EOL Prediction

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

  - Deal with uncertainty



Goals for Prognostics

What does prognostics aim to achieve?

Contingency Management View

- **Increase Safety and Mission Reliability**
  - Improved mission planning; go-no go decisions
  - Ability to reassess mission feasibility; design for resilience
- **Decrease Collateral Damage**
  - Avoid cascading effects onto healthy subsystems
  - Maintain consumer confidence, product reputation

Maintenance Management View

- **Decrease Logistics Costs**
  - More efficient maintenance planning; optimal shop loading
  - Reduced spares
- **Decrease Unnecessary Servicing**
  - Reduce downtime
  - Service only when it is needed

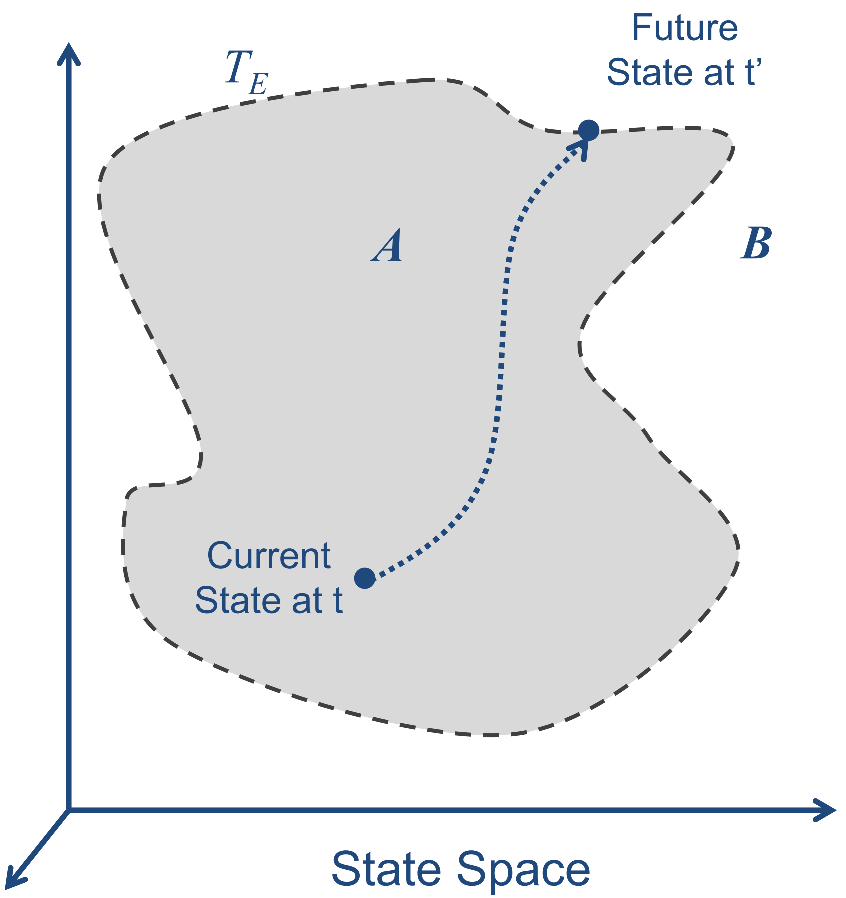
- Prognostics goals should be defined from users' perspectives
- Different solutions and approaches apply for different users

Prognostics

Setting up the Problem

Problem Formulation

- Interested in predicting threshold condition E
- System starts at some state in region $\boldsymbol{A}$, eventually evolves to some new state at which E occurs and moves to region $\boldsymbol{B}$
- T_E defines the boundary between $\boldsymbol{A}$ and $\boldsymbol{B}$
- Must predict the time of event E, k_E, and the time until event E, Δk_E



Problem Formulation

- System described by

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{f}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{v}(k)) \\ \mathbf{y}(k) &= \mathbf{h}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{n}(k))\end{aligned}$$

  – $\mathbf{x}$: states, $\boldsymbol{\theta}$: parameters, $\mathbf{u}$: inputs, $\mathbf{y}$: outputs, $\mathbf{v}$: process noise, $\mathbf{n}$: sensor noise
- Define system event of interest E
- Define threshold function, that evaluates to true when E has occurred

$$T_E(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k))$$

Problem Formulation

- Define k_E

$$k_E(k_P) \triangleq \inf\{k \in \mathbb{N} \colon k \geq k_P \wedge T_E(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k)) = 1\}$$

- Define Δk_E

$$\Delta k_E(k_P) \triangleq k_E(k_P) - k_P$$

- May also be interested in the values of some system variables at k_E

$$\mathbf{z}(k) = \boldsymbol{\psi}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k))$$

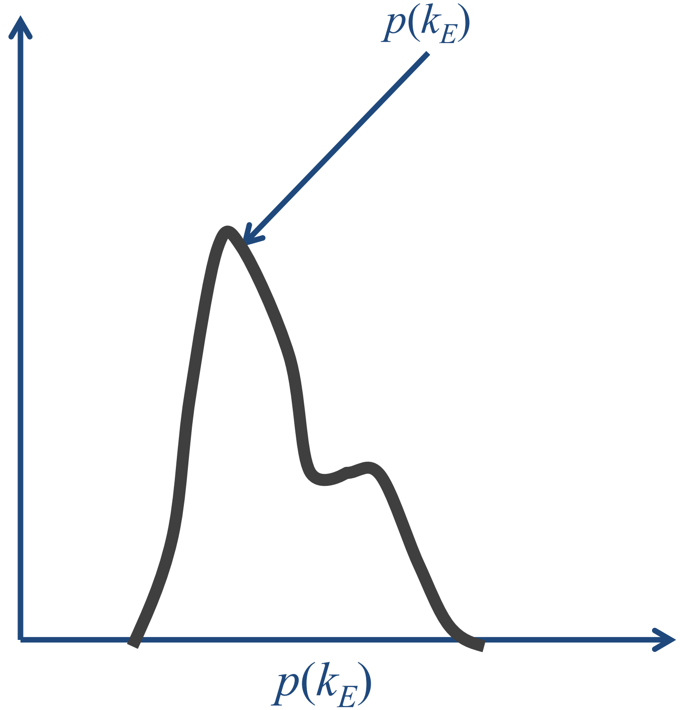
$$\mathbf{z}_E(k_P) \triangleq \mathbf{z}(k_E(k_P))$$

$$\Delta \mathbf{z}_E(k_P) = \mathbf{z}_E(k_P) - \mathbf{z}(k_P)$$

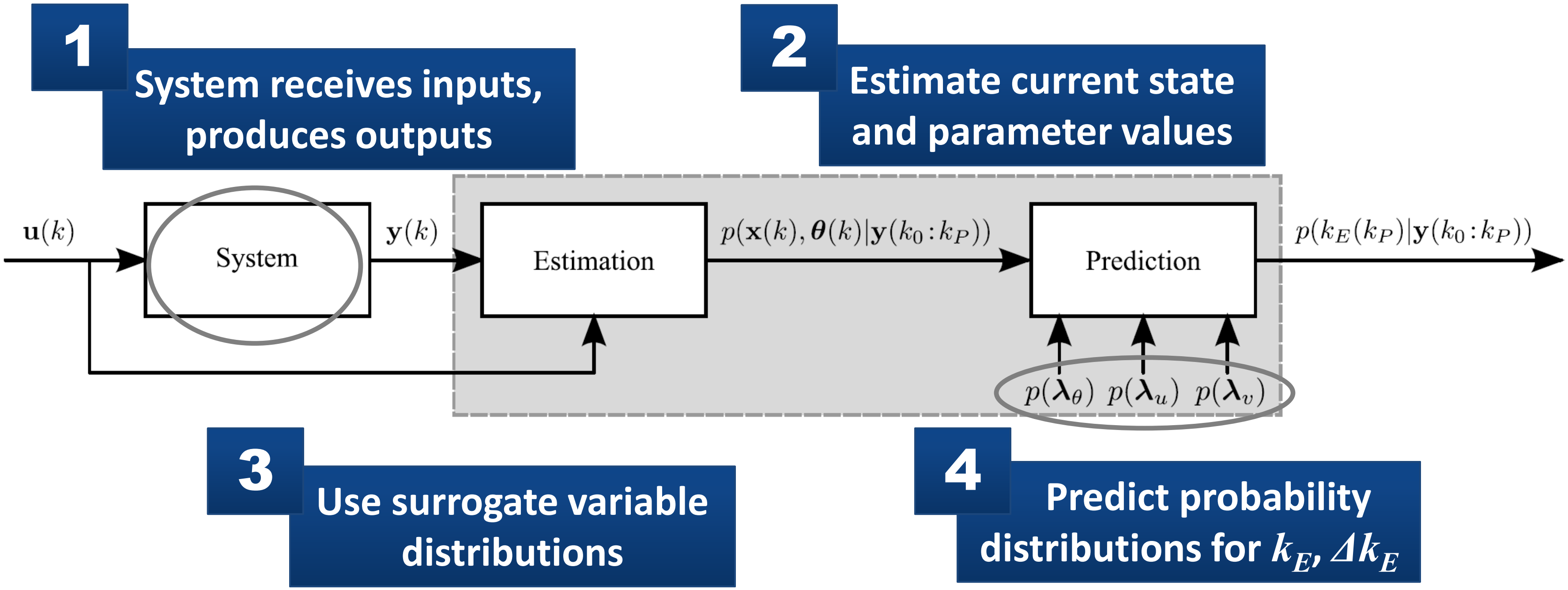
- **Goal** is to compute k_E and its derived variables

