# Model Based Diagnostics and Prognostics Framework for Systems Health Management

Portia Banerjee, Matteo Corbetta, Chetan Kulkarni

SHARP Lab (Sytems Health Analytics Resilience and Prognostics)
Diagnostics and Prognostics Group
KBR Inc., NASA Ames Research Center, Moffett Field, CA

**Advanced Composites Technology Review**

**Stanford University**

**Feb 27, 2020**

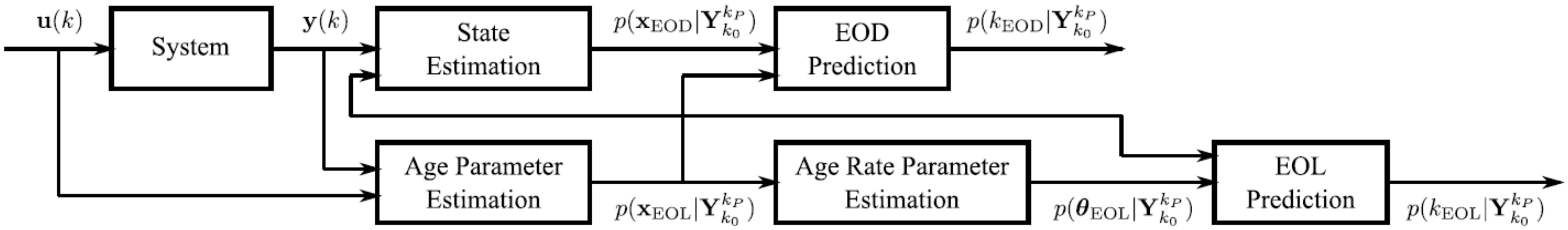
# Overview

Ames Research Center

- Goals
  - Understand system behavior through dynamic models
  - Develop model-based algorithms for state estimation, end of discharge (EOD) prediction, and end of life (EOL) prediction
  - Validate algorithms in the lab and fielded applications
- Algorithms
  - Dynamic state and parameter estimation
  - Uncertainty Representation
  - Prognostics
- Models
  - Electric circuit equivalent (for EOD prediction)
  - Electrochemistry-based model (for EOD and EOL prediction)
- Laboratory capabilities and fielded systems
  - MACCOR battery tester, environmental test chamber
  - Planetary rover testbed
  - Subscale electric aircraft (Edge 540)
  - UAVs vehicles and testbed

# Prognostics Architecture






- System gets input and produces output
- Estimation module estimates the states and parameters, given system inputs and outputs
  - Must handle sensor noise
  - Must handle process noise
- For some event $E$, e.g., end-of-discharge or end-of-life, prediction module predicts $k_E$
  - Must handle state-parameter uncertainty at $k_P$
  - Must handle future process noise trajectories
  - Must handle future input trajectories
  - A diagnosis module can inform the prognostics what model to use
- In model-based approaches, require a dynamic model of the battery

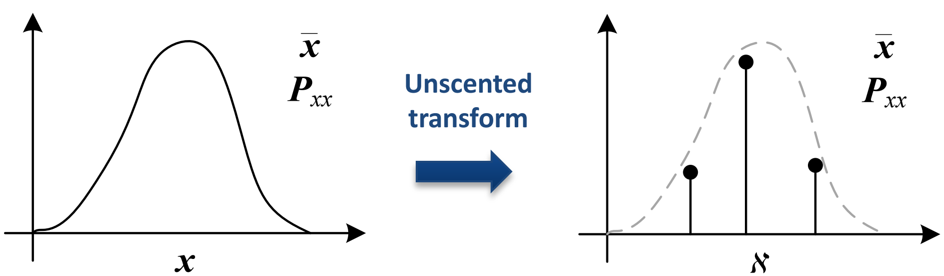
# State Estimation

Ames Research Center

- What is the current system state and its associated uncertainty?
  - Input: system outputs $y$ from $k_0$ to $k$, $y(k_0{:}k)$
  - Output: $p(x(k),\theta(k)|y(k_0{:}k))$
- Most of the models are nonlinear e.g battery, so require nonlinear state estimator (e.g., extended Kalman filter, particle filter, unscented Kalman filter)
- Use unscented Kalman filter (UKF)
  - Straightforward to implement and tune performance
  - Computationally efficient (number of samples linear in size of state space)

# Unscented Kalman Filter

- The UKF is an approximate nonlinear filter, and assumes additive, Gaussian process and sensor noise
- Handles nonlinearity by using the concept of sigma points
  - Transform mean and covariance of state into set of samples, called sigma points, selected deterministically to preserve mean and covariance
  - Sigma points are transformed through the nonlinear function and recover mean and covariance of transformed sigma points




$$w^i = \begin{cases} \dfrac{\kappa}{(n_x + \kappa)}, & i = 0 \\ \dfrac{1}{2(n_x + \kappa)}, & i = 1, \ldots, 2n_x \end{cases}$$

$$\mathcal{X}^i = \begin{cases} \bar{\mathbf{x}}, & i = 0 \\ \bar{\mathbf{x}} + \left(\sqrt{(n_x+\kappa)\mathbf{P}_{xx}}\right)^i, & i = 1, \ldots, n_x \\ \bar{\mathbf{x}} - \left(\sqrt{(n_x+\kappa)\mathbf{P}_{xx}}\right)^i, & i = n_x+1, \ldots, 2n_x \end{cases}$$

