LIB PHM implementation

2022-1 Digital Twin & Automation

17 박정우 | 17 유진수 | 17 홍세현

한동대학교
HANDONG GLOBAL UNIVERSITY

LIB PHM implementation


Journal of Power Sources 412 (2019) 442–450

Contents lists available at ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

A data-driven remaining capacity estimation approach for lithium-ion batteries based on charging health feature extraction

Peiyao Guo*, Ze Cheng, Lei Yang

School of Electrical and Information Engineering, Tianjin University, Tianjin, 300072, China

HIGHLIGHTS

- Health features extracted from charging voltage, current and temperature curves.
- Feature optimization based on grey relational and principal component analysis.
- Remaining capacity estiamtion with relevance vector machine.
- Validations with battery data in various operating conditions.

ARTICLE INFO

Keywords:
Lithium-ion battery
Health factor
Capacity estimation
Relevance vector machine

ABSTRACT

Capacity degradation monitoring of lithium batteries is necessary to ensure the reliability and safety of electric vehicles. However, capacity of cell is related to its complex internal physicochemical reactions and thermal effects and cannot be measured directly. A data-driven remaining capacity estimation approach for lithium-ion batteries based on charging health feature extraction is presented in this work. The proposed method utilizes rational analysis and principal component analysis to extract and optimize health features of charging stage which adapt to various working conditions of battery. The remaining capacity estimation is realized by relevance vector machine and validations of different working conditions are made with six battery data sets provided by NASA Prognostics Center of Excellence. The results show high efficiency and robustness of the proposed method.

1. Introduction

With growing energy and environment crisis worldwide, electric vehicles (EVs) technologies have received much attention and developed rapidly. Lithium-ion batteries have been widely used in EVs due to its merits of high operating voltage, high energy density, low self-discharge rate and no memory effect. Battery management system (BMS) plays a critical role in ensuring a long mileage, safety and reliability of EVs [1,2]. The accurate battery state estimation, as a key part of BMS, is desired to provide information for safety management and charging/discharging optimal control. Remaining useful life (RUL) is a measure of the change of the ability to store and release electrical energy of a battery compared with a fresh new one, essentially reflecting the aging and damage conditions of the battery [3,4]. It is necessary to replace batteries before the failure of battery make the whole system crash. Generally, the end of life (EoL) of battery reach when its actual capacity has decreased to 70% or 80% of its nominal value.

Capacity degradation of lithium-ion battery is closely related to its internal physicochemical reaction and thermal effects. Over the repeated charging/discharging cycles, side reactions occur between electrode and electrolyte continuously yielding the growth of solid electrolyte interface (SEI) with poor conductivity and the loss of cyclable lithium ion [5,6]. In addition, cycling cause morphological damage of electrodes (i.e., porosity decrease and particle crack) and active electrode material loss. Extreme operating conditions such as overcharging, overdischarging, high voltage, both low and high temperatures, would accelerate battery aging process [7,8].

Over recent years, extensive research on RUL and SOH estimation has been conducted. In general, the methodologies adopted can be divided into data-driven and model-based (e.g., equivalent circuit model (ECM), electrochemical model, etc.) methods. ECMs made up of various circuit elements neglect the complex internal physicochemical aging mechanisms and mimic the output dynamics of the battery. Allafi et al. [9] established a modified Wiener battery model, consisting of a linear ECM and a new static sigmoid block and the parameters of the model were identified at different temperatures and SOC. Fleischer et al.

* Corresponding author.
E-mail address: guopeiyao@tju.edu.cn (P. Guo).

https://doi.org/10.1016/j.jpowsour.2018.11.072
Received 30 August 2018; Received in revised form 25 October 2018; Accepted 22 November 2018

EV Interest Increase

↓

BMS Interest Increase

↓

RUL & SOH Interest Increase

LIB PHM implementation

NATIONAL AERONAUTICS
AND SPACE ADMINISTRATION

+ABOUT NASA | +LATEST NEWS | +MULTIMEDIA | +MISSIONS | +WORK FOR NASA

+ NASA Home
+ Ames Home
+ Intelligent Systems Division
+ Discovery and Systems Health

Data Repository

+ Home
+ Data Repository
+ Open Source Code
+ Publications
+ Roadmap
+ Awards

PCoE Datasets

Overview

The Prognostics Data Repository is a collection of data
universities, agencies, or companies. The data reposito
i.e., data sets that can be used for the development of p
series data from a prior nominal state to a failed state. T
ongoing process.

