James Hofmeister, M.S.

Sr. Principal Research Engineer

James Hofmeister is a Senior Principal Research Engineer at Ridgetop Group. He has more than 30 years of experience and has provided Ridgetop with a wide range of engineering expertise over numerous projects in many roles, from engineer to lead design engineer, principal investigator, and engineering manager. His engineering innovation is demonstrated by his seven patent filings for Ridgetop since 2005 (four U.S. issued, two finals pending approval, one provisional) and his three U.S. issued patents for IBM.

ATTF™ - ARULE*

Advanced Time-to-Failure

Adaptive Remaining Useful Life Estimation

by

James (Hoffy) Hofmeister

April 25, 2012

* Patent Pending, Information only for Education and Planning Purposes

Agenda

- ARULE™ Introduction
- Reliability Modeling & Curve Fitting
- ARULE Modeling & Data Adaptation
- Examples
- Application Programming Interface
- Conclusion
- Questions and Answers

ARULE Introduction

- Algorithm to determine Remaining Useful Life (RUL) Estimates on complex systems from sensor data
- Application Programming Interface (API) – Used by ATTF
- Innovative, simple but powerful
  - Adapts model as each data point is processed
  - Uses adapted model as the requisite memory
  - Adapted model is used to make RUL estimates
  - Data comes from fault-to-failure progression (FFP) signatures
- Versatile application use
  - Electronic and physical fatigue damage
  - Fault-to-failure progression (FFP) signatures
- Fast, accurate RUL (time-to-failure) estimates
- State-of-health (SoH) estimation

ARULE Introduction

- Language
  - MATLAB, Java, C, Dynamic Link Library (DLL) executables
    - December 2009
    - AMRDEC, helicopter power supply
  - Sentinel Network Integration (DLL)
    - 1Q 2010
    - AMRDEC, helicopter power supply
- Deliverables - Demonstrations
  - Raytheon *Tewksbury, MA*
  - AMRDEC *Huntsville, AL*
  - BAE Systems *New York*
  - Boeing *Philadelphia*
  - Rolls Royce, AEC (Aero Engine Controls)

Modeling

- Weibull
  - Fatigue-related damage or wear-out
  - Environment and usage
    - Temperature
    - Humidity
    - Shock
    - Vibration
    - Expansion-contraction (coefficient of thermal expansion differences)
    - Voltage
    - Current
  - Environment and usage models
    - Coffin-Manson

Modeling

- Distribution and hazard rate models
  - Probability Distribution Function (PDF)

$$(\qquad) \quad -(-) \qquad (-)$$

  - Cumulative Distribution Function (CDF)

$$(\quad)$$

  - Hazard (failure) Rate

$$(\qquad) \quad -(-)$$

- Lifetime modeling
  - *x* becomes t (time), λ (lambda) = scale (mean life), *k* = shape (Q of the curve)

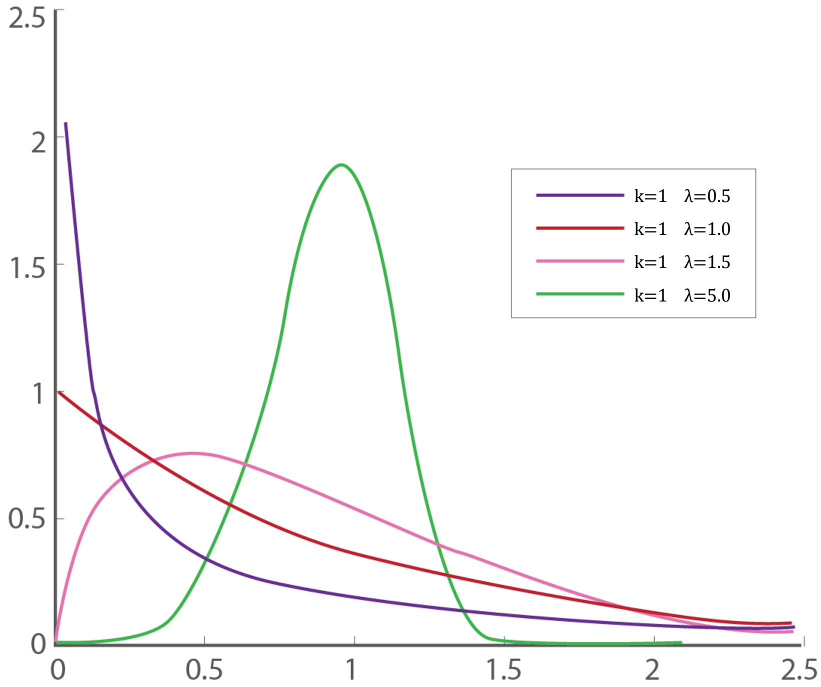
Modeling

- Models are fitted to the data
  - Represents the “mean” of a set of like components, …, systems
  - Well-defined & well-structured environment and test conditions
- Condition-Based Maintenance (CBM)
  - Requires continuous data fitting, which can be compute-intensive
  - Failure progression is often not well-behaved
  - Large variance(s) leading to large uncertainty in predictions
- Prognostic algorithms
  - Reliability model-based
  - Environment-based
  - Fault/failure signature-based

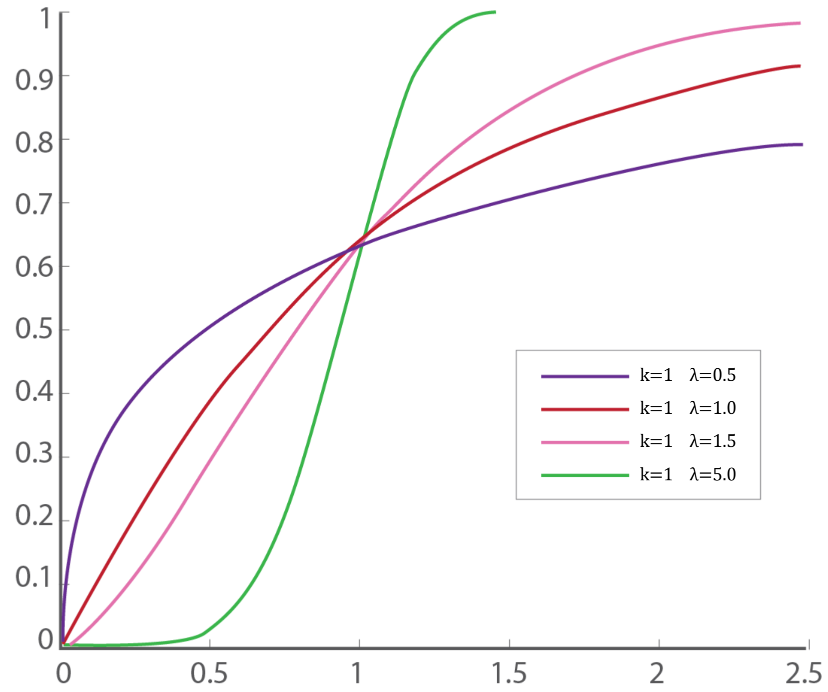
Modeling

Examples

PDFs – Lambda = 1



CDFs – Lambda = 1



Reliability-Statistical Modeling

- Start with hazard rate

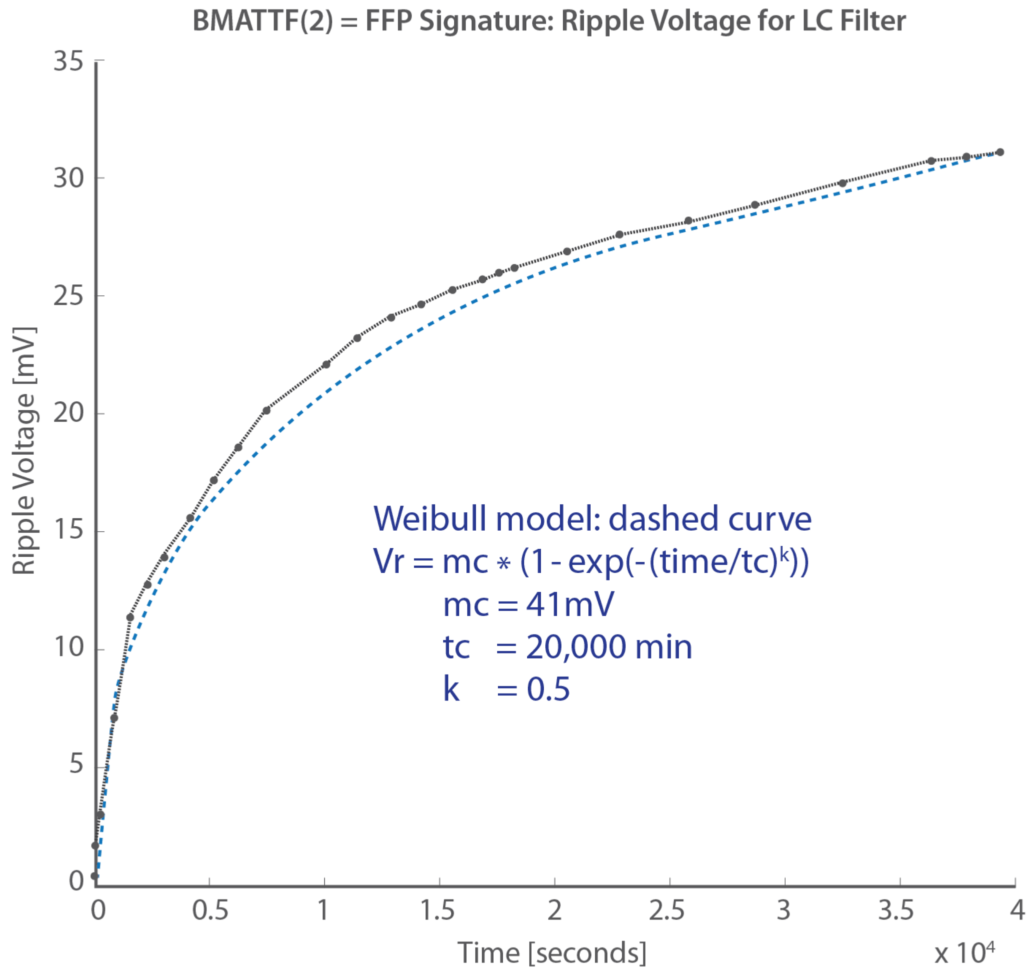
$$h(x;k,\lambda) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}$$