Uncertainty

- Goal of prognostics algorithm is to predict true distribution of k_E
  - A misrepresentation of true uncertainty could be disastrous when used for decision-making
- Prognostics algorithm itself adds additional uncertainty
  - Initial state not known exactly
  - Sensor and process noise (stochastic processes with unknown distributions)
  - Model not known exactly
  - System state at k_P not known exactly
  - Future input trajectory distribution not known exactly



Prognostics Architecture



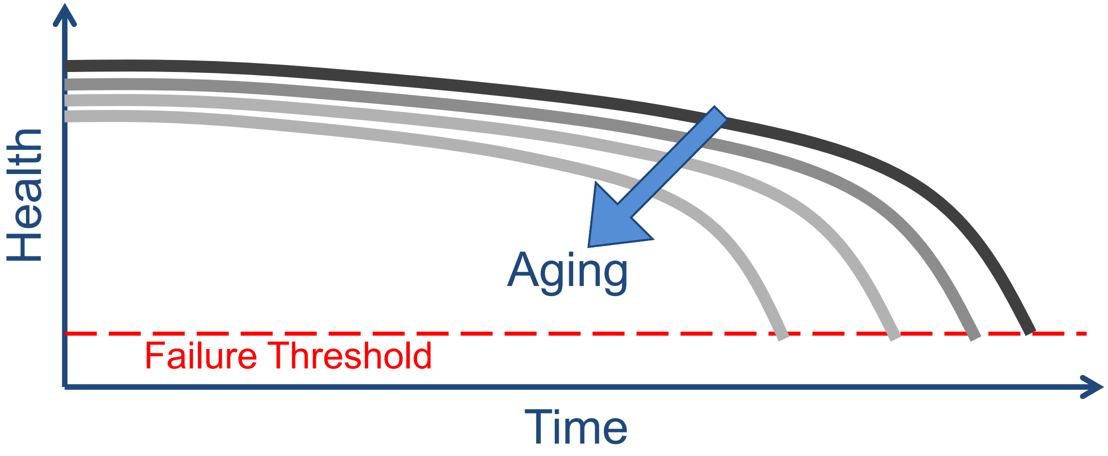
- System gets input and produces output
- Estimation module estimates the states and parameters, given system inputs and outputs
  - Must handle sensor noise and process noise
- Prediction module predicts k_E
  - Must handle state-parameter uncertainty at k_P
  - Must handle future process noise trajectories and input trajectories
  - A diagnosis module can inform the prognostics what model to use

Modeling

What needs to be modeled?
What features do models need?
What are the modeling trade-offs?

What Kind of Models?

- Models for prognostics require the following features
  - Describe dynamics in nominal case (no aging/degradation)
  - Describe dynamics in the faulty/degraded/damaged case
  - Describe dynamics of aging/degradation



- What are the dynamics describing discharge?
- What model parameters change as a result of aging?
- How do the aging parameters change in time?

Estimation Algorithms

How can the system state be estimated?
How does fault diagnosis fit in?
How is uncertainty in estimation handled?

Estimation Problem

- First problem of prognostics is state-parameter estimation
  - What is the current system state and its associated uncertainty?
  - Input: system outputs y from k_0 to k, $y(k_0{:}k)$
  - Output: $p(x(k),\theta(k)|y(k_0{:}k))$
- There are several algorithms that accomplish this, e.g.,
  - Kalman filter (linear systems, additive Gaussian noise)
  - Extended Kalman filter (nonlinear systems, additive Gaussian noise)
  - Unscented Kalman filter (nonlinear systems, additive Gaussian noise)
  - Particle filter (nonlinear systems)

Prediction Algorithms

How is uncertainty represented concisely?
How is uncertainty folded into prediction?
What algorithms are used for prediction?

Prediction Problem

- Most algorithms operate by simulating samples forward in time until E
- Algorithms must account for several sources of uncertainty besides that in the initial state
  - A representation of that uncertainty is required for the selected prediction algorithm
  - A specific description of that uncertainty is required (e.g., mean, variance)
  - Usually no closed-form solution

Prediction

- The `P` function takes an initial state, and a parameter, an input, and a process noise trajectory
  - Simulates state forward using $\mathbf{f}$ until E is reached to compute k_E for a single sample
- Top-level prediction algorithm calls `P`
  - These algorithms differ by how they compute samples upon which to call `P`
- Monte Carlo algorithm (`MC`) takes as input
  - Initial state-parameter estimate
  - Probability distributions for the surrogate variables for the parameter, input, and process noise trajectories
  - Number of samples, N
- `MC` samples from its input distributions, and computes k_E
- The "`construct`" functions describe how to construct a trajectory given surrogate variable samples

Algorithm 1 $k_E(k_P) \leftarrow \mathtt{P}(\mathbf{x}(k_P), \mathbf{\Theta}_{k_P}, \mathbf{U}_{k_P}, \mathbf{V}_{k_P})$

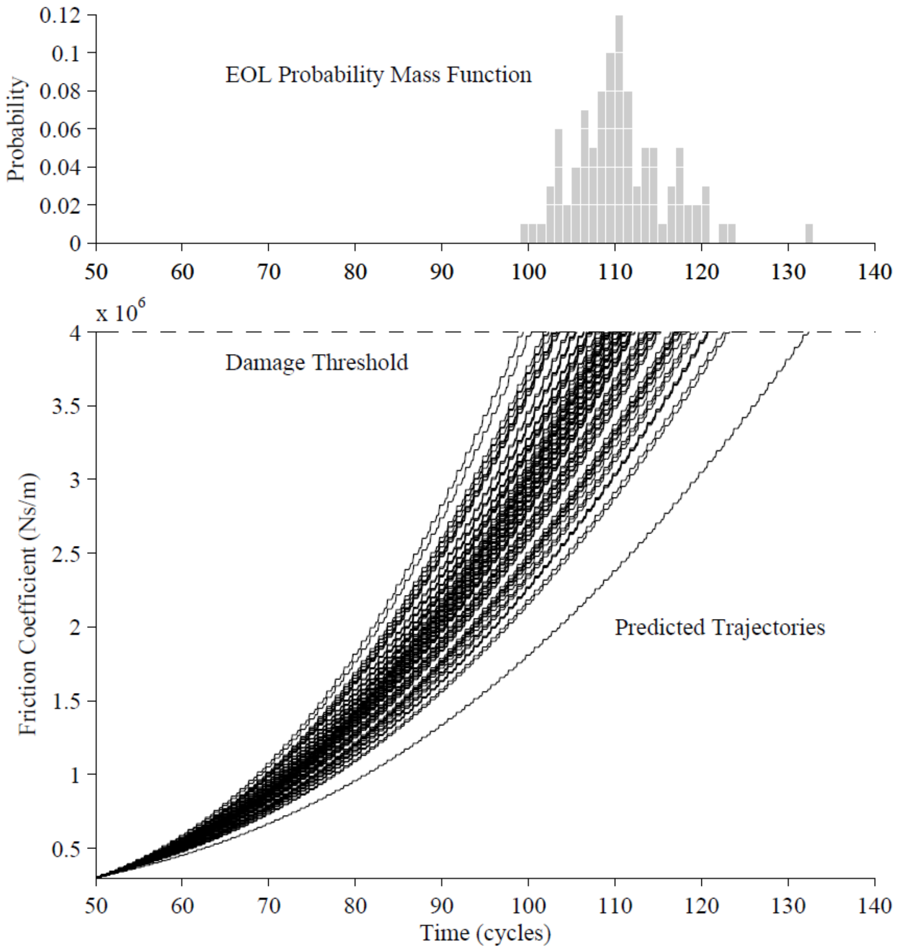
$k \leftarrow k_P$
$\mathbf{x}(k) \leftarrow \mathbf{x}(k_P)$
while $T_E(\mathbf{x}(k), \mathbf{\Theta}_{k_P}(k), \mathbf{U}_{k_P}(k)) = 0$ **do**
  $\mathbf{x}(k+1) \leftarrow \mathbf{f}(k, \mathbf{x}(k), \mathbf{\Theta}_{k_P}(k), \mathbf{U}_{k_P}(k), \mathbf{V}_{k_P}(k))$
  $k \leftarrow k+1$
  $\mathbf{x}(k) \leftarrow \mathbf{x}(k+1)$
end while
$k_E(k_P) \leftarrow k$