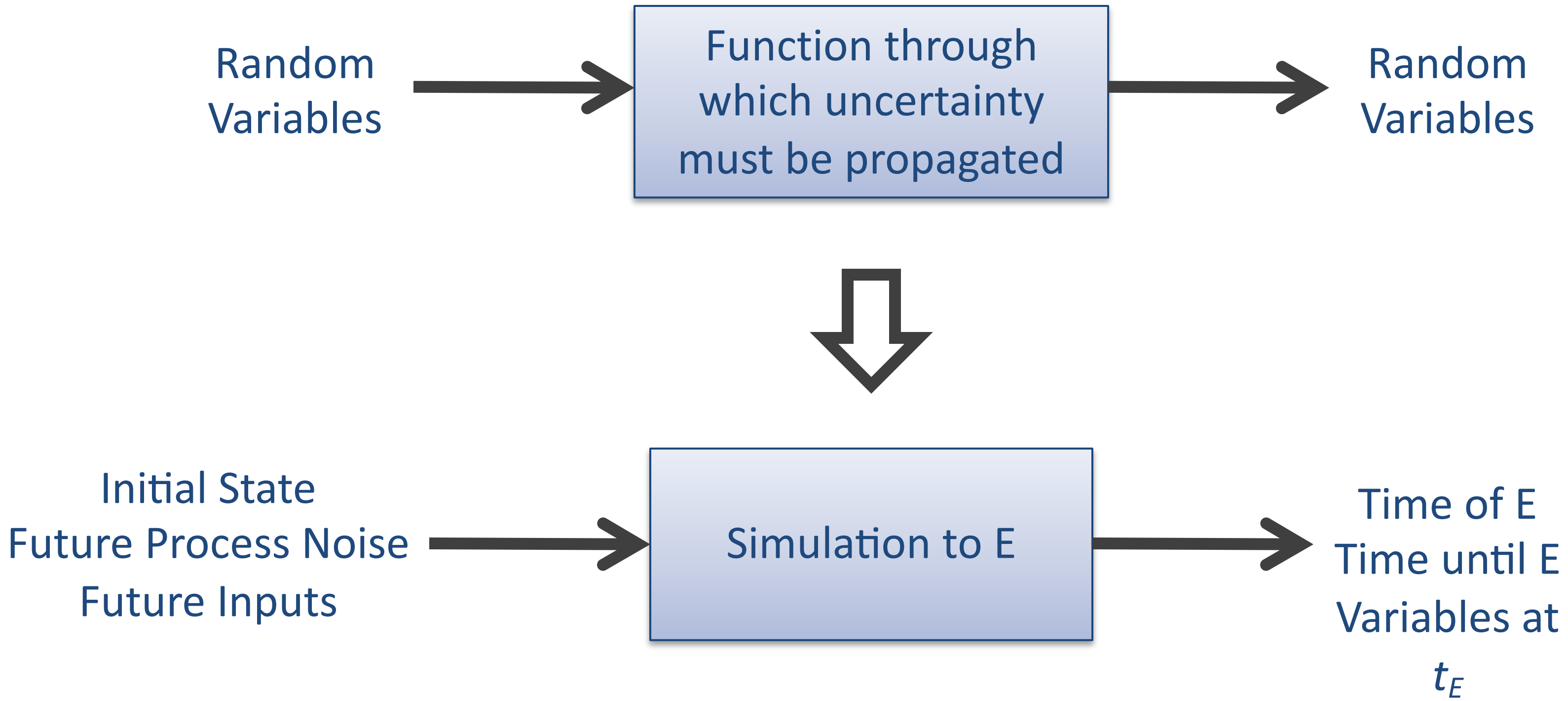
**Symmetric Unscented Transform**

- Number of sigma points is linear in the size of the state dimension

# Prediction

- What is $k_E$ and what is its uncertainty?
  – Input: $p(x(k),\theta(k)|y(k_0:k))$
  – Output: $p(k_E)$
- 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)

# Uncertainty Quantification



# Uncertainty Representation

Ames Research Center

- To predict $k_E$, need to account for following sources of uncertainty:
  - Initial state at $k_P$: $\mathbf{x}(k_p)$
  - Parameter values for $k_P$ to $k_E$: $\mathbf{\Theta}_{k_P}$
  - Inputs for $k_P$ to $k_E$: $\mathbf{U}_{k_P}$
  - Process noise for $k_P$ to $k_E$: $\mathbf{V}_{k_P}$
- Trajectories represented indirectly through parameterized equations describing the trajectories, where probability distributions for the parameters are specified
  - Sample these parameter variables to sample a trajectory
  - For example, constant power trajectory represented through $u(k) = c$, for all $k > k_P$, where $c$ is random