- Run LT, ALT, and HALT regimes (highly accelerated life test)
- "Guess" at lambda relationship to HALT cycles
  - Estimate # of door open/closes per day -> week -> month -> year
  - If mean failing HALT = 30000 and estimate 3000 cycles per year
  - Then reliability (mean time to failure) = 10 year
- Issues with such modeling
  - HALT cycle translation to real time may or may not be "accurate"
  - Mean values are used – may or may not apply to a particular unit
  - HALT regime rarely replicates actual use & stress in real life
  - HALT does not account for random events that initiate onset of damage and so on

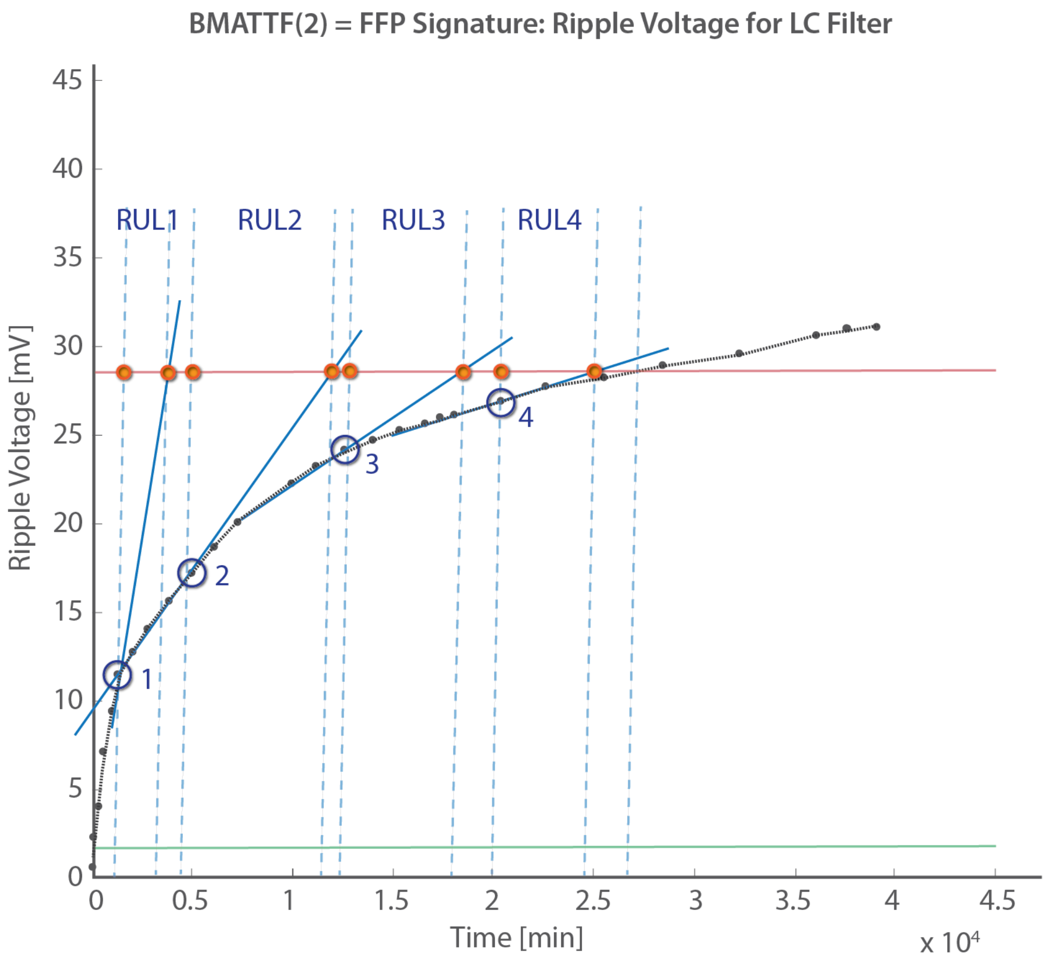
Curve Fitting

- Fit data to a curve model:
  - Linear
  - Non-linear
  - Mean square
  - Others
- Challenges
  - Data intensive
  - Computationally intractable, in practice usually very inaccurate: 25 to 50% under/over estimation