Algorithm 2 $\{k_E^{(i)}\}_{i=1}^N = \mathtt{MC}(p(\mathbf{x}(k_P), \boldsymbol{\theta}(k_P)|\mathbf{y}(k_0{:}k_P)), p(\boldsymbol{\lambda}_\theta), p(\boldsymbol{\lambda}_u), p(\boldsymbol{\lambda}_v), N)$

for $i = 1$ **to** N **do**
  $(\mathbf{x}^{(i)}(k_P), \boldsymbol{\theta}^{(i)}(k_P)) \sim p(\mathbf{x}(k_P), \boldsymbol{\theta}(k_P)|\mathbf{y}(k_0{:}k_P))$
  $\boldsymbol{\lambda}_\theta^{(i)} \sim p(\boldsymbol{\lambda}_\theta)$
  $\mathbf{\Theta}_{k_P}^{(i)} \leftarrow \mathtt{construct}\Theta(\boldsymbol{\lambda}_\theta^{(i)}, \boldsymbol{\theta}^{(i)}(k_P))$
  $\boldsymbol{\lambda}_u^{(i)} \sim p(\boldsymbol{\lambda}_u)$
  $\mathbf{U}_{k_P}^{(i)} \leftarrow \mathtt{constructU}(\boldsymbol{\lambda}_u^{(i)})$
  $\boldsymbol{\lambda}_v^{(i)} \sim p(\boldsymbol{\lambda}_v)$
  $\mathbf{V}_{k_P}^{(i)} \leftarrow \mathtt{constructV}(\boldsymbol{\lambda}_v^{(i)})$
  $k_E^{(i)} \leftarrow \mathtt{P}(\mathbf{x}^{(i)}(k_P), \mathbf{\Theta}_{k_P}^{(i)}, \mathbf{U}_{k_P}^{(i)}, \mathbf{V}_{k_P}^{(i)})$
end for

It'll Break at this Time:

- Damage progression, EOL prediction

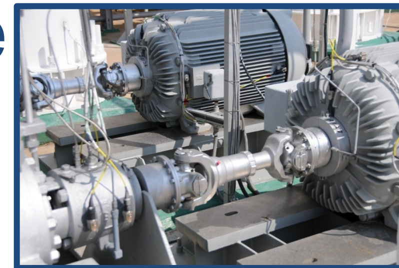


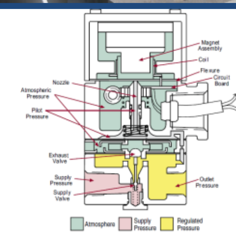
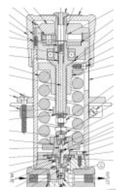
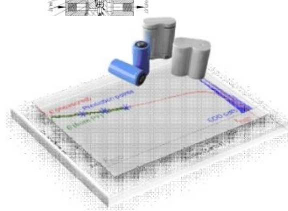
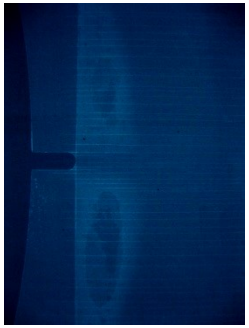
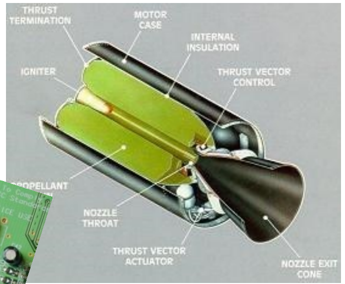
Application Example

- How is this done in practice?

Model Development

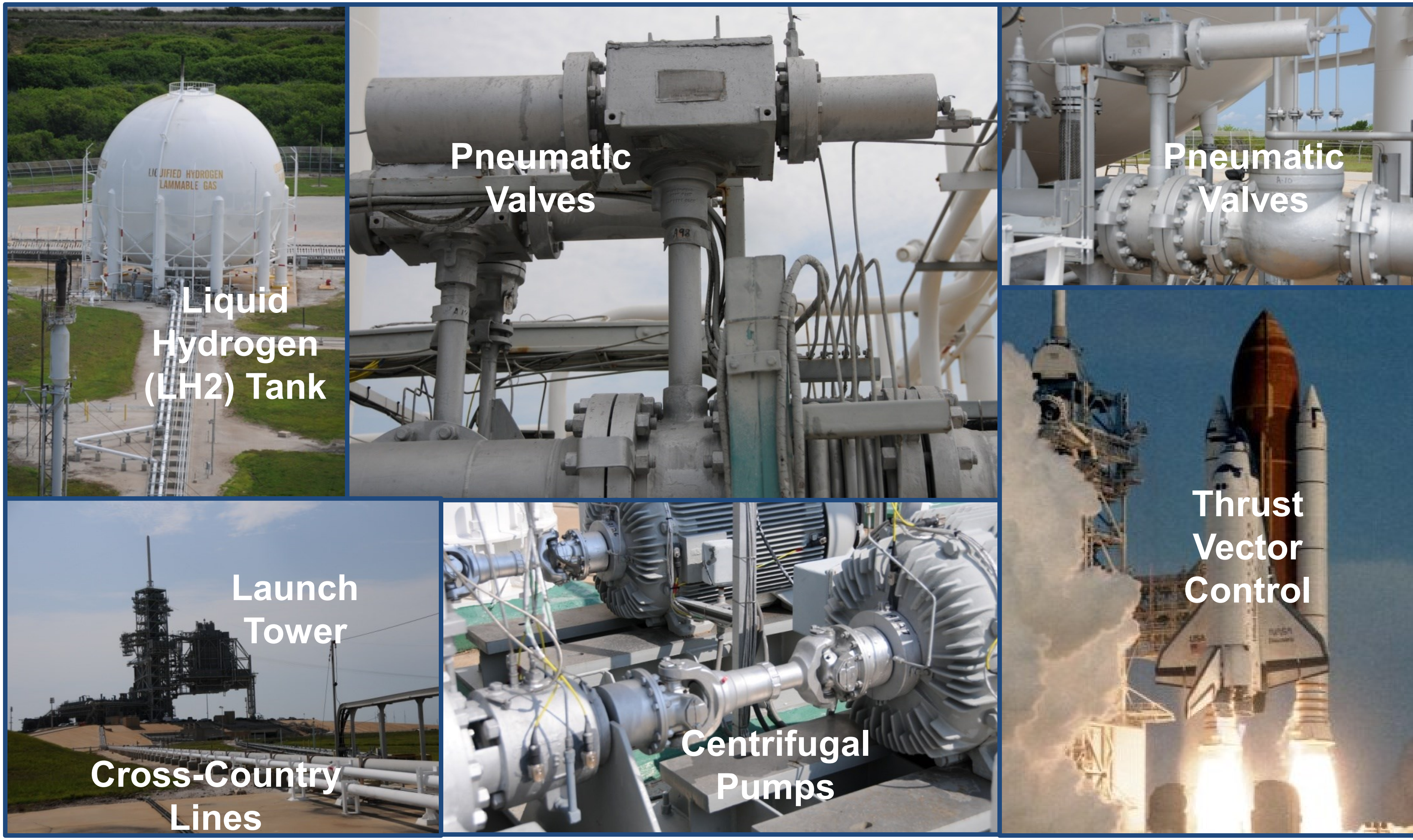
- Models developed include
  - Centrifugal Pumps
  - Cryo Valves (RO* valves)
  - Current/Pressure Transducers
  - Filters
  - Pressure Regulators
  - Solenoid Valves
  - Batteries
  - Composites Structures
  - Electronics (power semiconductors)
  - Motors
  - Electro-Mechanical Actuators