# Prediction Algorithm



- The $\mathtt{P}$ 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 computes $k_E$ for a single sample
- Top-level prediction algorithm calls $\mathtt{P}$
  - These algorithms differ by how they compute samples upon which to call $\mathtt{P}$
- Monte Carlo algorithm ($\mathtt{MC}$) takes as input
  - Initial state-parameter estimate
  - Probability distributions for the surrogate variables for the parameter, input, and process noise trajectories
  - Number of samples, $N$
- $\mathtt{MC}$ samples from its input distributions, and computes $k_E$
- The "$\mathtt{construct}$" functions describe how to construct a trajectory given trajectory parameters

**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**

    $(\mathbf{x}^{(i)}(k_P), \boldsymbol{\theta}^{(i)}(k_P)) \sim p(\mathbf{x}(k_P), \boldsymbol{\theta}(k_P)|\mathbf{y}(k_0{:}k_P))$

    $\boldsymbol{\lambda}_\theta^{(i)} \sim p(\boldsymbol{\lambda}_\theta)$

    $\mathbf{\Theta}_{k_P}^{(i)} \leftarrow \mathtt{construct\Theta}(\boldsymbol{\lambda}_\theta^{(i)}, \boldsymbol{\theta}^{(i)}(k_P))$

    $\boldsymbol{\lambda}_u^{(i)} \sim p(\boldsymbol{\lambda}_u)$

    $\mathbf{U}_{k_P}^{(i)} \leftarrow \mathtt{constructU}(\boldsymbol{\lambda}_u^{(i)})$

    $\boldsymbol{\lambda}_v^{(i)} \sim p(\boldsymbol{\lambda}_v)$

    $\mathbf{V}_{k_P}^{(i)} \leftarrow \mathtt{constructV}(\boldsymbol{\lambda}_v^{(i)})$

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

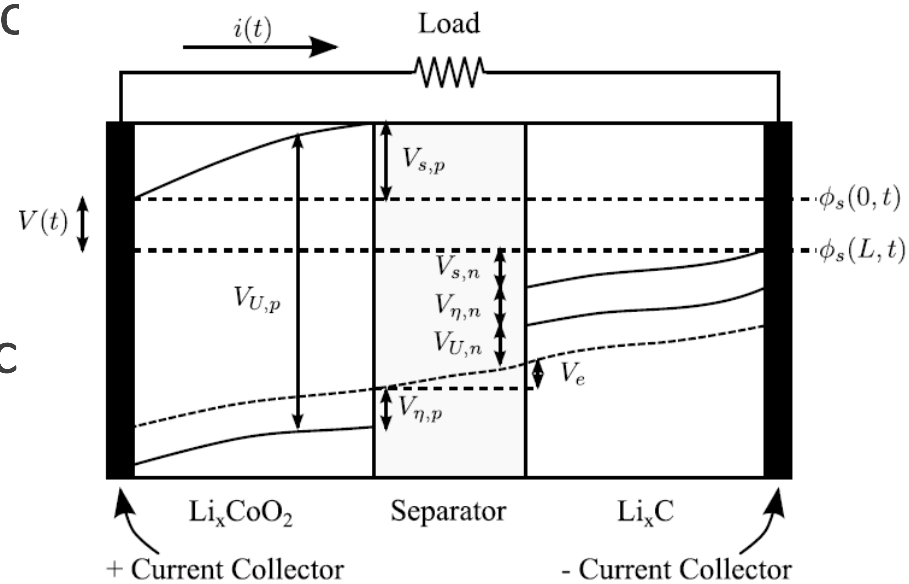
**end for**

# Battery Prognostics

# Electrochemistry Battery Modeling

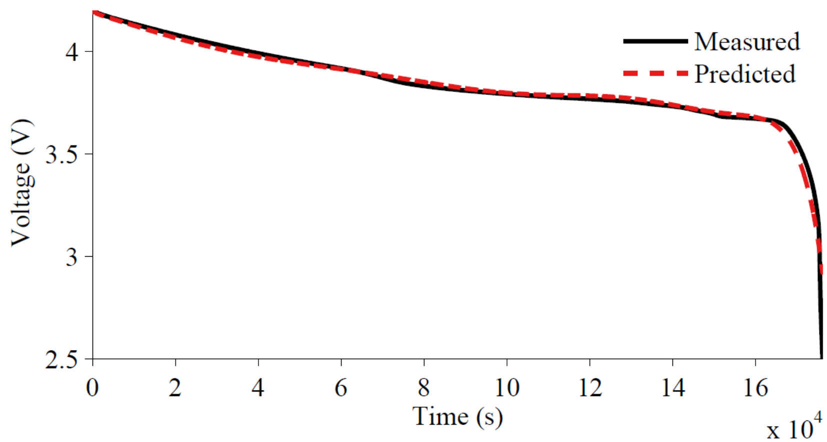
Ames Research Center

- Lumped-parameter, ordinary differential equations
- Capture voltage contributions from different sources
  - Equilibrium potential → Nernst equation with Redlich-Kister expansion
  - Concentration overpotential → split electrodes into surface and bulk control vc
  - Surface overpotential → Butler-Volmer equation applied at surface layers
  - Ohmic overpotential → Constant lumped resistanc accounting for current collector resistances, electrolyte resistance, solid-phase ohmic resistan