Curve Fitting Example – Real Data

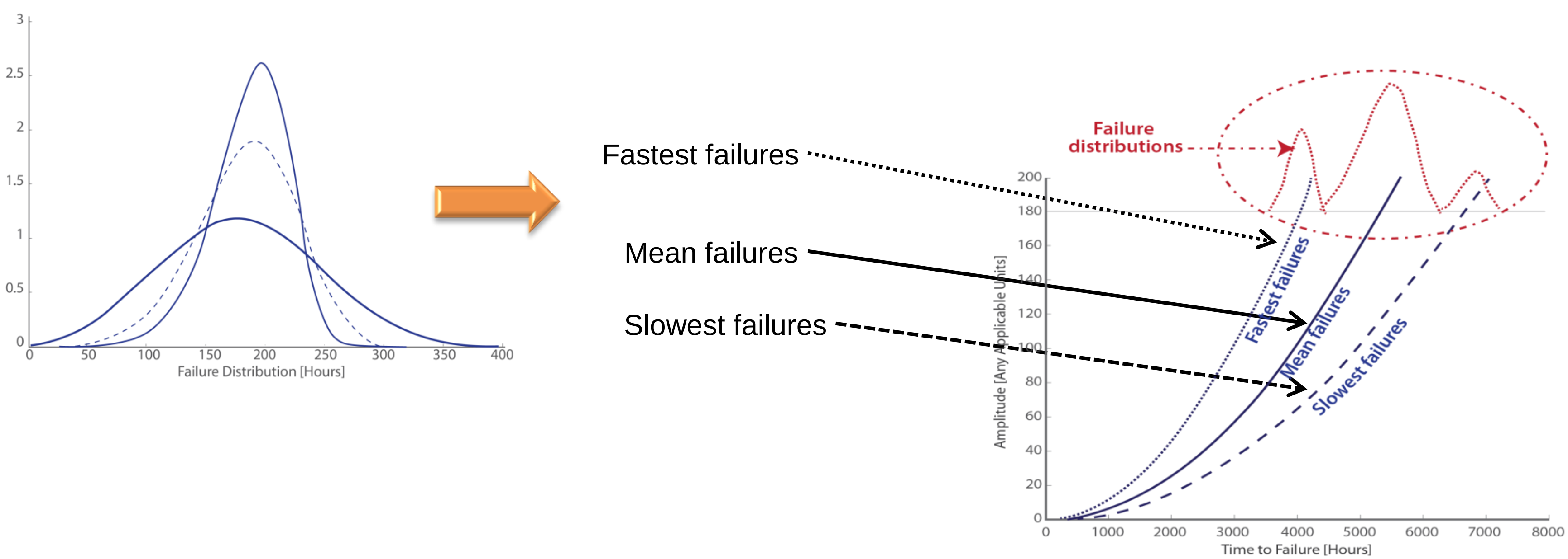


Curve Fitting Example – Curve Fit Results



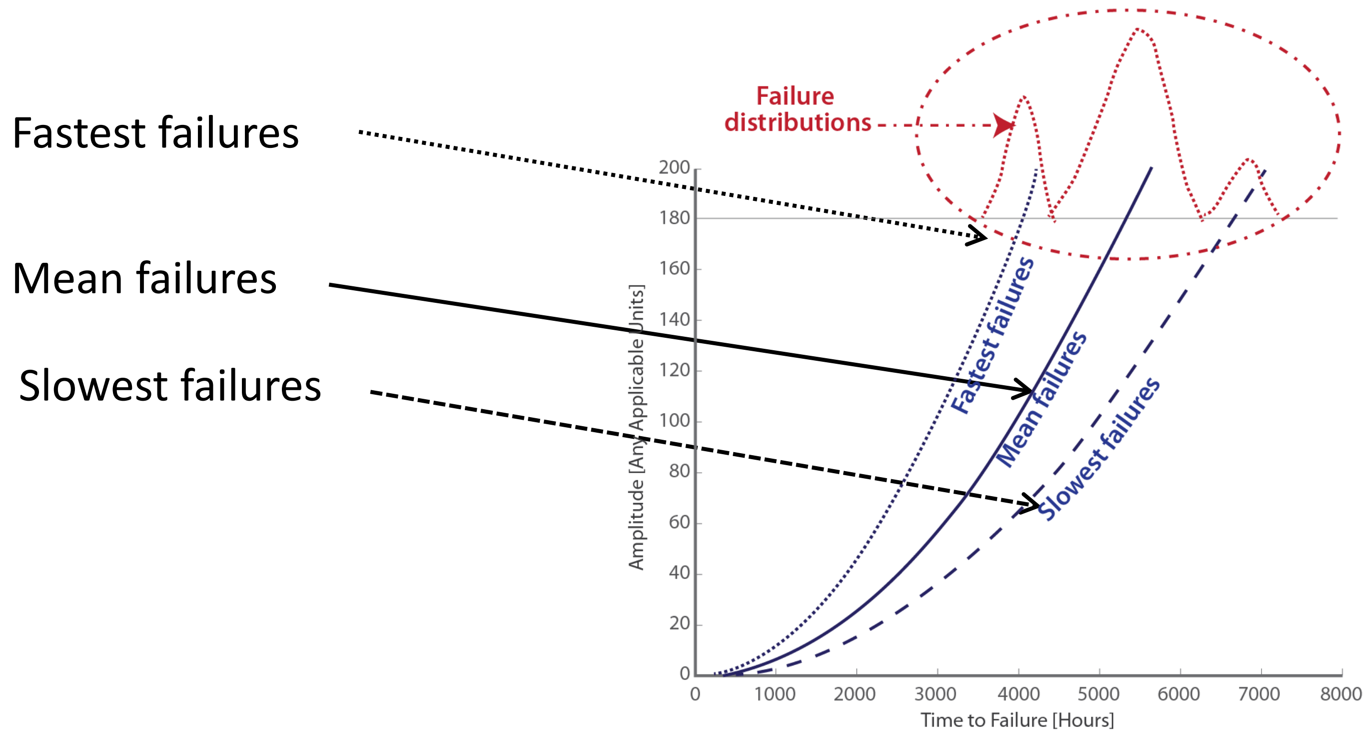
Multiple Sets of Data: Family of Curves

- Family of Weibull PDF Curves
  - Lambda = 200 hours
  - K (Q) = 3, 5, 7 (highest amplitude)
  - Each curve could, for example, be representative of the failure distribution of components about an approximate Mean Time to Failure (MTTF) = 200 hours

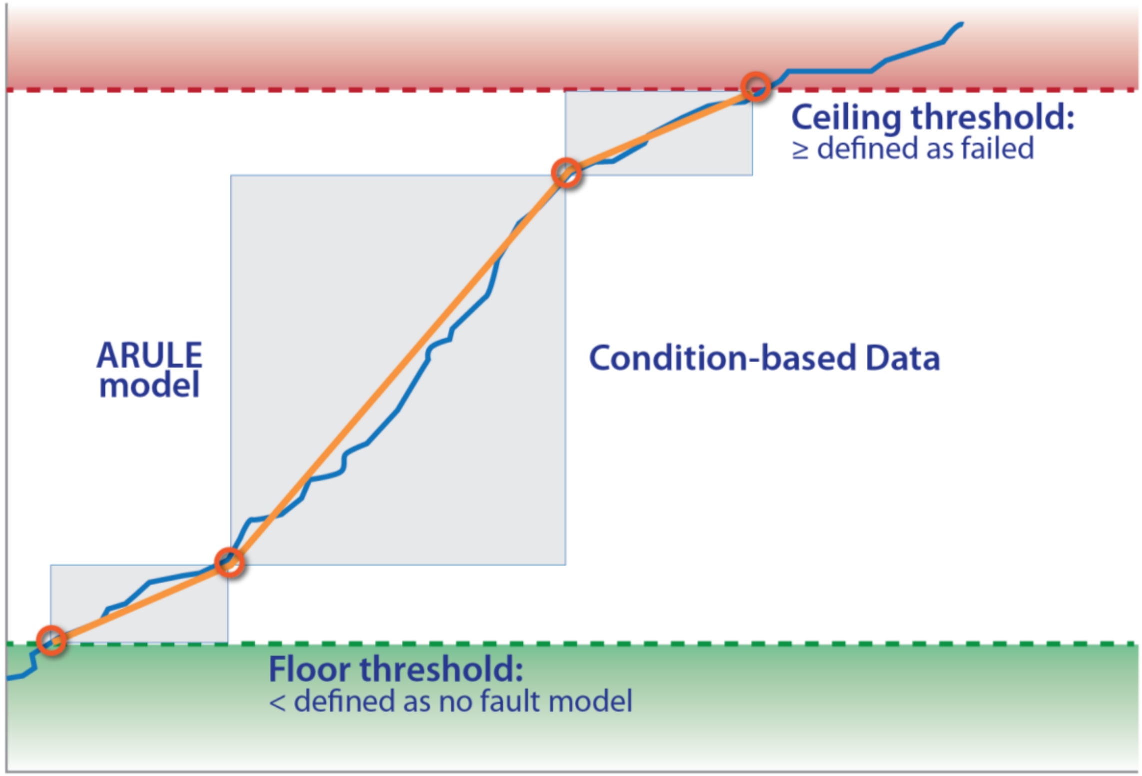


Family of Curves: Failure Distribution

- Family of FFP signatures
  - Environment & usage variances
  - Manufacturing and material variances