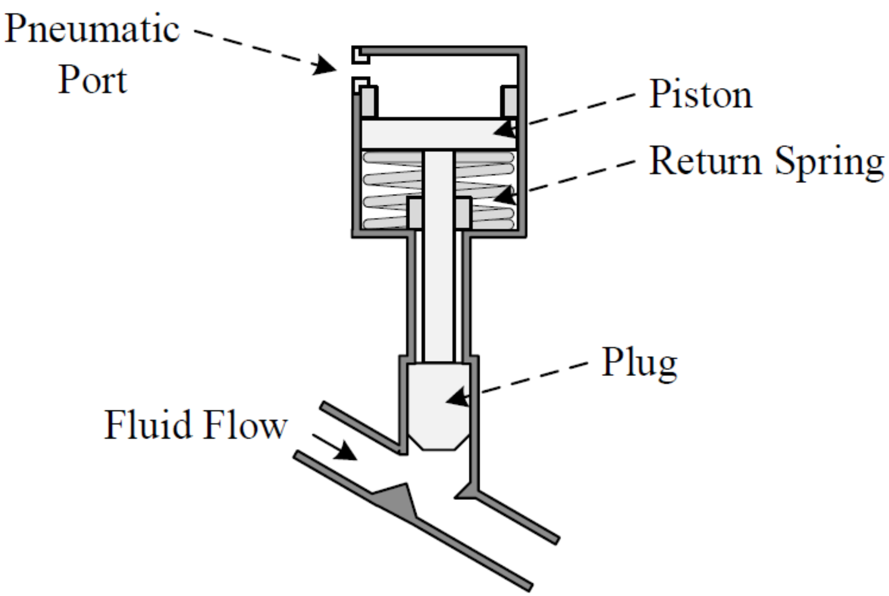


Propellant Loading System



Discrete Control (DV) Valve

- Apply framework to pneumatic valve in propellant loading system
  - Complex mechanical devices used in many domains including aerospace
  - Failures of critical valves can cause significant effects on system function
- Valve operation
  - The valve is opened by filling the chamber with gas up to the supply pressure
  - Evacuating the chamber above the piston down to atmospheric pressure
  - Return spring ensures valve will close upon loss of supply pressure

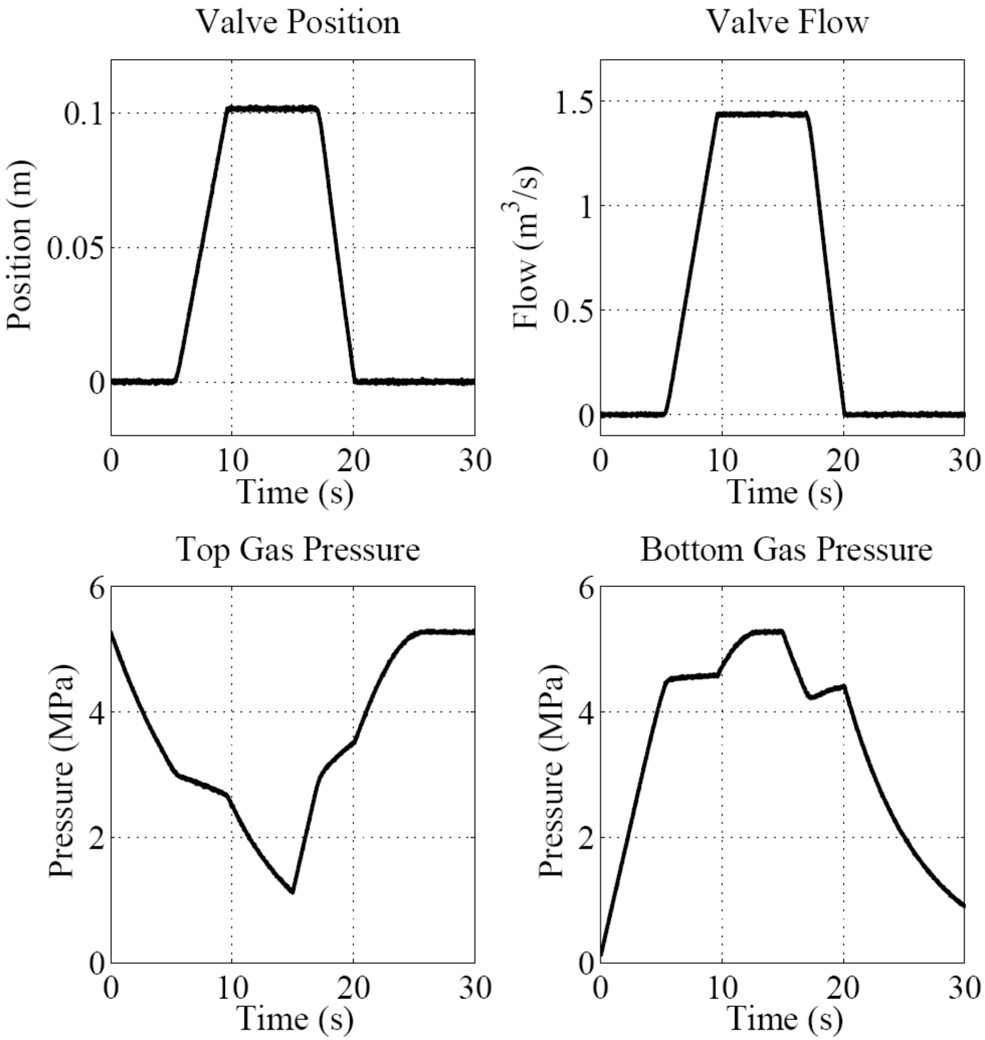


Fault Matrix

Component	Fault Mode	Effects	Injecting Component
Solenoid Valve	Leak across NC seat	If SV energized, and DV valve is open, no effect; if DV valve is closed, no effect. If SV de-energized, and DV valve is closed, DV valve potentially opens; if DV valve is open, DV closes more slowly	V2
	Leak across NO seat	If SV energized, and DV valve is open, loses pressure and DV can start to close; if DV valve is closed, it will open more slowly. If SV de-energized, and DV valve is closed, no effect; if DV valve is open, will close more slowly	V1
	Leak at cylinder port	Same as leak across NC seat	V2
DV	Pneumatic gas leak at valve port	Same effects as leak at SV cylinder port or leak across NO seat	V1
IPT	Leak at output port	Lowers regulated signal pressure which affects the open time of the CV	V3
CV	Pneumatic gas leak at supply pressure port	Lower supply pressure so valve may not open fully, open more slowly	V4
	Pneumatic gas leak at signal pressure port	Lowers regulated pressure	V3
Li-ion Battery	Additional resistance	Reduced charge leaves the DV unable to actuate properly	R1

Pneumatic Valve Modeling

- Piston movement governed by sum of forces, including
  - Friction
  - Spring force
  - Contact forces
  - Gas pressures
  - Fluid pressures
- Mass flows determined by choked and non-choked gas flow equations for orifices
- Nominal operation
  - Opens and closes within 15 seconds
  - Valve closes completely upon loss of supply pressure



Valve Modeling

Some equations

– Pressures $p_t(t)$ and $p_b(t)$

$$p_t(t) = \frac{m_t(t) R_g T}{V_{t_0} + A_p(L_s - x(t))}$$

$$p_b(t) = \frac{m_b(t) R_g T}{V_{b_0} + A_p x(t)}$$

– Gas flows

$$f_t(t) = f_g(p_t(t), u_t(t))$$

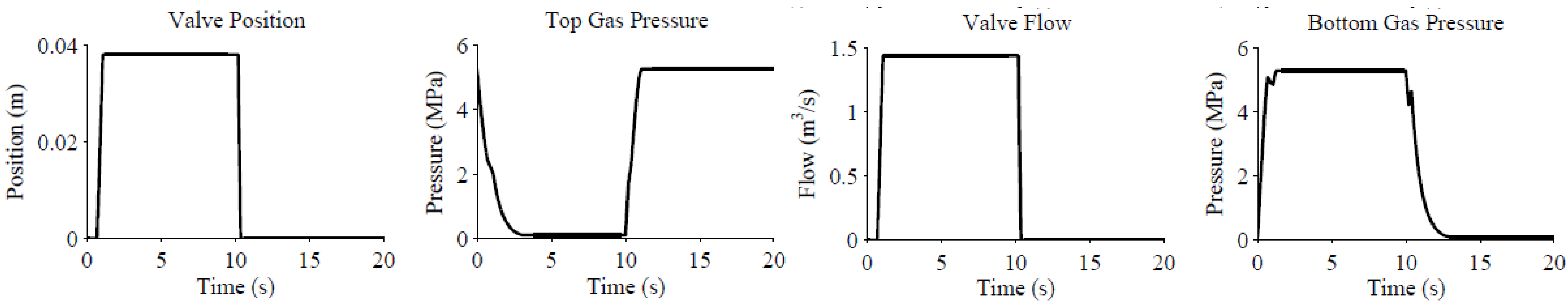
$$f_b(t) = f_g(p_b(t), u_b(t)),$$

– Choked flow

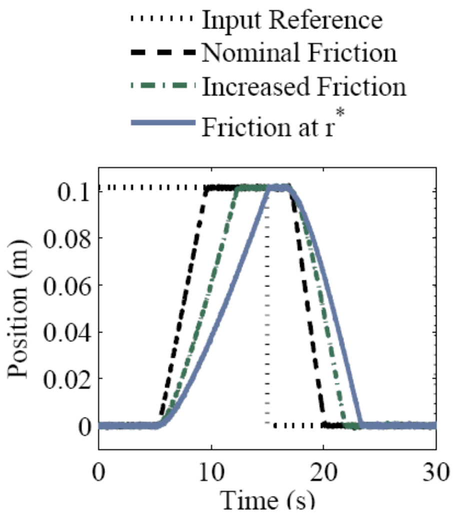
$$f_{g,c}(p_1, p_2) = C_s A_s p_1 \sqrt{\frac{\gamma}{Z R_g T}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}}$$