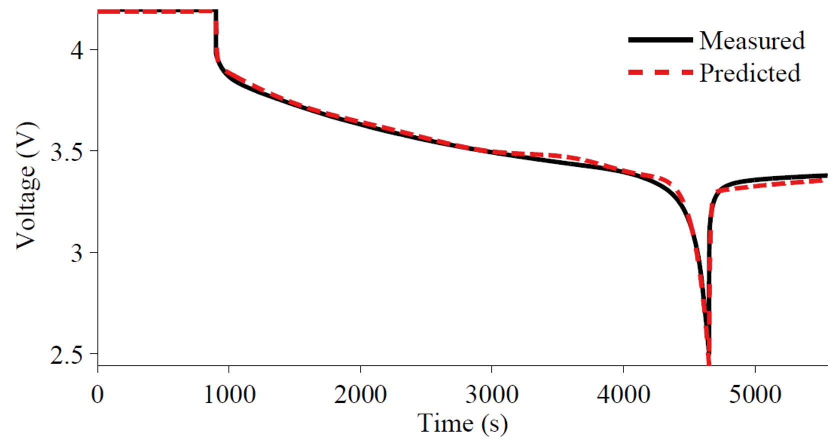


# Battery Model Validation



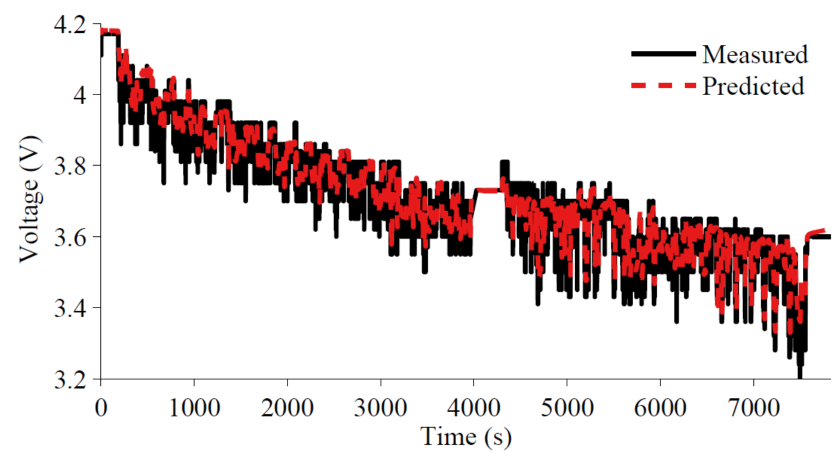



"Open-Circuit" Discharge Curve




Nominal 2A Discharge Curve

Model matches well for open-circuit, nominal discharge, and variable-load discharges on the rover.



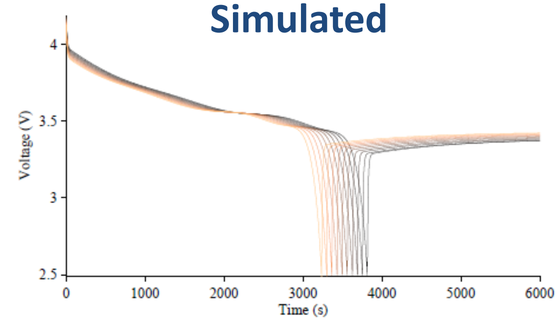

Rover Battery Discharge Curve

# Battery Aging

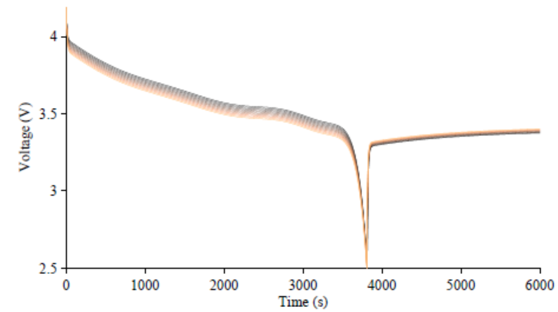


- Contributions from both decrease in mobile Li ions (lost due to side reactions related to aging) and increase in internal resistance
  - Modeled with decrease in "$q^{max}$" parameter, used to compute mole fraction
  - Modeled with increase in "$R_o$" parameter capturing lumped resistances