Publications making use of databases obtained from thi
the assistance received by using this repository and the
obtain the same data sets and replicate your experimer

Users employ the data at their own risk. Neither NASA
liability for the use of the data or any system developed

If you have suggestions concerning the repository send
christopher.a.teubert[at]nasa.gov

5. **Battery Data Set**
Publications using this data set

Description	Experiments on Li-Ion batteries. Charging and discharging at different temperatures. Records the impedance as the damage criterion. The data set was provided by the Prognostics CoE at NASA Ames.
Format	The set is in .mat format and has been zipped.
Datasets	+ Download Battery Data Set 1 (47758 downloads) + Download Battery Data Set 2 (20544 downloads) + Download Battery Data Set 3 (16807 downloads) + Download Battery Data Set 4 (13361 downloads) + Download Battery Data Set 5 (14138 downloads) + Download Battery Data Set 6 (14933 downloads)
Dataset Citation	B. Saha and K. Goebel (2007). "Battery Data Set", NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA

B0005	*1x1 struct*
B0006	*1x1 struct*
B0007	*1x1 struct*
B0018	*1x1 struct*
B0025	*1x1 struct*
B0026	*1x1 struct*
B0027	*1x1 struct*
B0028	*1x1 struct*
B0029	*1x1 struct*
B0030	*1x1 struct*
B0031	*1x1 struct*
B0032	*1x1 struct*
B0033	*1x1 struct*
B0034	*1x1 struct*
B0036	*1x1 struct*
B0038	*1x1 struct*
B0039	*1x1 struct*
B0040	*1x1 struct*
B0041	*1x1 struct*
B0042	*1x1 struct*
B0043	*1x1 struct*
B0044	*1x1 struct*
B0045	*1x1 struct*
B0046	*1x1 struct*
B0047	*1x1 struct*
B0048	*1x1 struct*
B0049	*1x1 struct*
B0050	*1x1 struct*
B0051	*1x1 struct*
B0052	*1x1 struct*
B0053	*1x1 struct*
B0054	*1x1 struct*
B0055	*1x1 struct*
B0056	*1x1 struct*

LIB PHM implementation

NASA Battery Dataset

B0005

1x1 struct 1 필드 포함

필드	값
cycle	*1x616 struct*

B0005 | B0005.cycle

B0005.cycle

필드	type	ambient_temperature	time	data
1	'charge'	24	[2008,4,2,13,8,17.9210]	*1x1 struct*
2	'discharge'	24	[2008,4,2,15,25,41.5930]	*1x1 struct*
3	'charge'	24	[2008,4,2,16,37,51.9840]	*1x1 struct*
4	'discharge'	24	[2008,4,2,19,43,48.4060]	*1x1 struct*

B0005 | B0005.cycle | B0005.cycle(1).data

B0005.cycle(1).data

필드	값
Voltage_measured	*1x789 double*
Current_measured	*1x789 double*
Temperature_measured	*1x789 double*
Current_charge	*1x789 double*
Voltage_charge	*1x789 double*
Time	*1x789 double*