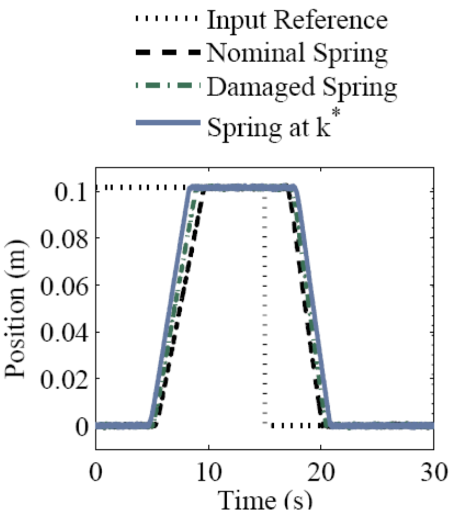
– Fluid flow through valve

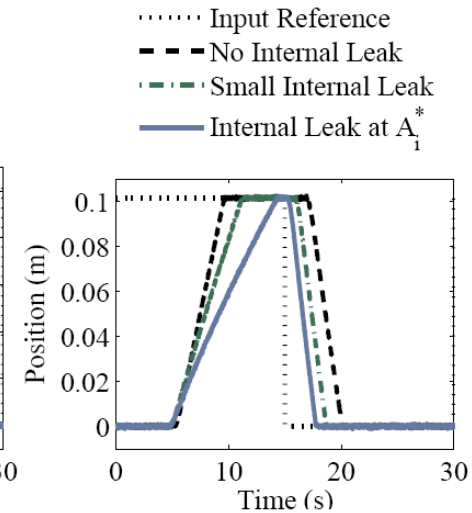
$$f_v(t) = \frac{x(t)}{L_s} C_v A_v \sqrt{\frac{2}{\rho}\left|p_{fl} - p_{fr}\right|}\,\mathrm{sign}(p_{fl} - p_{fr})$$