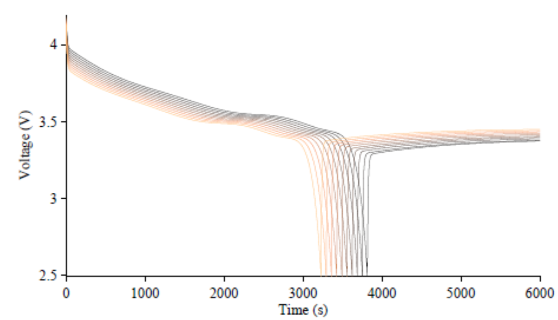
Simulated



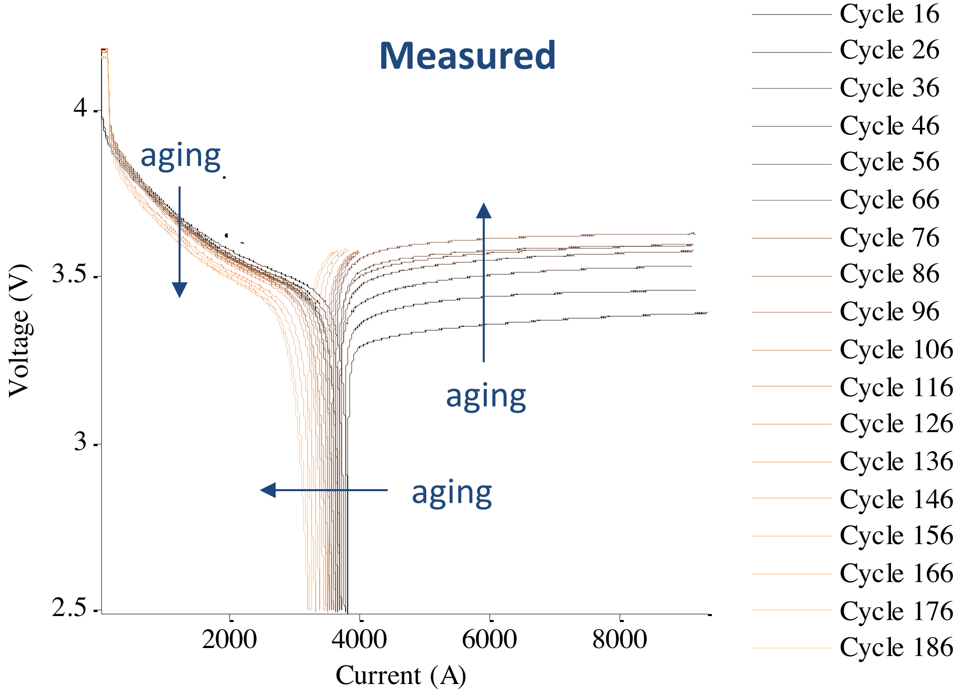

(a) Decreasing $q^{max}$.




(b) Decreasing $R_o$.



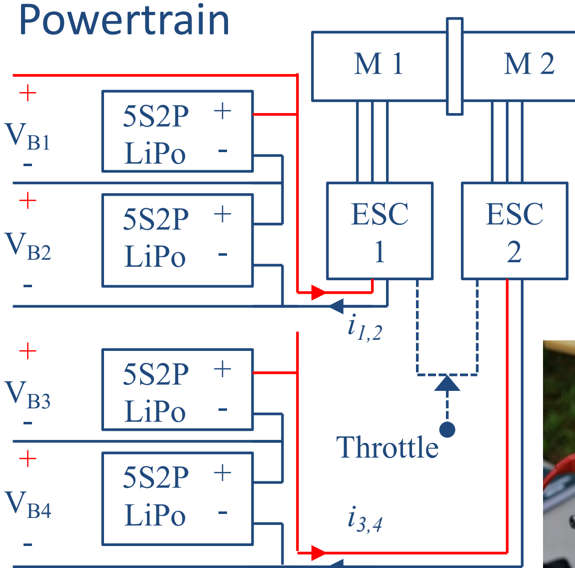

(c) Decreasing $q^{max}$ and $R_o$.

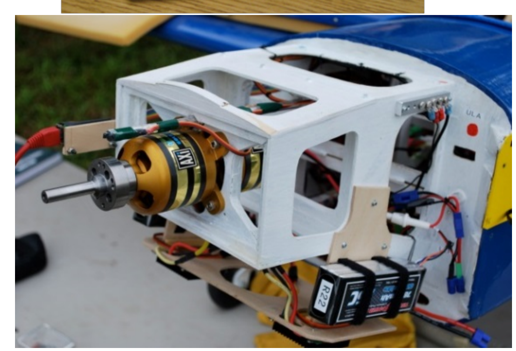


# Edge 540-T

- Subscale electric aircraft operated at NASA Langley Research Center
- Powered by four sets of Li-polymer batteries
- Estimate SOC online and provide EOD and remaining flight time predictions for ground-based pilots