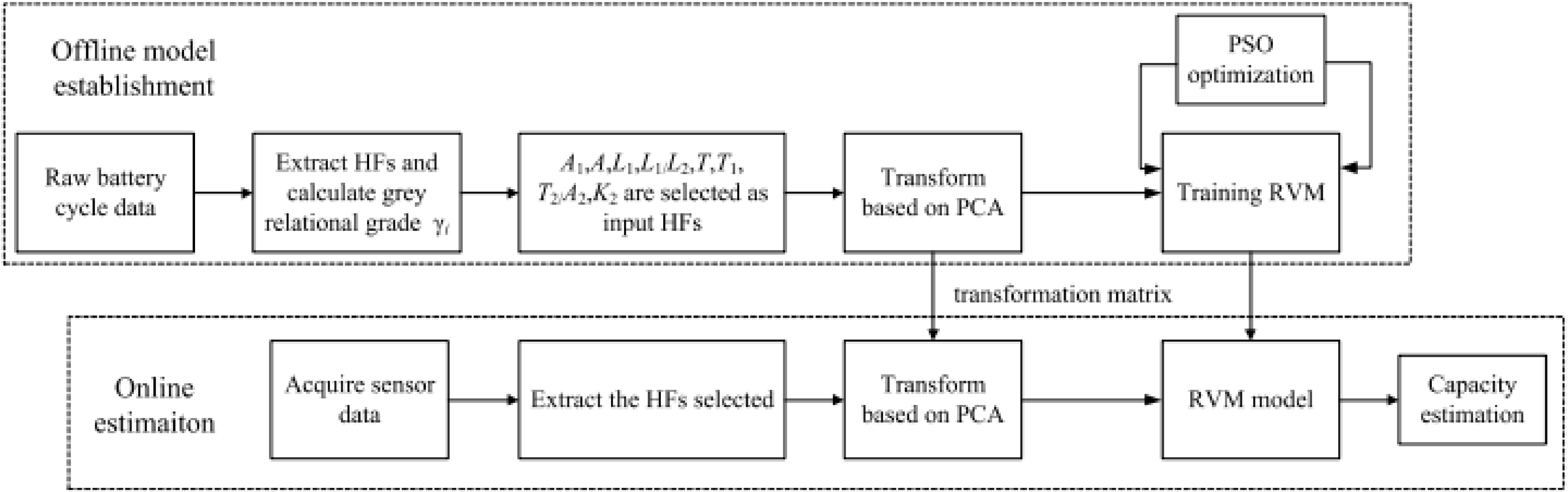
B0005 | B0005.cycle | B0005.cycle(2).data

B0005.cycle(2).data

필드	값
Voltage_measured	*1x197 double*
Current_measured	*1x197 double*
Temperature_measured	*1x197 double*
Current_load	*1x197 double*
Voltage_load	*1x197 double*
Time	*1x197 double*
Capacity	1.8565

LIB PHM implementation

Flow Chart



Algorithm

MODEL	ADVANTAGE	DISADVANTAGE
Neural – Network	Strong nonlinear approximation ability	Need a large amount of data for training and suffer from optimality
SVM (Support Vector Machine)	Handle nonlinear systems particularly with small set of training sample and find one global solution	Strong against binary classification but weak against multiple classification Hyperparameter tuning also needs to be validated
RVM (Relevance Vector Machine)	It has better performance than SVM. Hyperparameters are automatically tuned.	Longer training than SVM

LIB PHM implementation

Grey Relational Analysis

BLACK	GRAY	WHITE
no information	incomplete information	perfect information

Grey Relational Analysis

The grey relation coefficient for the i_{th} factor

$$\xi_i(k) = \frac{\min_{\forall i}\min_{\forall k} |x_0(k) - x_i(k)| + \rho \max_{\forall i}\max_{\forall k} |x_0(k) - x_i(k)|}{|x_0(k) - x_i(k)| + \rho \max_{\forall i}\max_{\forall k} |x_0(k) - x_i(k)|}$$