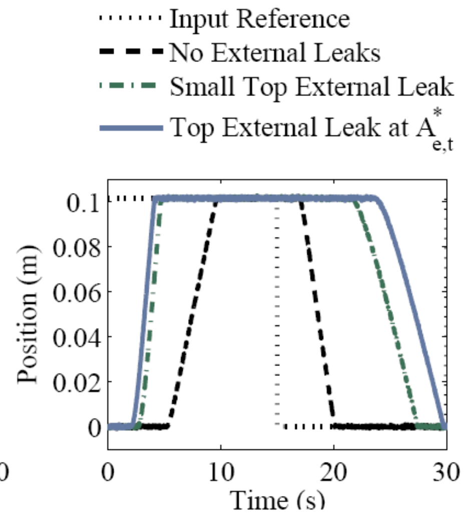


Modeling Damage









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

- Based on 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

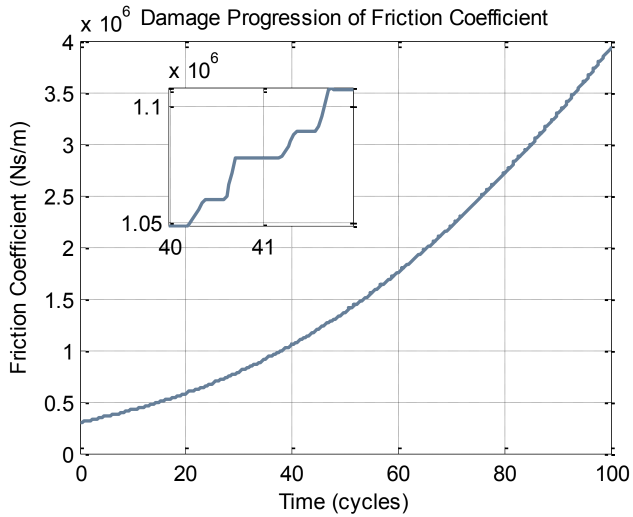
$$\dot{A}_i(t) = w_i |F_f(t) v(t)|$$

Growth of external leak

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

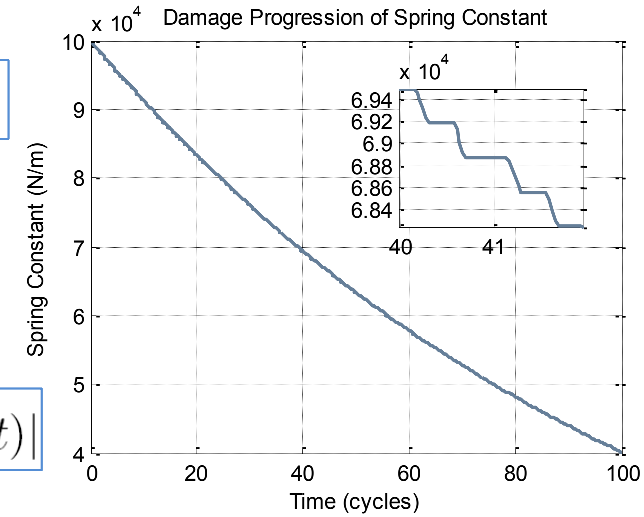
$$\dot{A}_e(t) = w_e$$

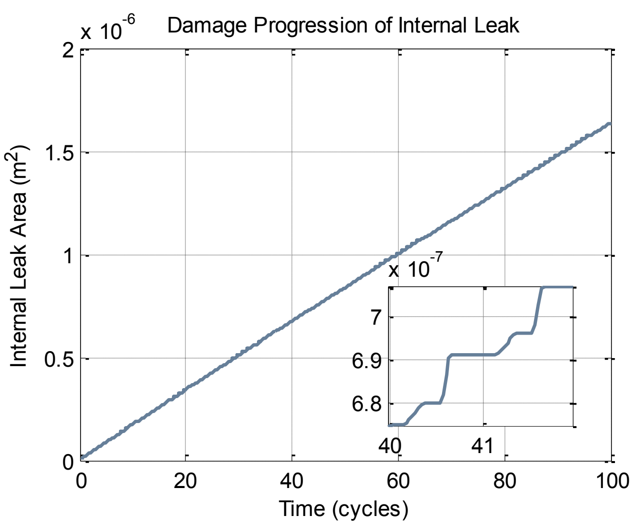
Damage Progression



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

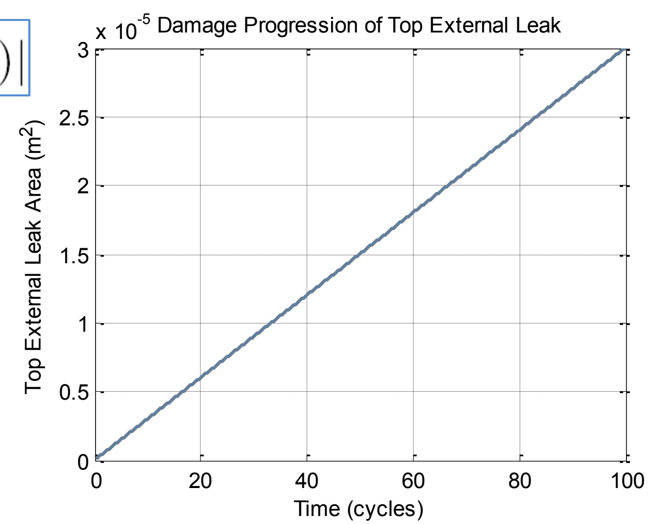
$$\dot{k}(t) = -w_k |F_s(t)v(t)|$$



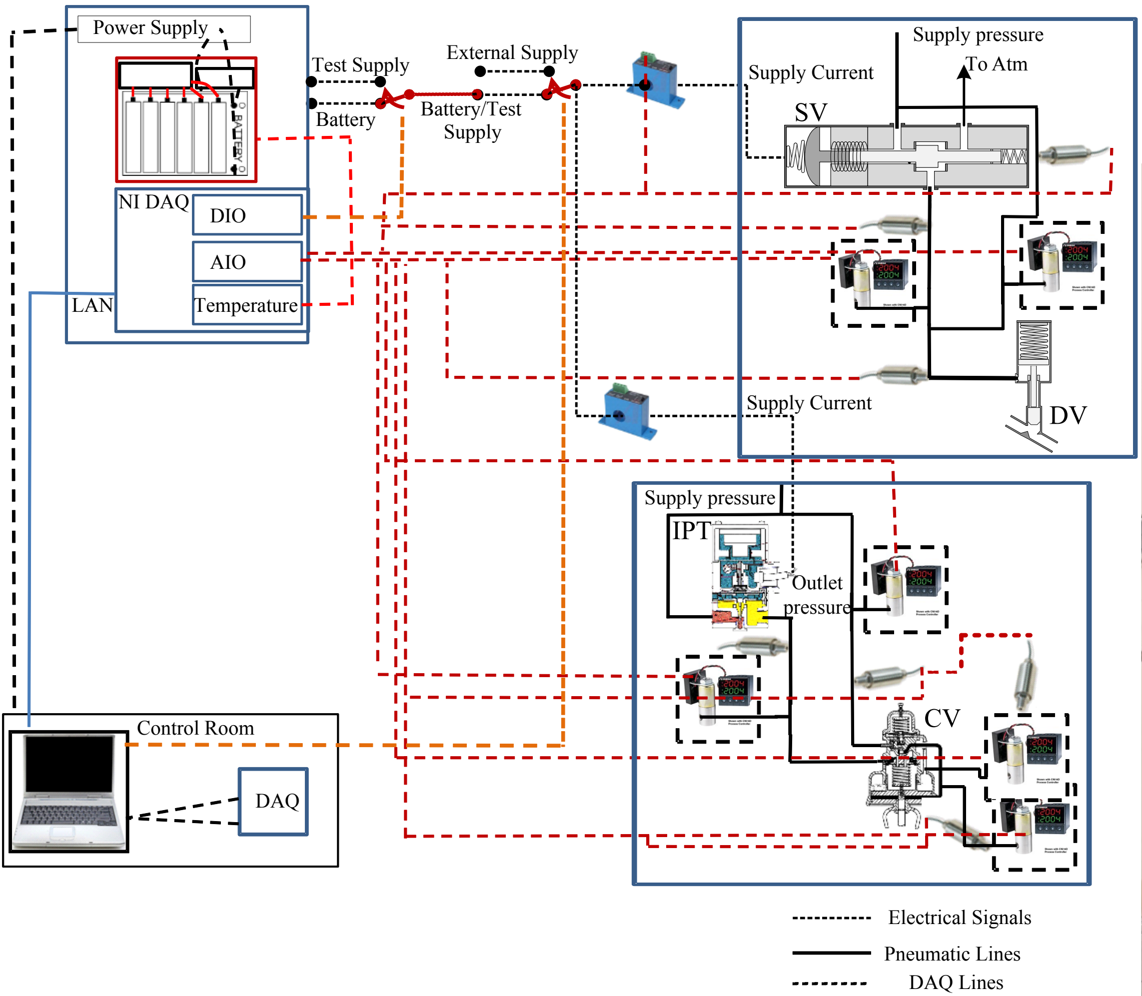


$$\dot{A}_i(t) = w_i |F_f(t)v(t)|$$

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

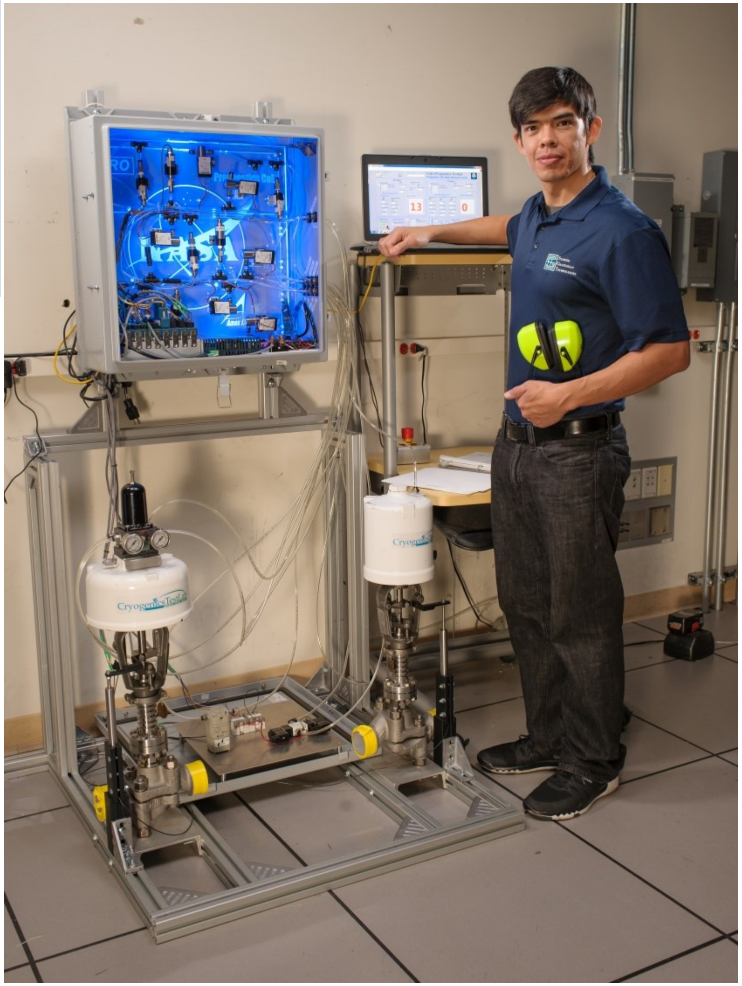


Failure Injection Testbed



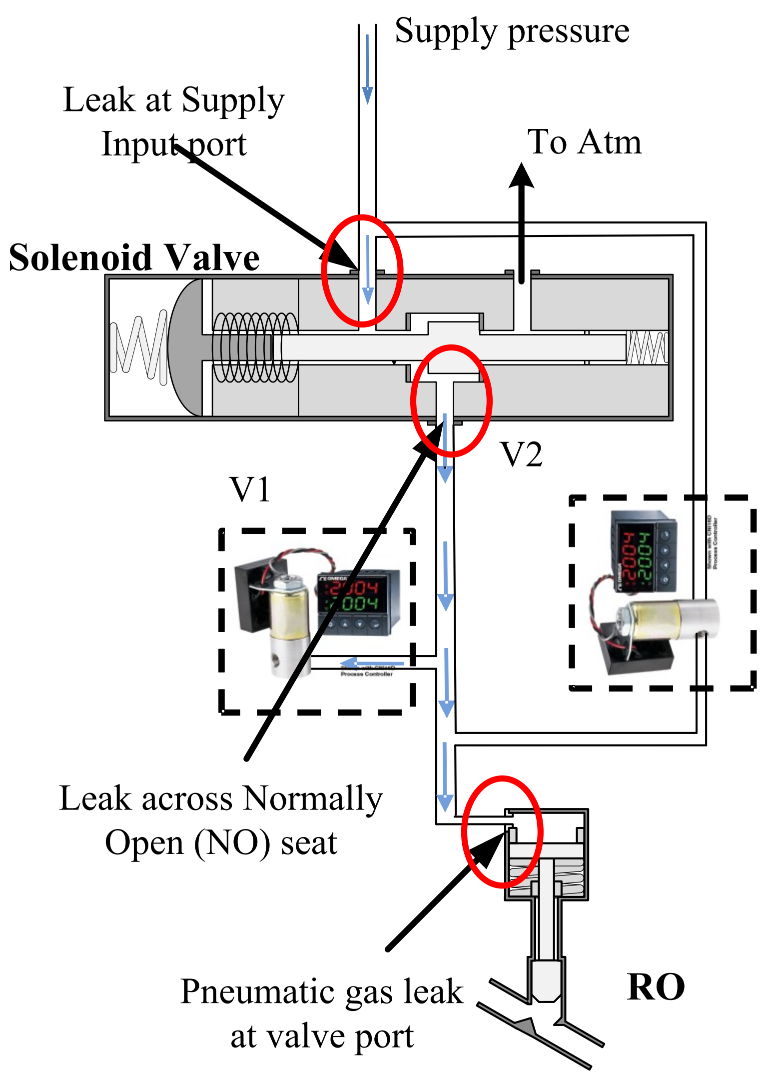
Schematic

Testbed



Atmospheric Leak Fault

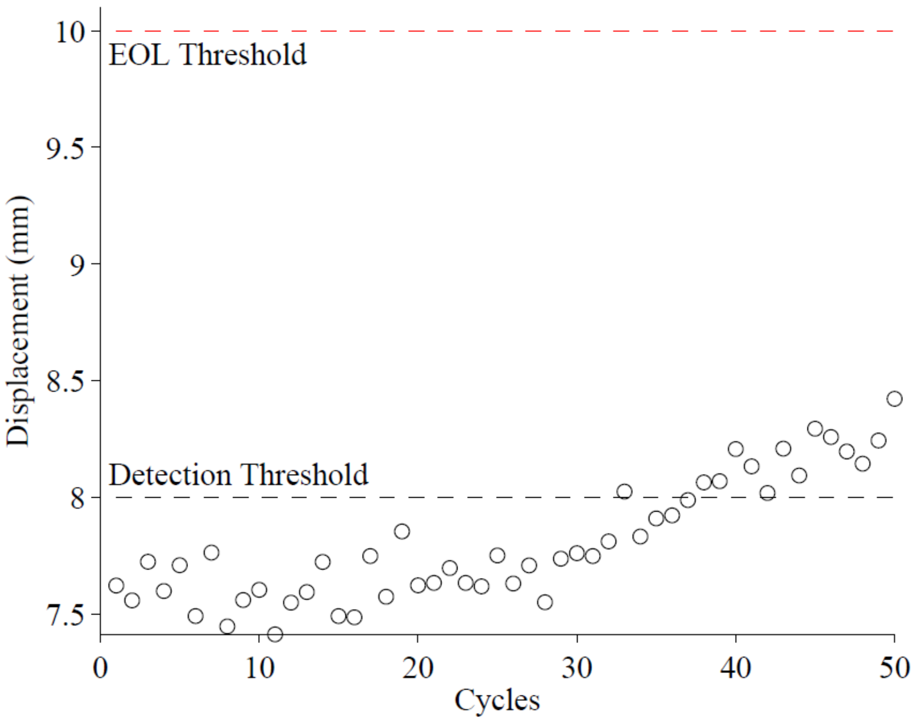
- Fault emulates a leak at the solenoid cylinder port or, when energized, a leak across the NO seat
- Fault injected using proportional valve V1 (at 1% per cycle)
- Affects the **closing time** of RO due to decreased supply pressure.