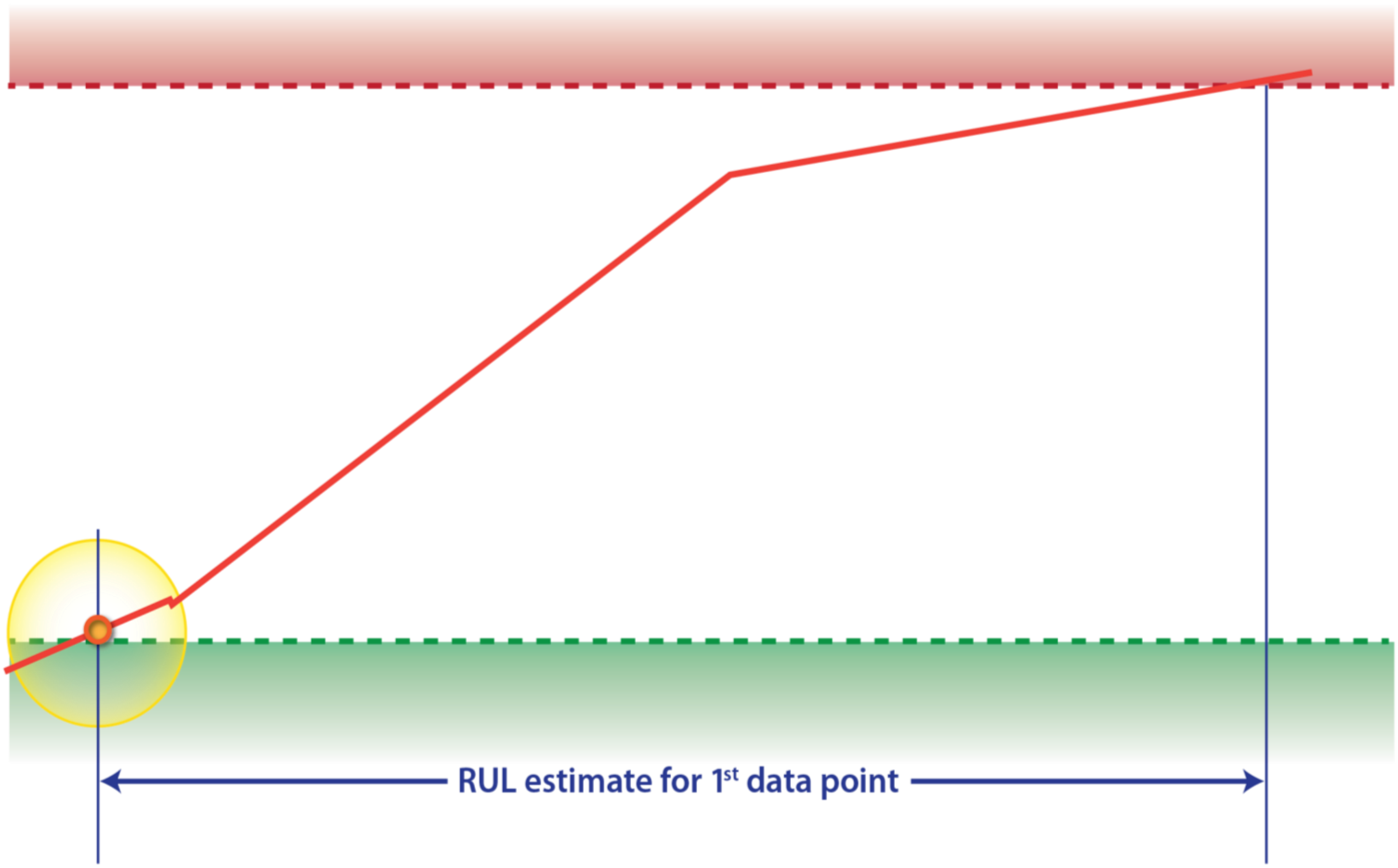


ARULE Model

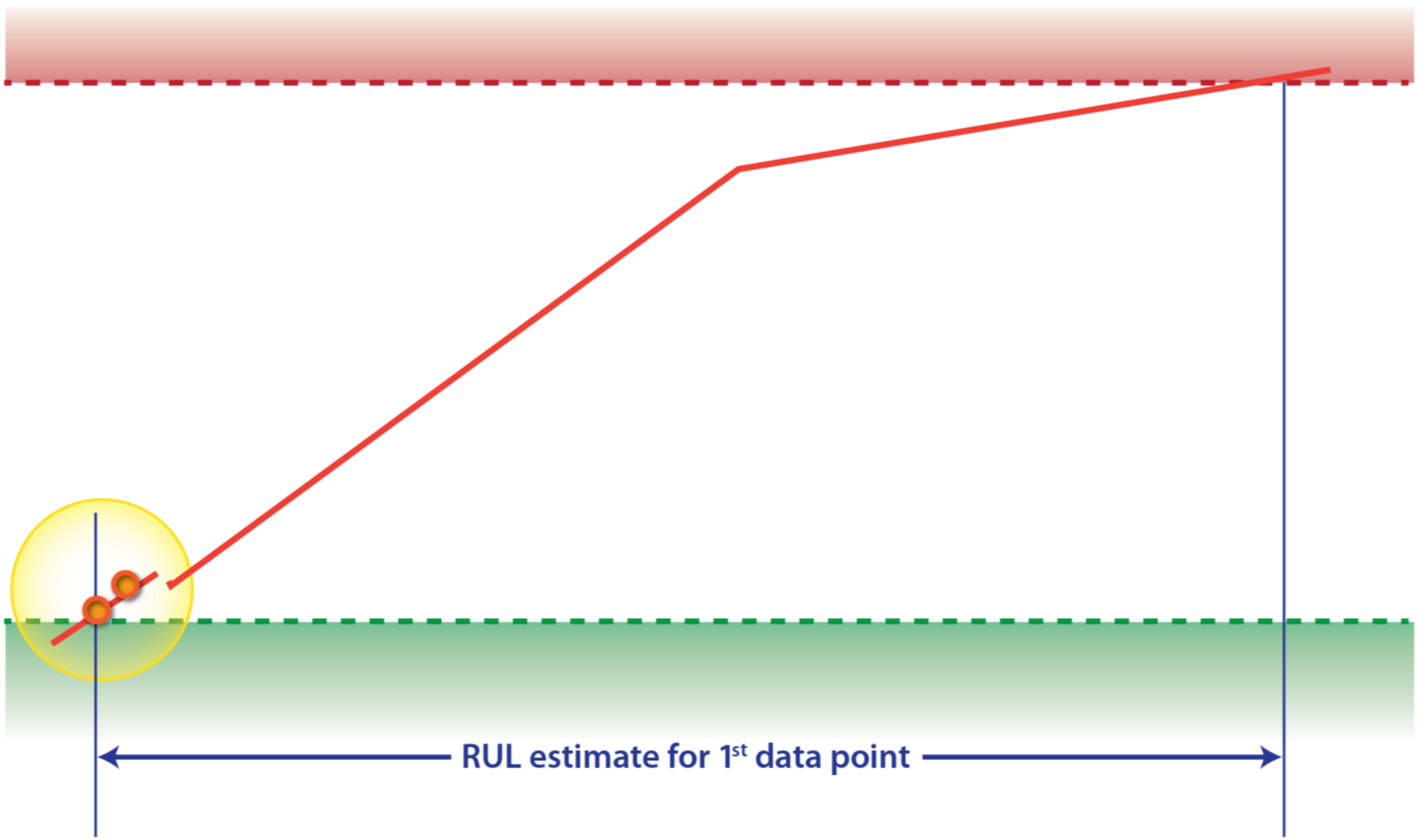


Ceiling threshold:
≥ defined as failed
ARULE model
Condition-based Data
Floor threshold:
< defined as no fault model