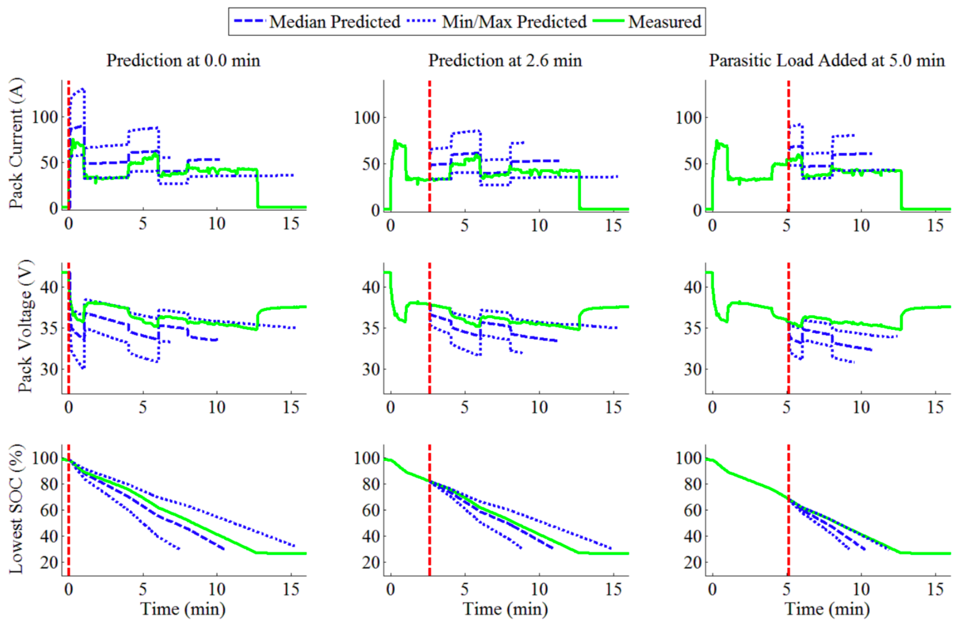
Powertrain



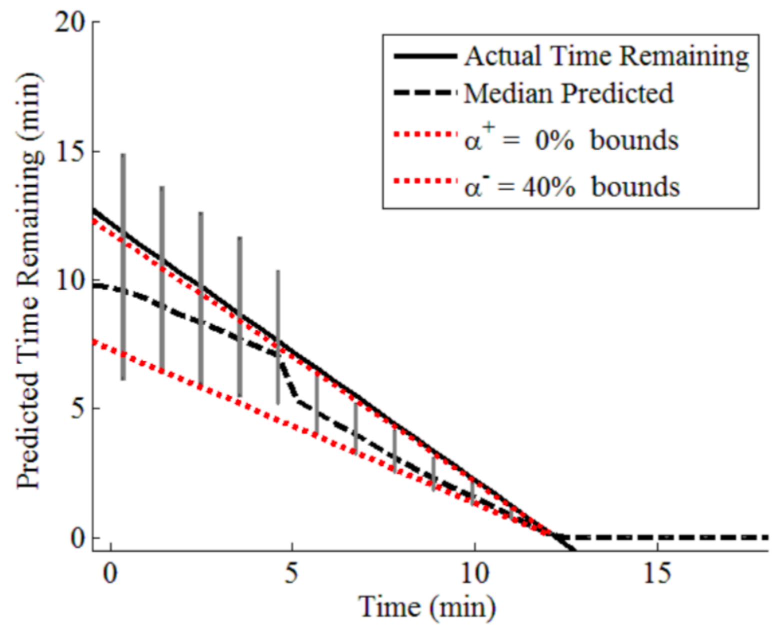



# Results: Edge

- Use UKF for state estimation with electrochemistry model



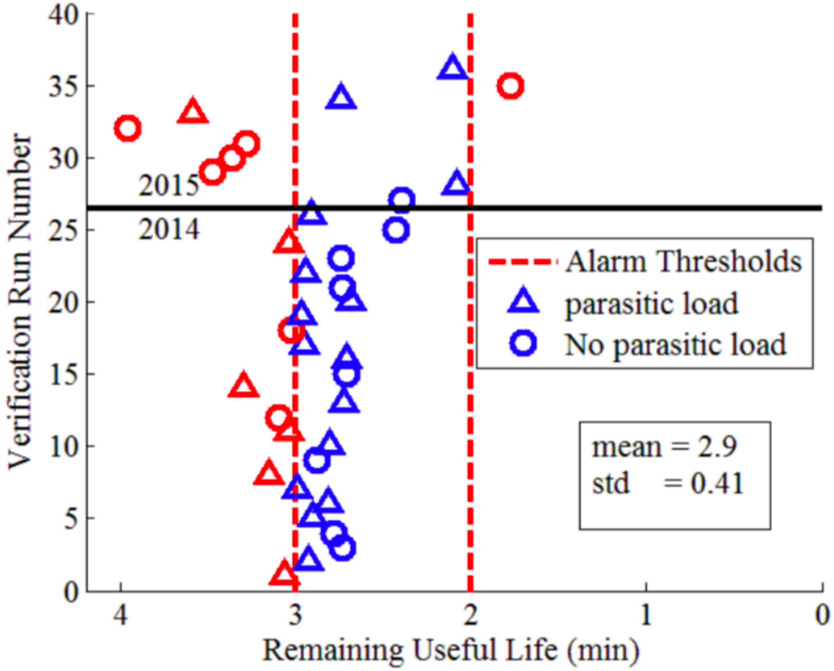

Example plot of measured and predicted battery current (top) and voltage (bottom) shown at three sample times over a trial battery discharge run