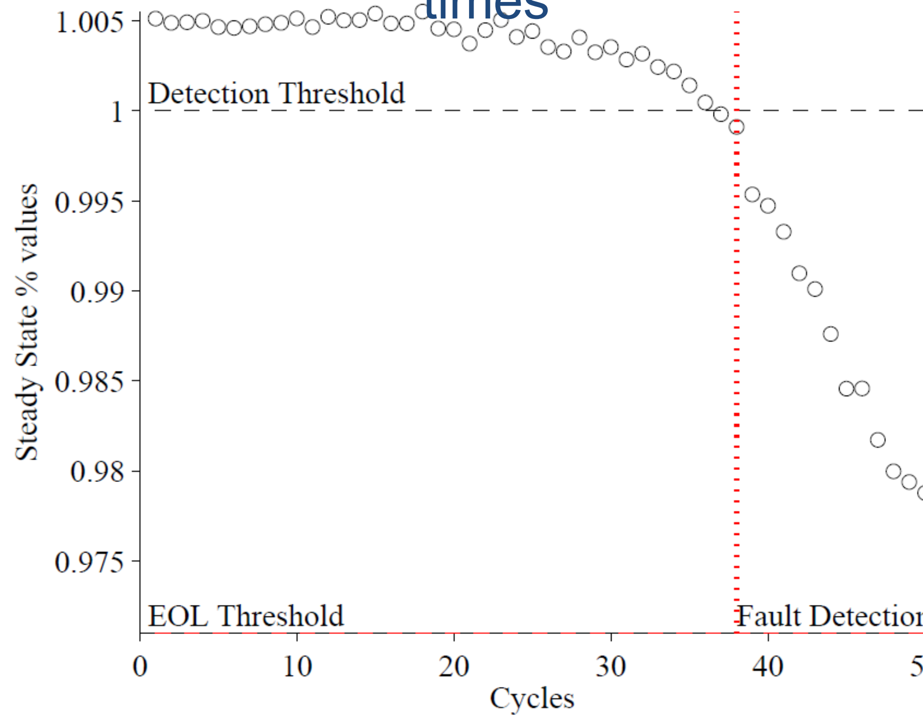
Experiments and results

Leak from Signal

CV open times

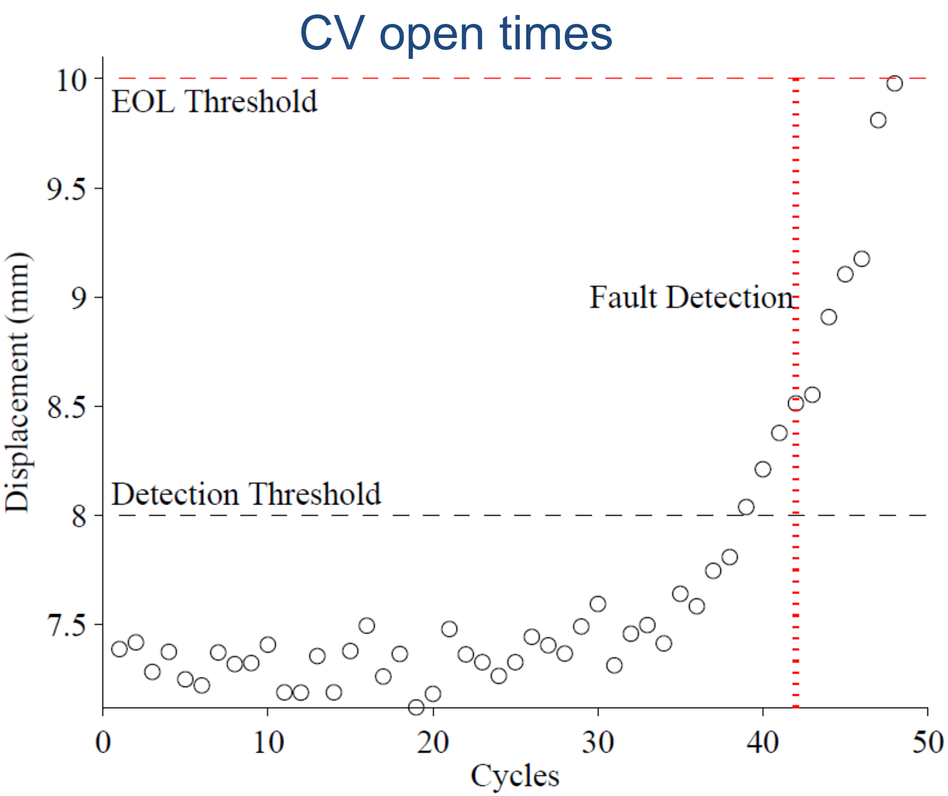


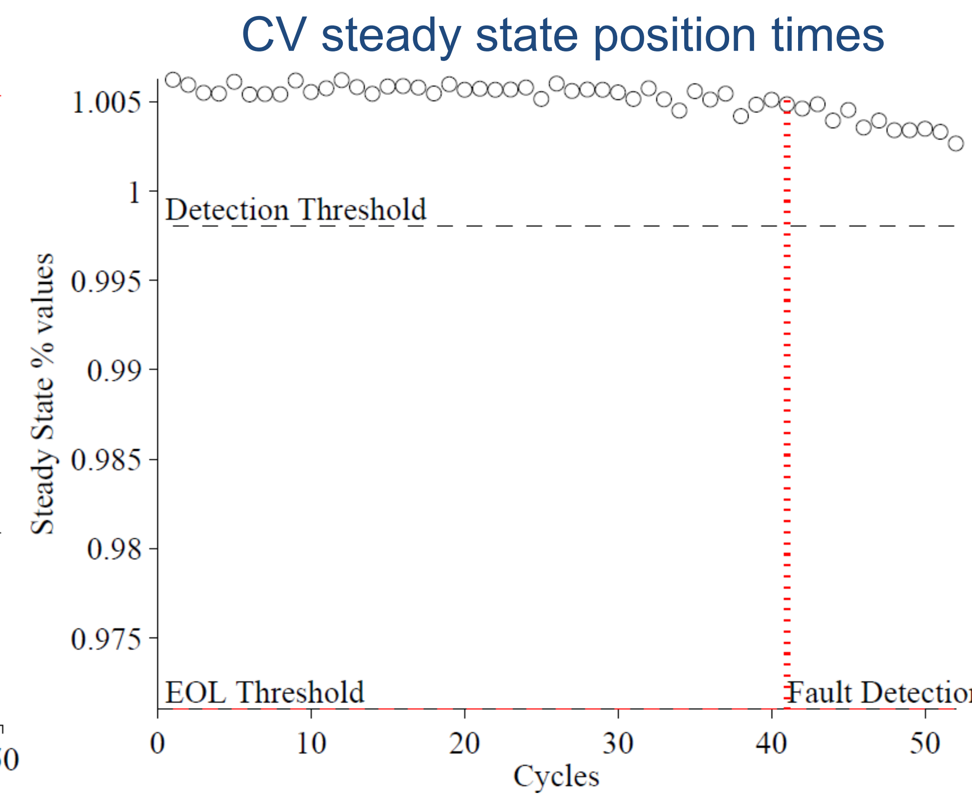
CV steady state position times



- Steady state position threshold detected at 38th cycle
- Relatively no change observed in the open time values