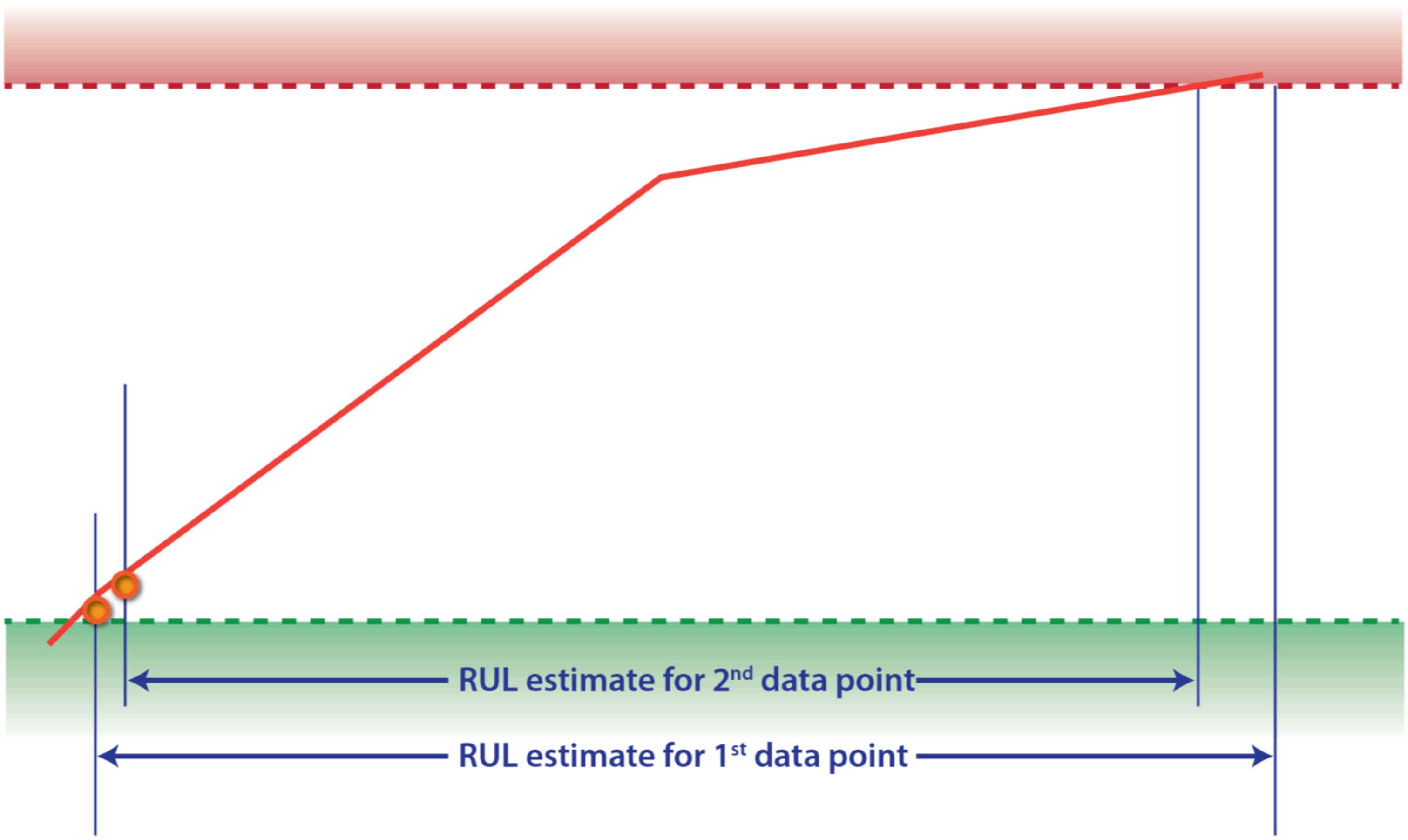
Adapting Model to Data, RUL Estimation



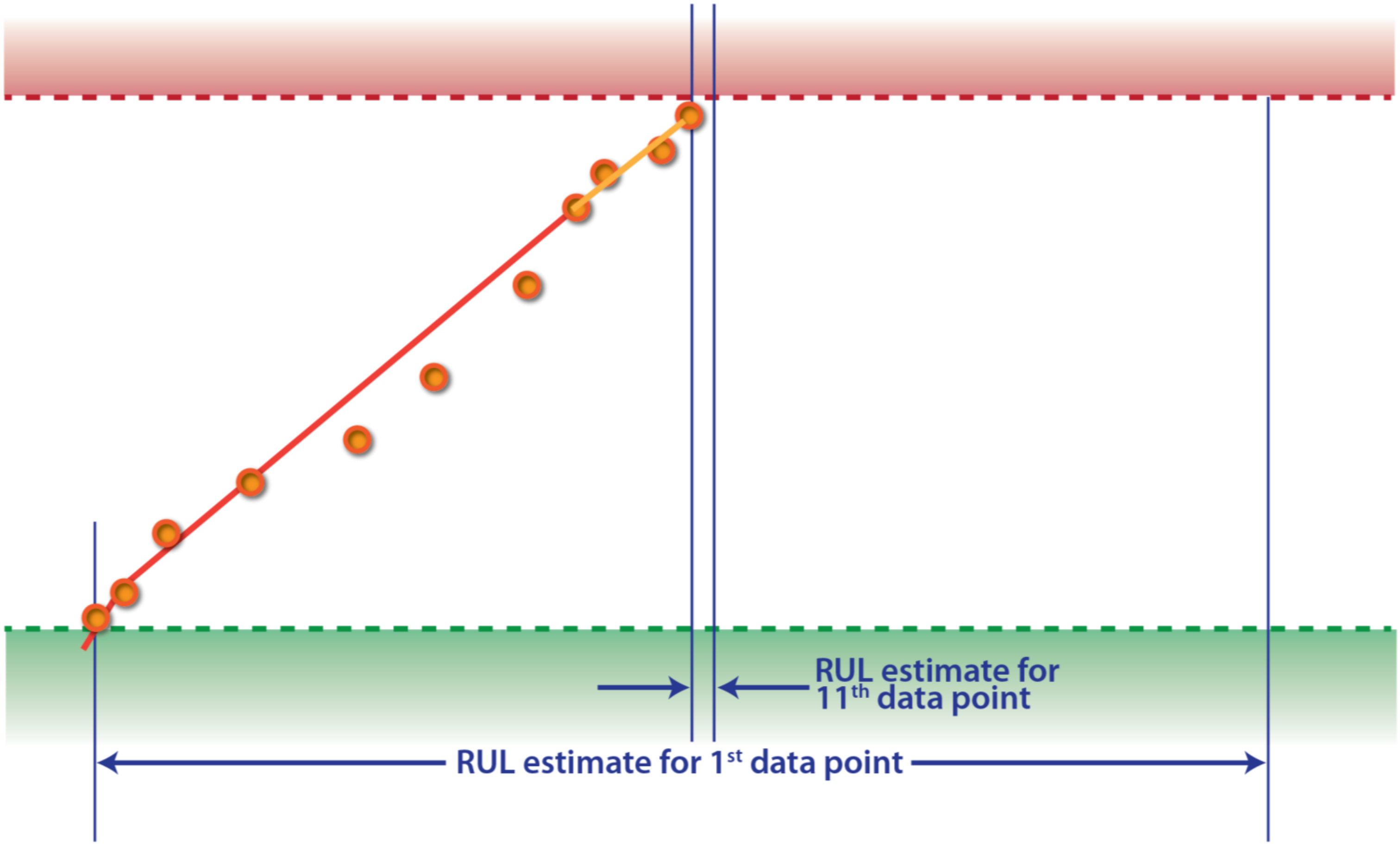
Adapting Model to Data, RUL Estimation



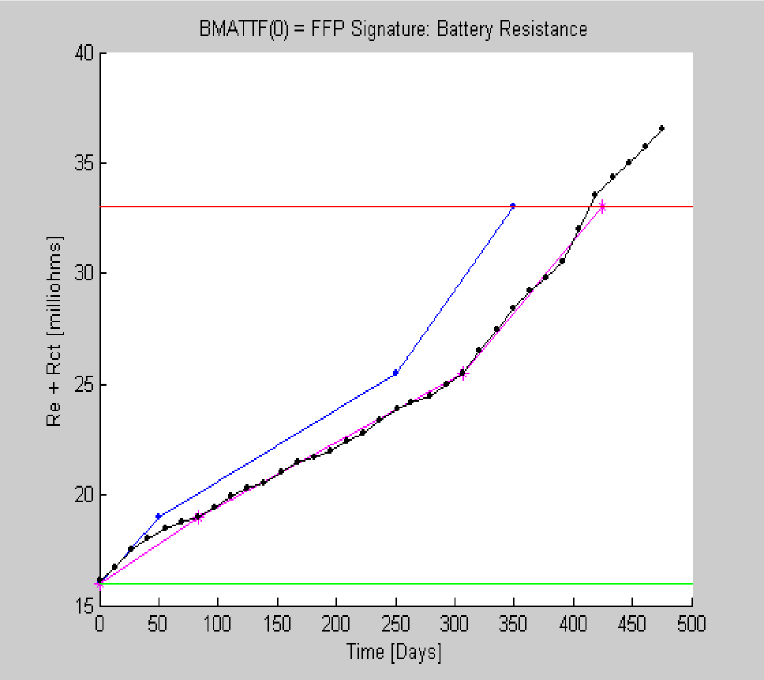
Adapting Model to Data, RUL Estimation

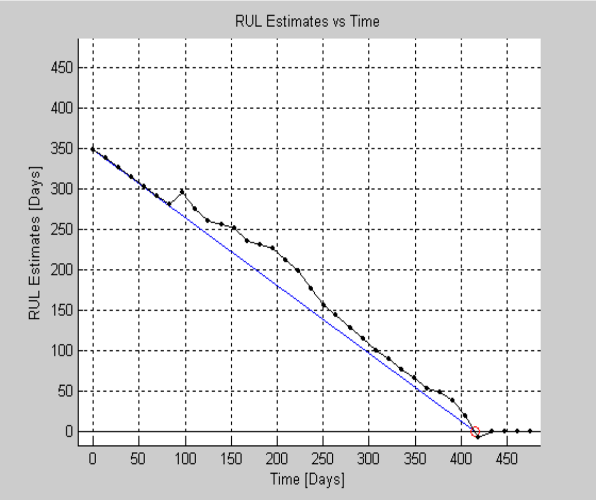


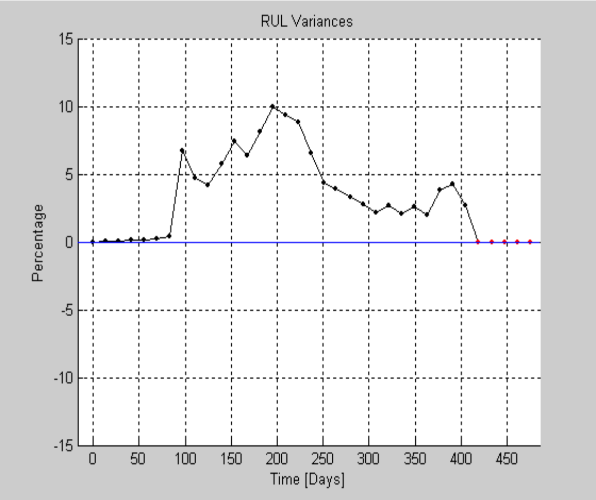
Adapting Model to Data, RUL Estimation



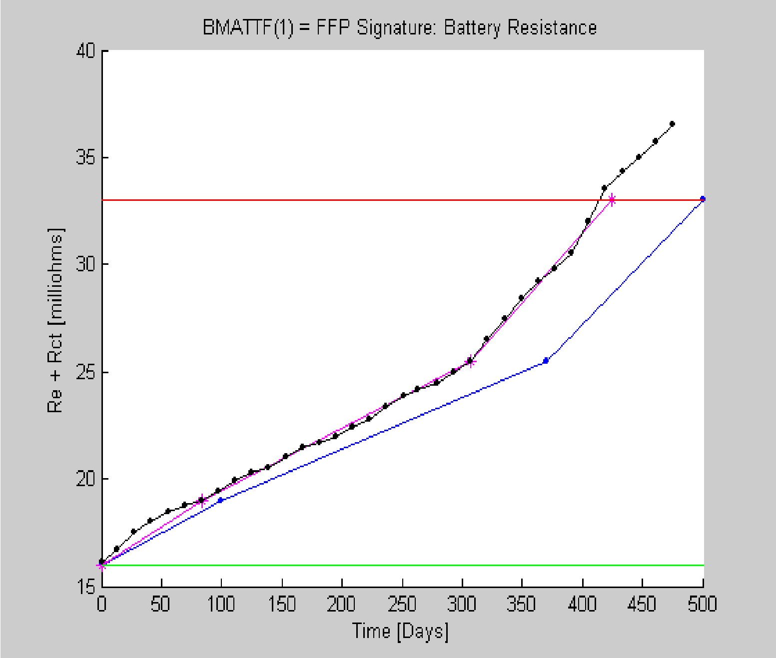
Battery Resistance: Early Prediction Model