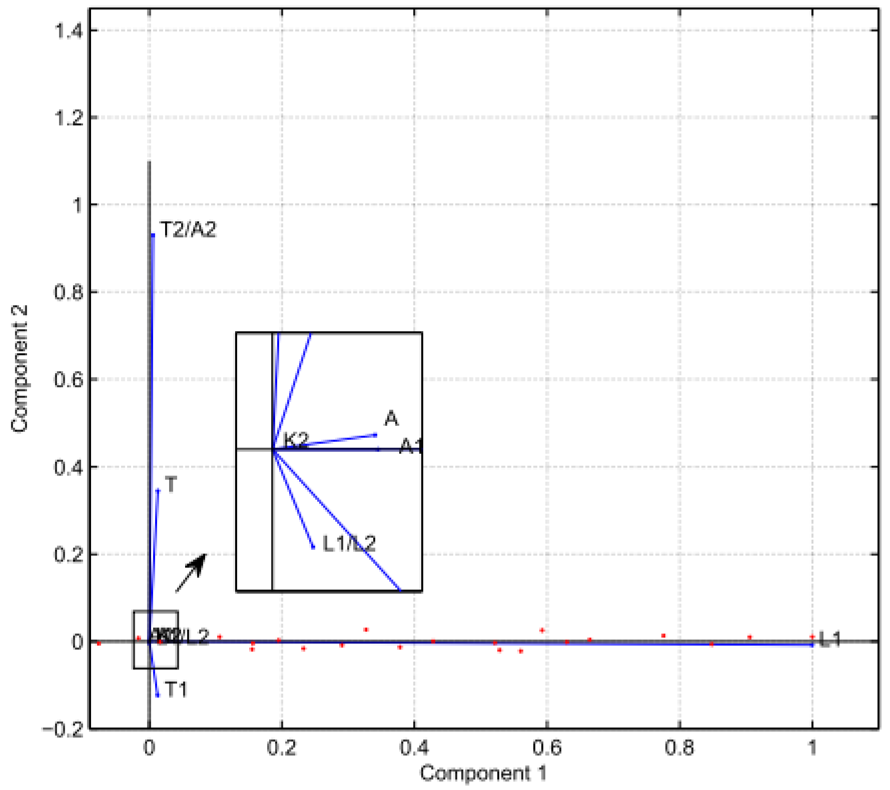
The grey relational grade

$$\gamma_i = \frac{1}{n}\sum_{k=1}^{n} \xi_i(k)$$

Determine the optimal condition of various input parameters to obtain the best quality characteristics.

LIB PHM implementation

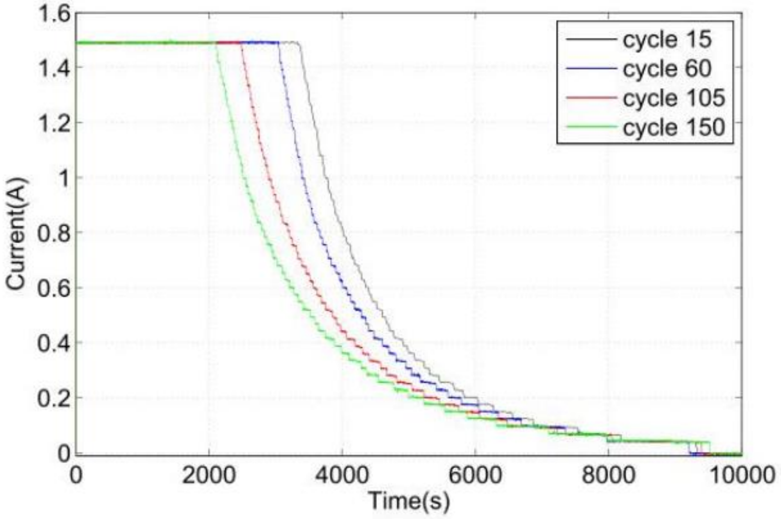
PCA (Principal Component Analysis)