Leak from Supply

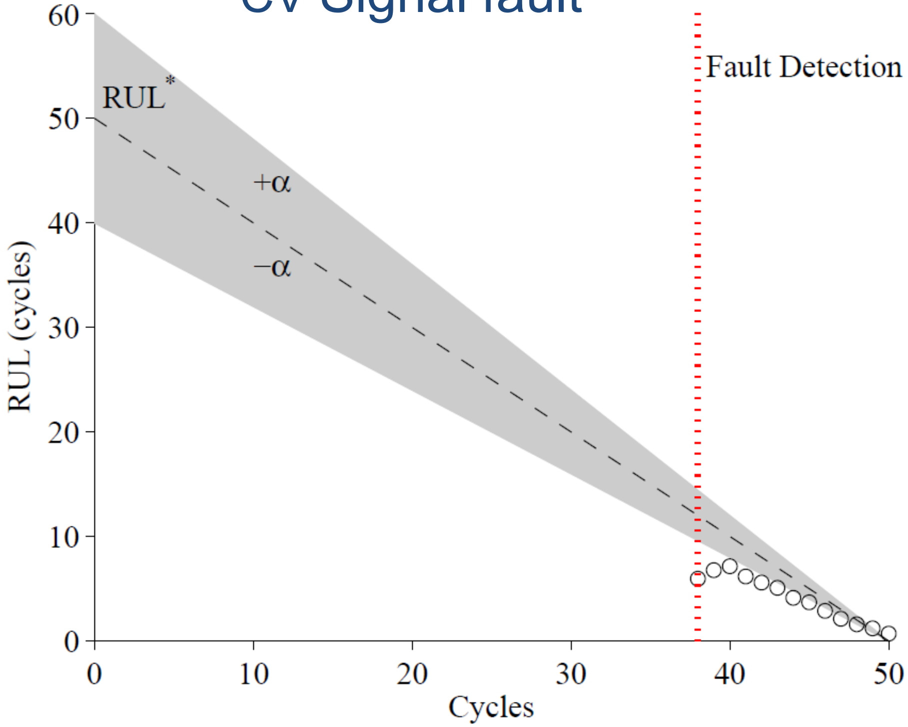




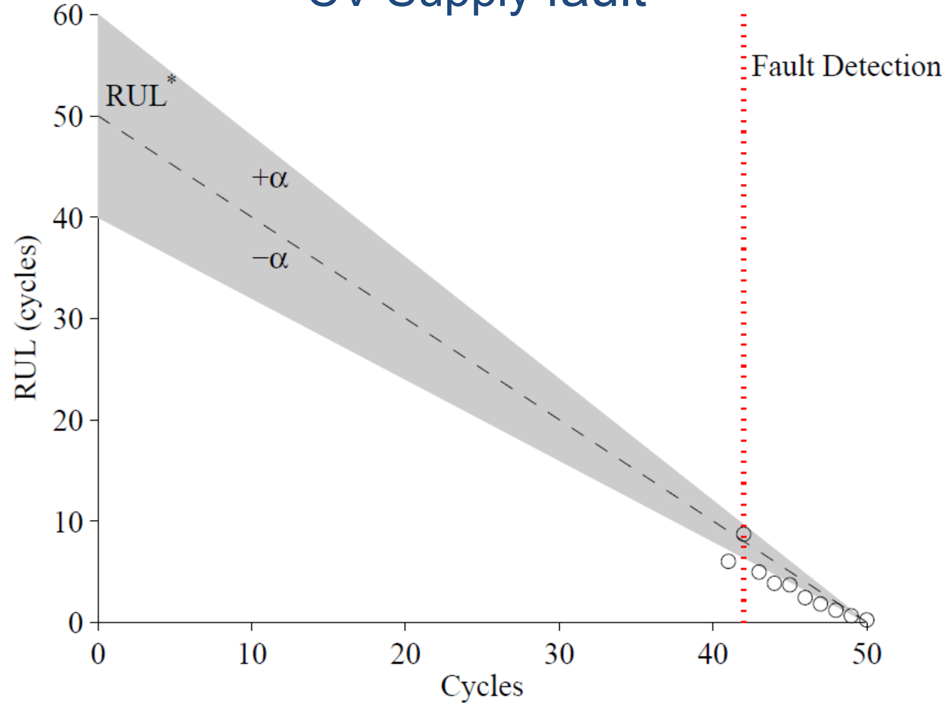
- Open time changes and fault detected at 43rd cycle
- Relatively no change observed in the steady state values

RUL Estimation α-λ Plot

CV Signal fault



CV Supply fault

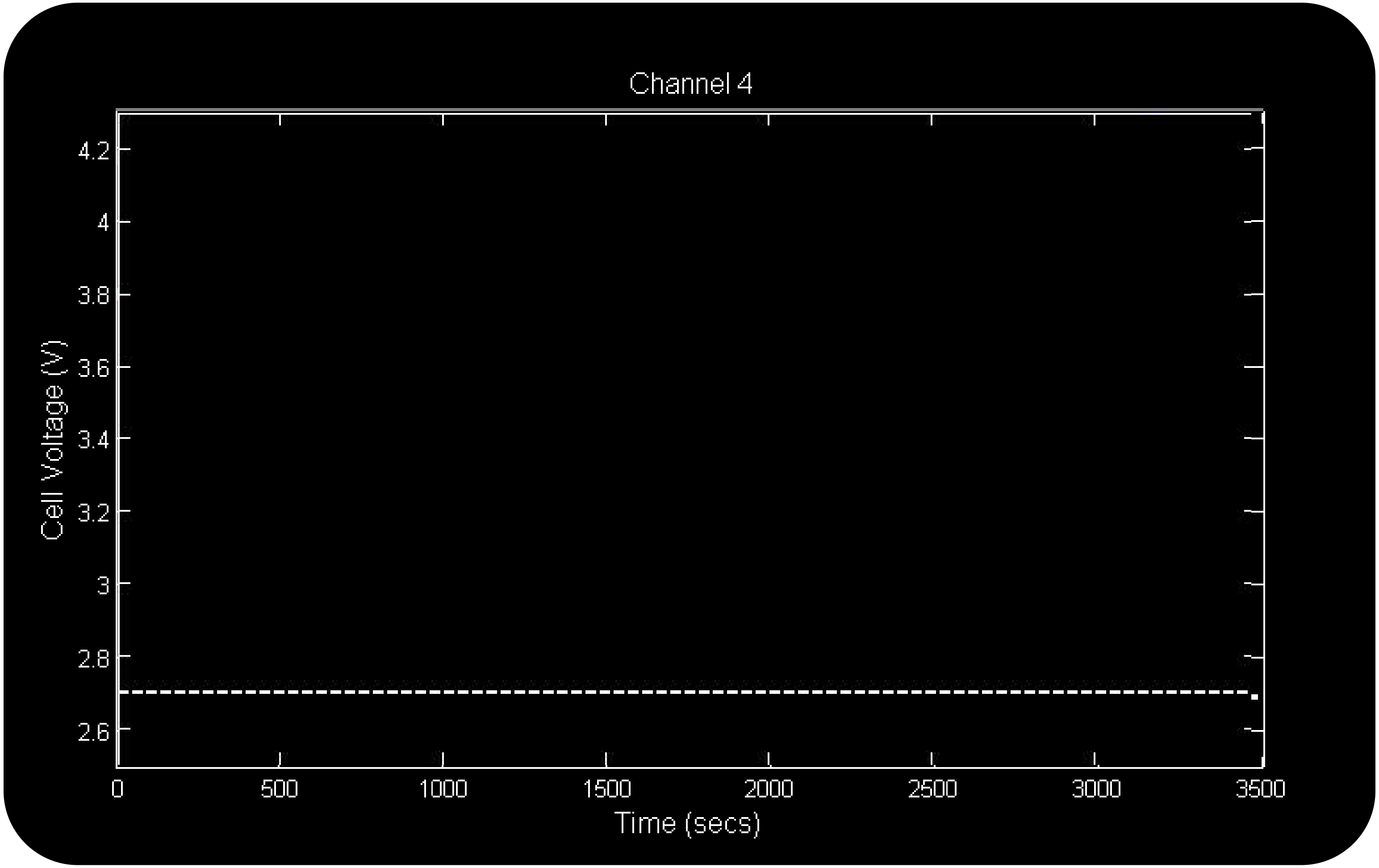


- α–λ is a performance metric for prognostics
- After fault detection within couple of cycles prediction in cone
- Model prediction accurate for both injected faults

Summary

- Predicting Time to Failure
  - Promises to have significant benefit
  - Rigorous Modeling required
  - Dealing with uncertainties important
  - Validation difficult
- Decision-Making
  - Act on remaining life information
  - Based on Prognostic horizon:
    - Fail-Safe Mode
    - Controller Reconfiguration
    - Mission Re-planning
    - Maintenance Scheduling
- Resources
  - Run-to-Failure Data Sets
    - Data repository
      - https://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/
  - Algorithms & Models
    - Open Source
      - https://github.com/nasa/PrognosticsModelLibrary

Prognostics in Action



Last slide