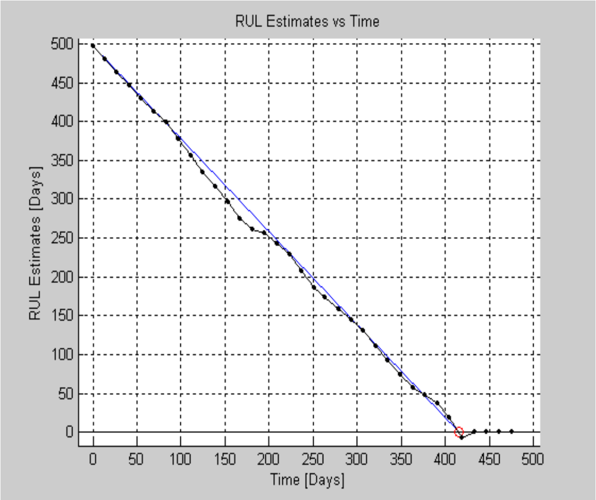


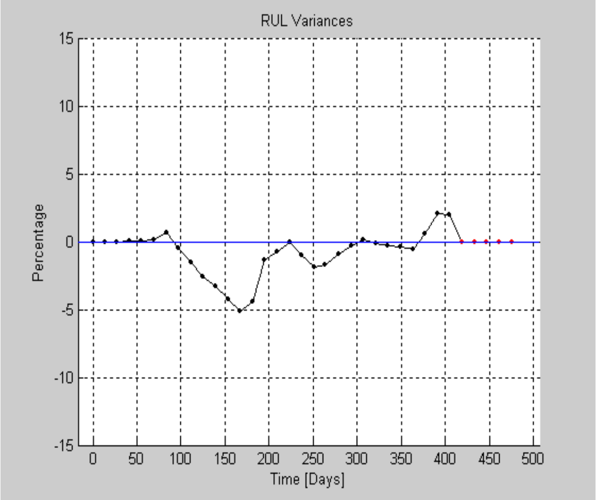




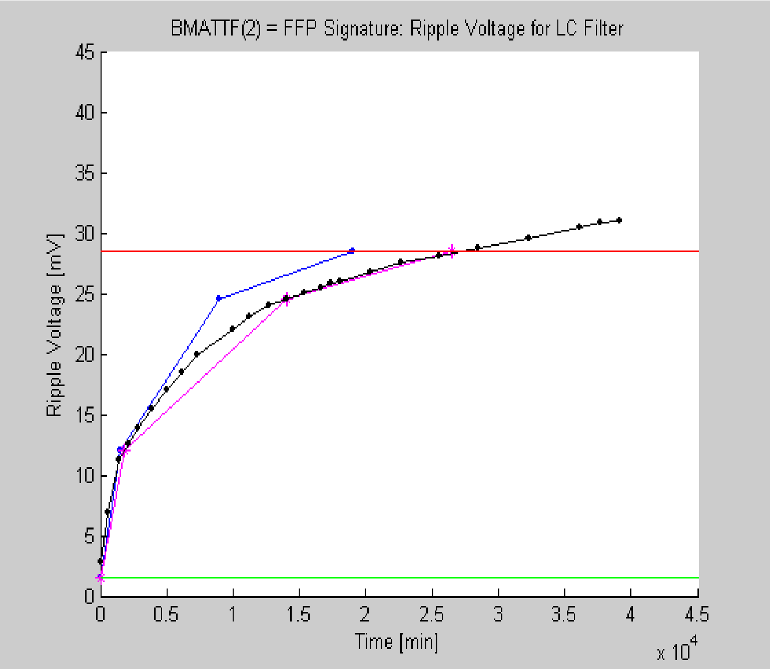
Battery Resistance: Late Prediction Model