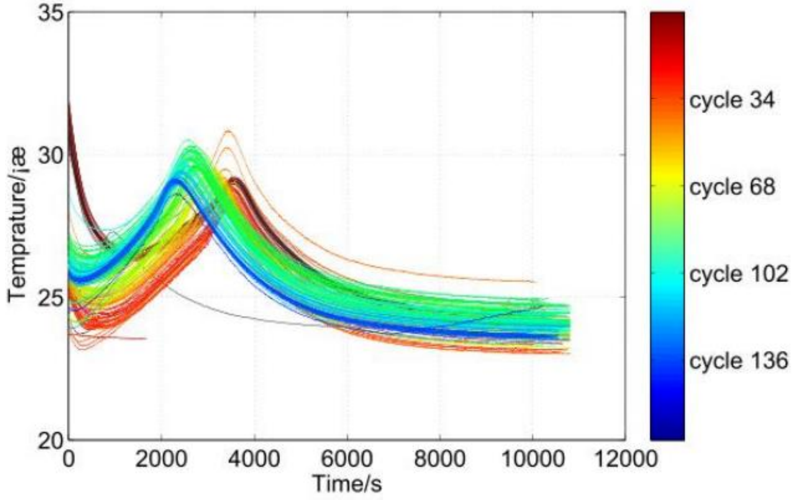


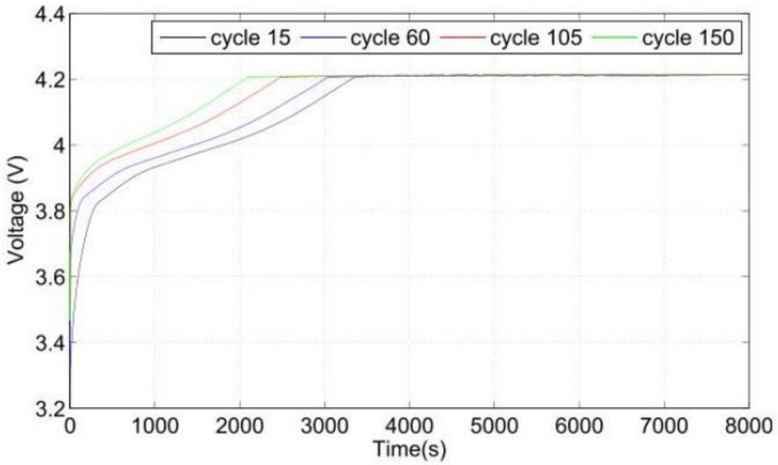
- **Information deduplication between HFs (= noise cancellation process).**