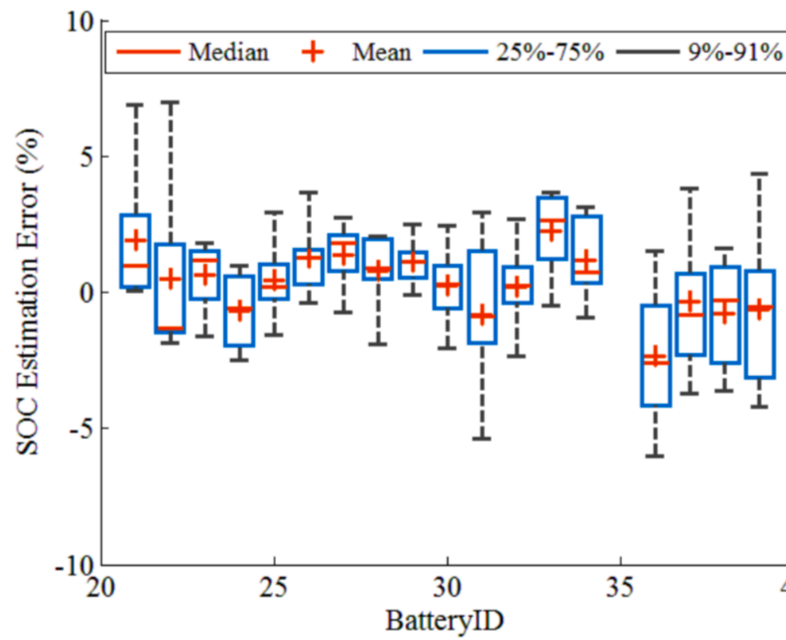

Predicted remaining flying time

# Results: Edge






Two-minute alarms for additional runs done a year later using revised battery capacity parameters.



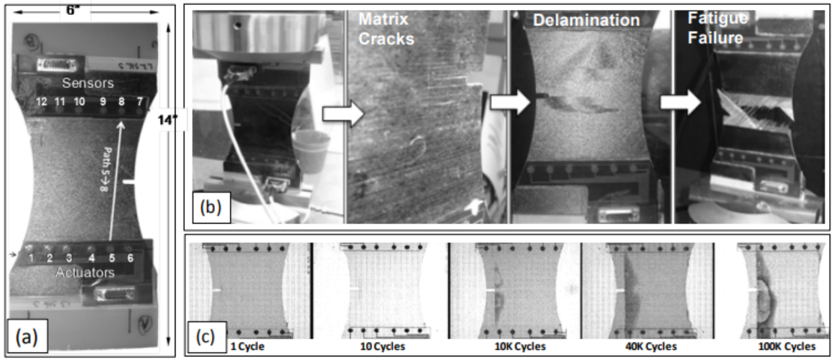

SOC estimation error from 10 additional verification runs in 2015 (36 runs that each use 4 batteries)

# NDE Analysis and Prognostics

# NDE/SHM of Composites

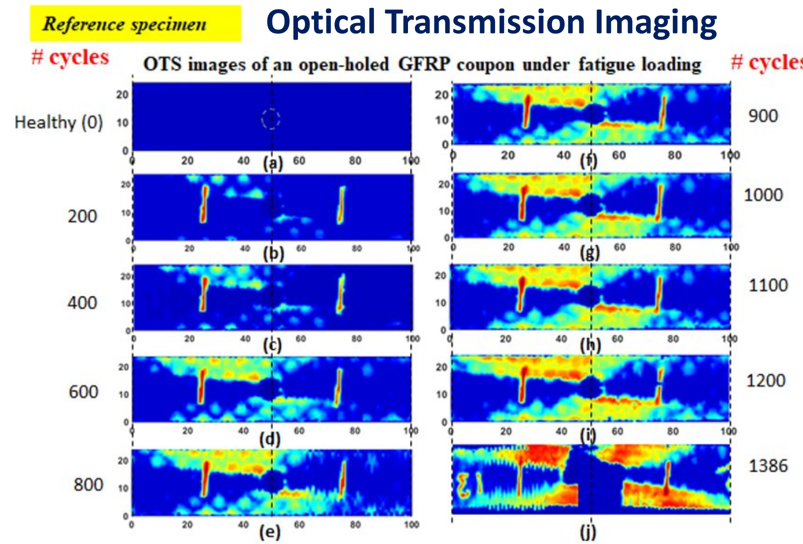
Ames Research Center

## X-ray Imaging

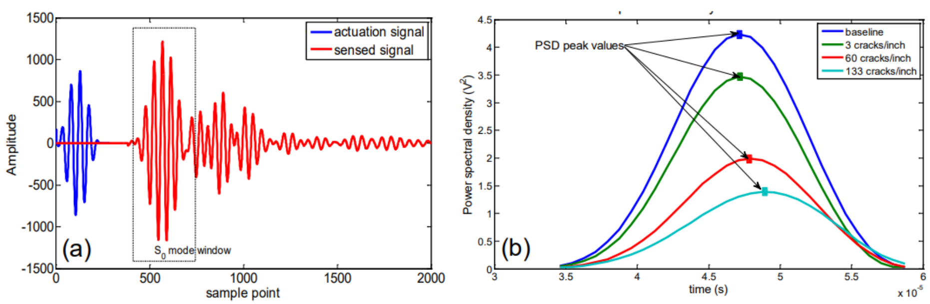



(a) Coupon specimen, SMART Layers location, and diagnostic path from actuator 5 to sensor 8. (b) Development of matrix cracks and delamination leading to fatigue failure. (c) Growth in delamination area in X-ray images.