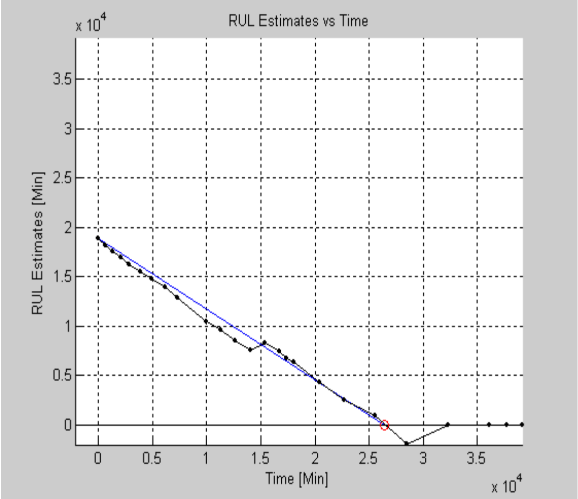


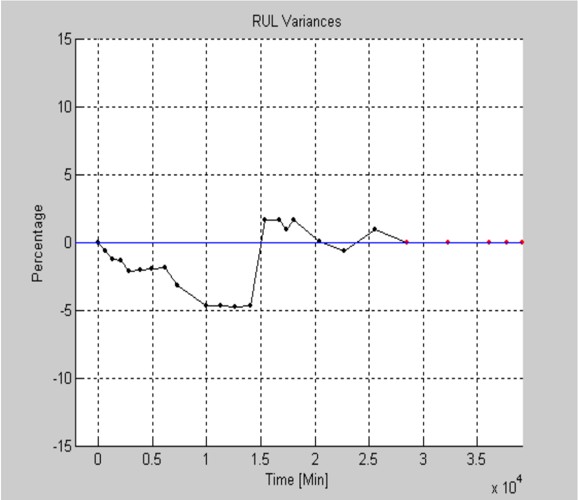




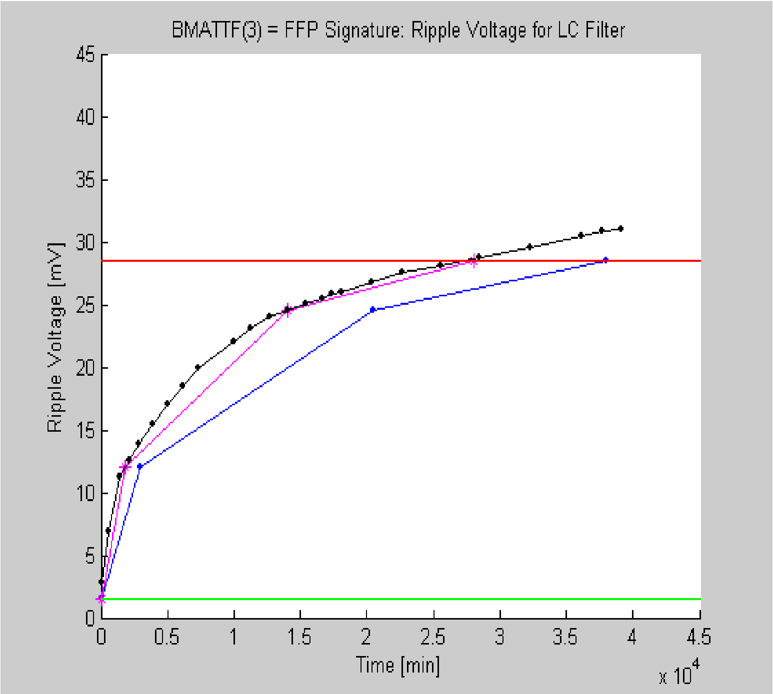
Ripple Voltage: Early Prediction Model

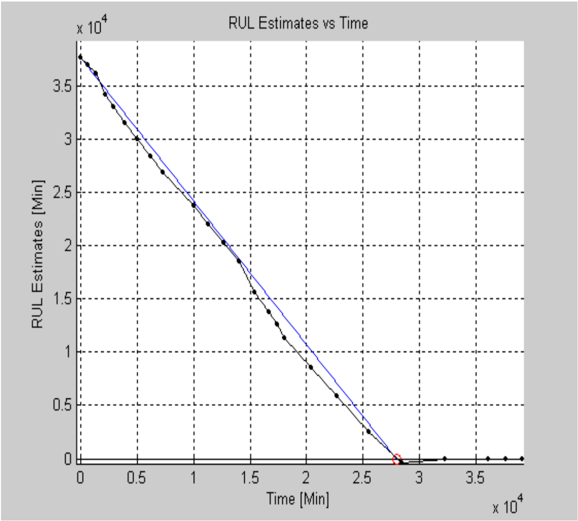


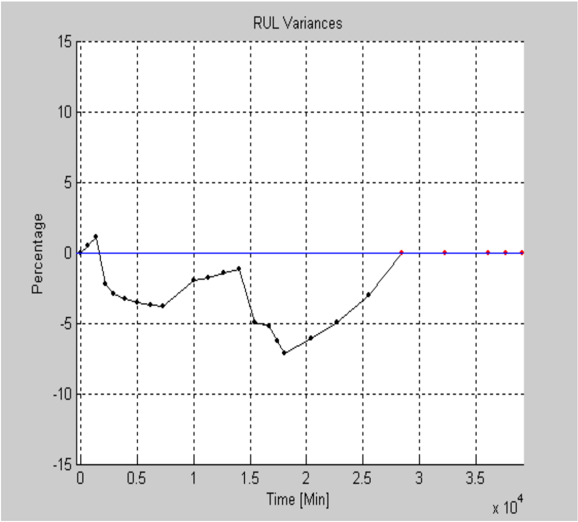




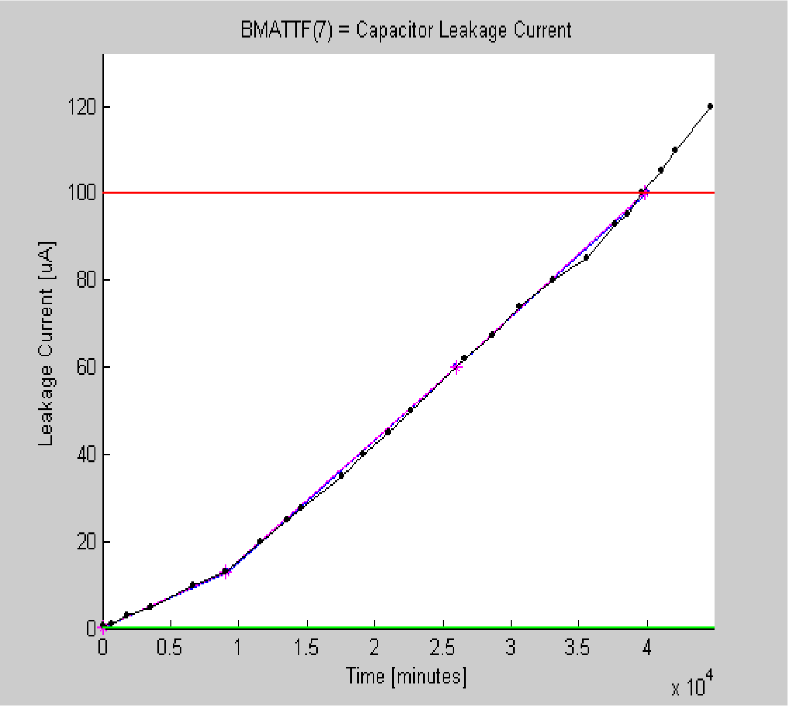
Ripple Voltage: Late Prediction Model

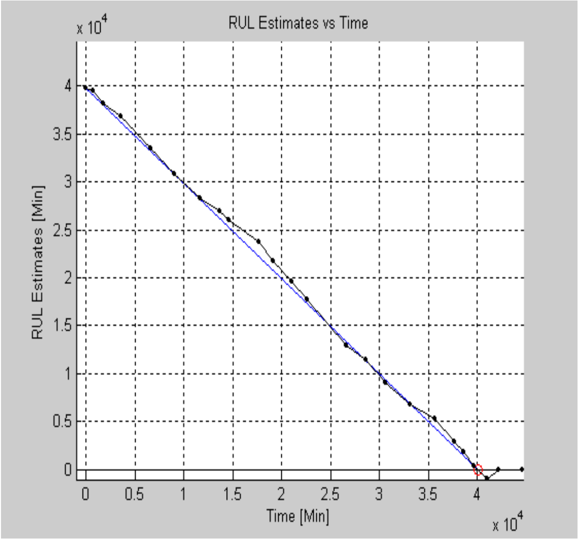


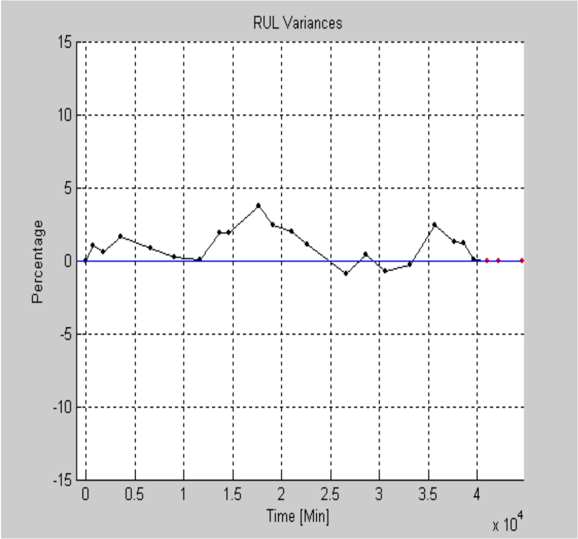




Leakage Current

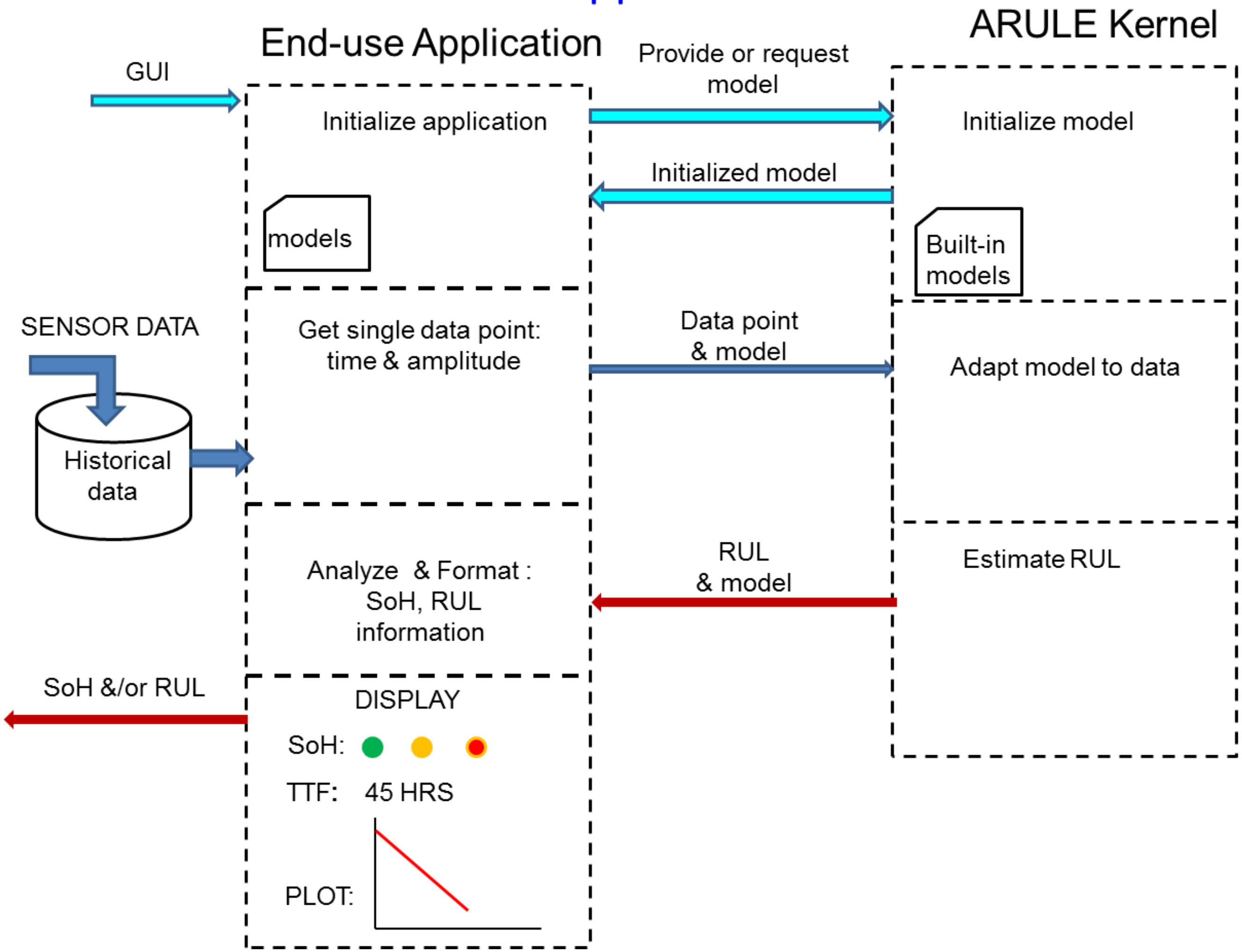






Architecture Block Diagram

ARULE Application – ATTF Kernel



Application Programming Interface: Model Header

```
%******************************************************************
% model for battery health: Re + Rc [milliOhms]
%******************************************************************
FT      = 'FFP';             % Fault-to-Failure Progression
%***define the horizontal lengths (time) **************************
t1      = 50.0        ;      % end t of 1st box 10
t2      = 200.0       ;      % end t of 2nd box 12
t3      = 100.0       ;      % end t of 3rd box 14
%***define the vertical heights (amplitudes)***********************
a0      = 16    ;            % no damage if below this level
a1      = 3.0   ;            % height a of end of 1st box
a2      = 6.5   ;            % height a of end of 2nd box
a3      = 7.5   ;            % height a of end of 3rd box
%***define the equation type*************************************   V5
eq1     = 0     ;            % 0 : straight line                    V5
eq2     = 0     ;            % 1 : A = Ao*[1 - exp (-(t/T))]        V5
eq3     = 0     ;            % 2 : A = Ao*[exp ((t/T) - 1]          V5
                             % 3 : undefined                        V5
%***define the exponential amplitude coefficients(must be > an value V5
ac1     = 0     ;            % must be > a1                         V5
ac2     = 0     ;            % must be > a2                         V5
ac3     = 0     ;            % must be > a3                         V5
%******************************************************************
```

API: Build a Model

```
%**************************************************************************
%   Build the ATTF model header
%**************************************************************************
MODDEF=struct('FFPNAME',{MNAME},'FT',{FT},'a0',a0, 't1',t1,'a1',a1, ...
        't2',t2,'a2',a2,  't3',t3,'a3',a3, ...
        'eq1',eq1,'eq2',eq2,'eq3',eq3, 'ac1',ac1,'ac2',ac2,'ac3',ac3 );
%**************************************************************************
% instead of identical models with different floor thresholds, let user
% change the floor
%**************************************************************************
[ newa0 ] = BMQUFLT(a0) % get the default or user-specified floor threshold

MODDEF.a0 = newa0 ;     % change a0 value

MODE    = 0 ;    % 0 = init, 1 = process data points
%**************************************************************************
%   call ATTFPGM to build model definition trailer
%**************************************************************************
[RC RS MODEL ] = ATTFPGM(MODE,MNAME,MODDEF) ;    %  initialize model

if RC == 0
    MODE = 1;                     % process data
    INMOD = struct(MODEL) ;       % get the model structure
else
    display(MNAME)
    error('INVALID MODEL HEADER - COULD NOT INITIALIZE MODEL')
end
OMOD = INMOD ;                % save original model
```

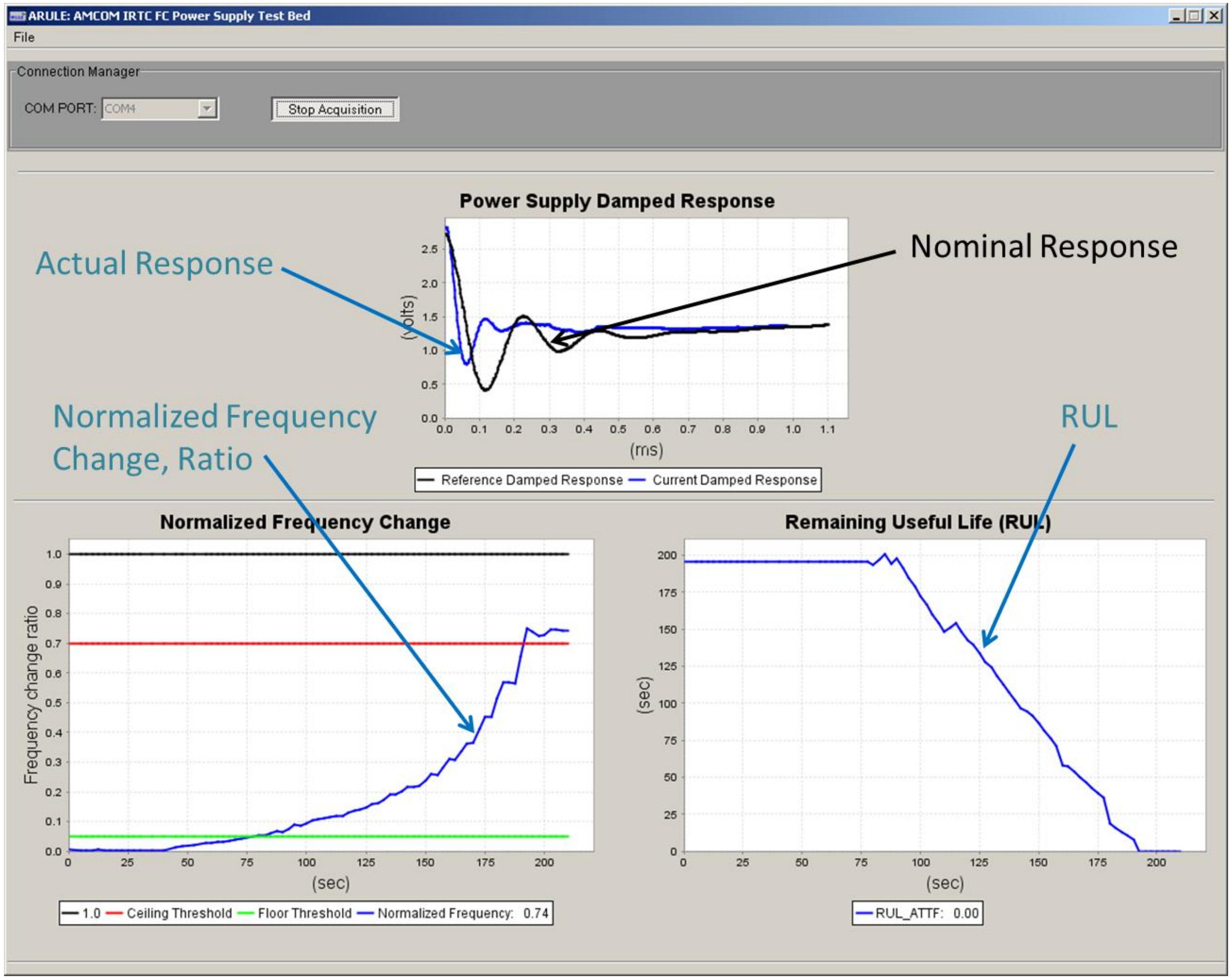
API: Call ATTF: Initialize the Model, Data -> RUL

```
MODE    = 0 ;   % 0 = init, 1 = process data points
%**************************************************************************
%   call ATTFPGM to build model definition trailer
%**************************************************************************
[RC RS MODEL ] = ATTFPGM(MODE,MNAME,MODDEF) ;   %  initialize model

if RC == 0
    MODE = 1;                     % process data
    INMOD = struct(MODEL) ;       % get the model structure
else
    display(MNAME)
    error('INVALID MODEL HEADER - COULD NOT INITIALIZE MODEL')
end
OMOD = INMOD ;                % save original model
%*********************************************************************
%   place a data point into the data space
%*********************************************************************
DATA = [dt da] ;      % build data array
[RC RS MODEL RUL] = ATTFPGM(MODE,MNAME,INMOD,DATA) ;  % call ATTF

INMOD = MODEL ;               % get updated model for next data pt
```

RingDown™ and ARULE



ARULE Specifications

- ARULE kernel is independent of units of measure
- Processing Speed (after program initialization)
  - Receive data
  - Adapt model to the data
  - Calculate RUL
  - Return RUL and updated model to caller
    - → ***Processing time:*** 250 microseconds average per data point
    - → ***Hardware:*** Laptop, Dual Core 64x2, 1.79 GHz, 1.87 GB RAM

Conclusion

- Adaptive RUL Estimator
  - ARULE kernel w/API
  - ARULE application using API
- Accurate processing
  - Even with large sample periods
  - Handles both increasing and decreasing degradation rates
  - Adapts model to received data
  - Uses adapted model to calculate time-to-failure (RUL estimates)
  - Fast processing (250 microsecond average per data point)
- Very flexible
- Minimal model definition

Questions?

Upcoming Webinars

Ridgetop Group Inc
ENGINEERING INNOVATION

Topic	Date	Time
IC Characterization with ProChek, a Compact Benchtop System	Wed. May 30, 2012	1:00 - 2:00 PM PDT
Implementation of Prognostics in Solar Applications	Wed. Jun 27, 2012	1:00 - 2:00 PM PDT
Troubleshooting Analysis and Decision Support in Complex Applications	Wed. Jul 25, 2012	1:00 - 2:00 PM PDT

For more information about Ridgetop Group Webinars, email us at information@ridgetopgroup.com

Thank You