## Optical Transmission Imaging

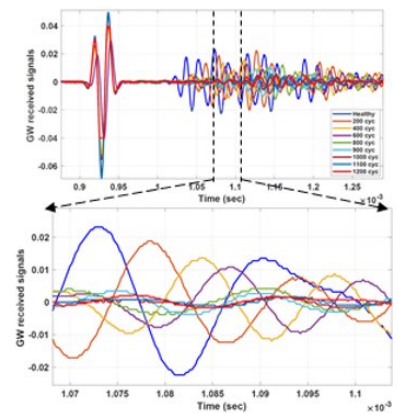



## Guided Wave Sensing




a) Isolating the first S0 mode by windowing the sensed signal. (b) Change in Power Spectral Density curves with increasing matrix crack density

## Guided Wave Sensing




Change in TOF with increasing delamination.

*Data extracted from Stanford SACL's fatigue tests on dog-bone CFRP specimens.

**Thanks to Prof. Fu-Kuo Chang, Dr. Cecilia Larrosa.**

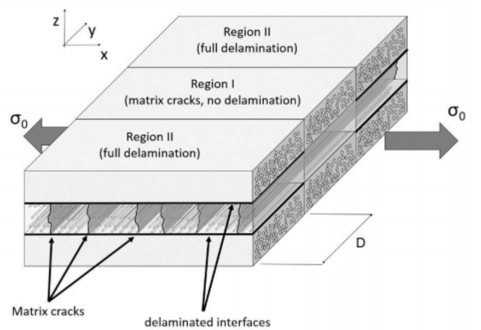
*Data extracted from Michigan State University NDE Lab's fatigue tests on notched GFRP specimens.

**Thanks to Prof. Yiming Deng, Prof. Mahmoodul Haq and Prof. Lalita Udpa.**

# Damage growth prediction in composites



- Investigate simple (yet robust) damage accumulation models for fiber-reinforced polymers that can be adopted in model-based prognostics.




**Zhang's model for a partially-delaminated cross-ply laminate**

**SERR and growth rates**

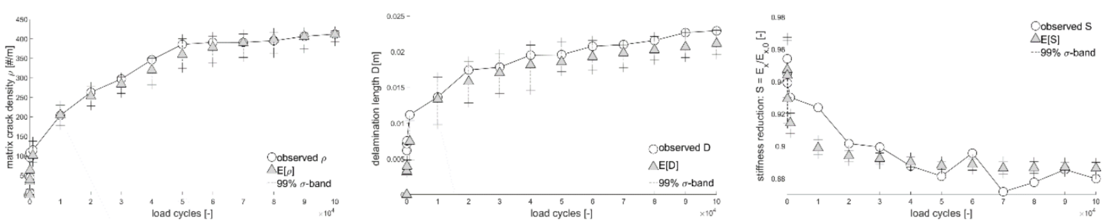
$$G = -\frac{\partial U}{\partial A} = f(\rho, D_x, D_y); \qquad \frac{d\rho}{dN} = f(G), \frac{dD_x}{dN} = f(G), \frac{dD_y}{dN} = f(G)$$

**State-space formulation**

$$\boldsymbol{x}_k = \begin{bmatrix} \rho_k \\ D_{y,k} \\ S_k \end{bmatrix} = \begin{bmatrix} \rho_k = \rho_{k-1} + \frac{d\rho}{dN}(\boldsymbol{\theta})\Big|_{k-1} e^{\omega_{\rho,k}} \\ D_{y,k} = D_{y,k-1} + \frac{dD_y}{dN}(\boldsymbol{\theta})\Big|_{k-1} e^{\omega_{D_y,k}} \\ S_k = \frac{E_{x,k}(\rho_k, D_{y,k})}{E_{x,0}} + \omega_{S,k} \end{bmatrix}$$

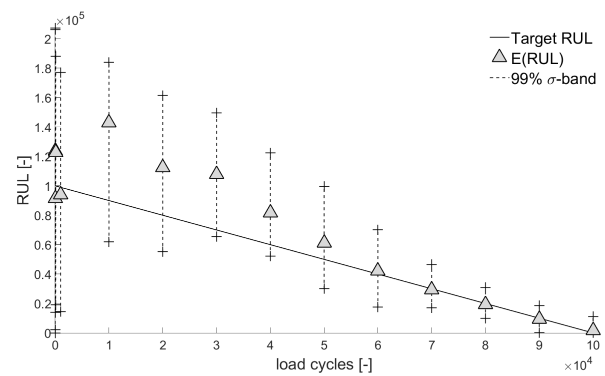
$$\boldsymbol{z}_k = \begin{bmatrix} \hat{\rho}_k \\ \hat{D}_{y,k} \\ \hat{S}_k \end{bmatrix} = \begin{bmatrix} \rho_k + \eta_{\rho,k} \\ D_{y,k} + \eta_{D_y,k} \\ S_k + \eta_{S,k} \end{bmatrix}$$

- Application of Bayesian filtering to fatigue damage progression




(a) (b) (c)

**Posterior estimation of the damage growth against load cycles; matrix crack density (a), delamination (b) and normalized stiffness (c)**




**RUL Prediction**

# Conclusions

Ames Research Center

- Focus on model-based approaches for system  state estimation and prediction
- Hybrid approaches
- Validate models and algorithms with data from lab experiments and fielded systems
- Future work involves
  - Thermal models
  - Higher fidelity models
  - More efficient algorithms