- **Data Dimension Reduction**

Feature Extraction

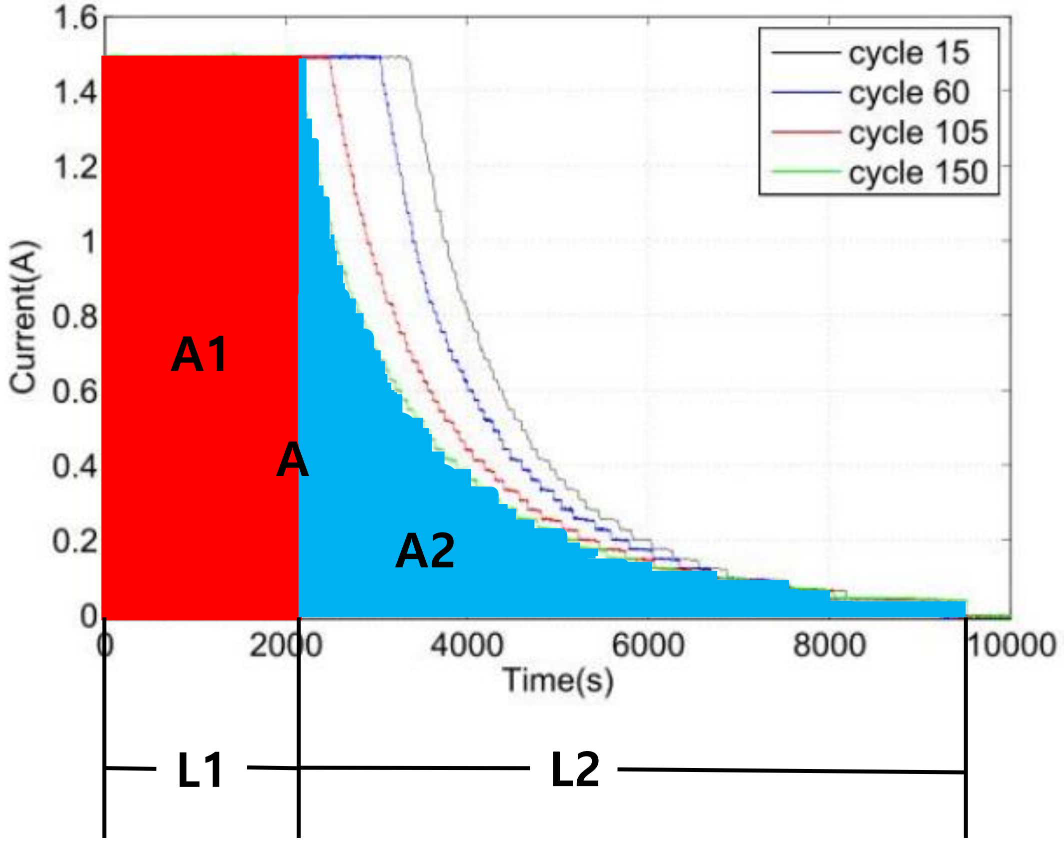


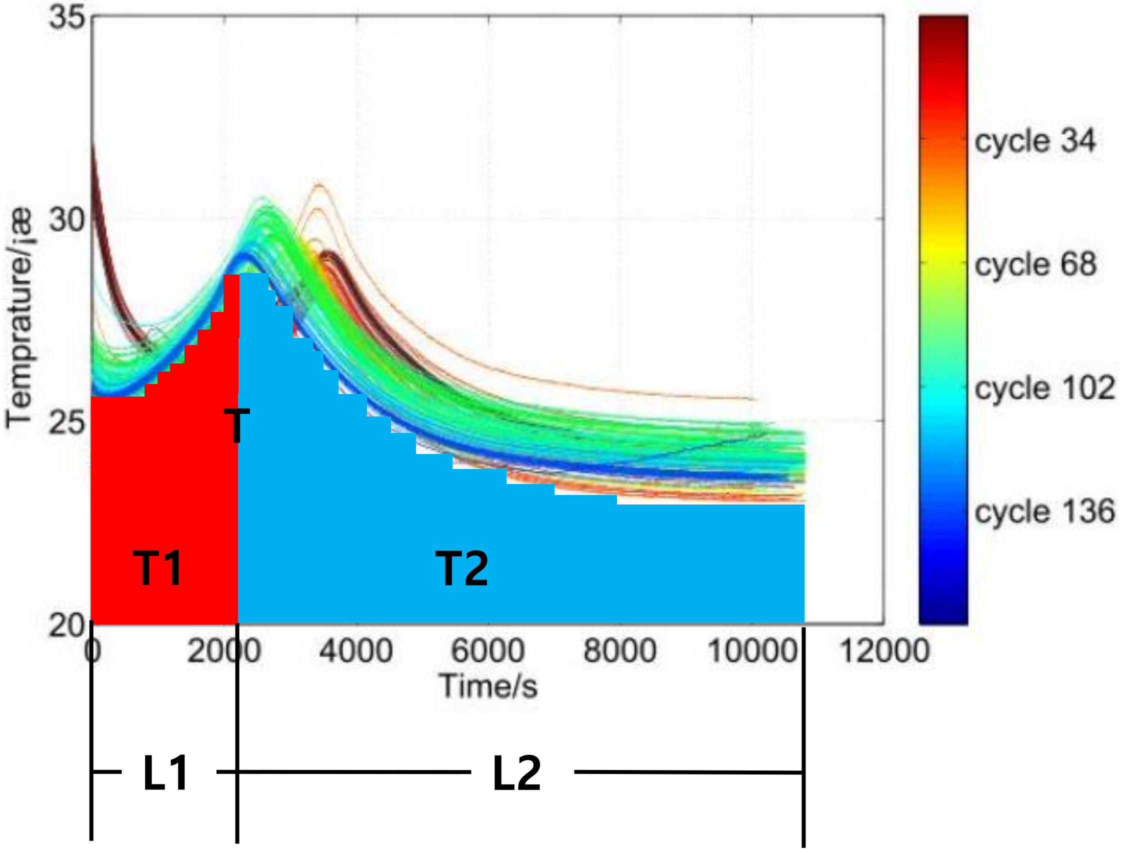




HFs are extracted from charging profiles(current, voltage and temperature)

LIB PHM implementation





Area	Time	Temperature	Slope
$A_1,\ A_2,\ A$	$L_1,\ L_2,\ L_1/L_2$	$T_1,\ T_2,\ T,\ T_1/A_1,$ $T_2/A_2,\ T/A$	$K_1,\ K_2$

Feature Selection: Grey Relational Analysis

Grey relational grades of HFs.

	K_1	K_2	L_1	L_2	L_1/L_2	T	T_1	T_2	T/A	T_1/A_1	T_2/A_2	A	A_1	A_2
B0005	0.5454	0.5930	0.7705	0.5120	0.8698	0.5987	0.7742	0.5025	0.5494	0.5729	0.6740	0.7130	0.7698	0.5112
B0006	0.5277	0.6626	0.9403	0.5163	0.8632	0.5931	0.9434	0.5061	0.5362	0.6386	0.7386	0.7783	0.9395	0.4982
B0025	0.5864	0.6214	0.7056	0.5809	0.7402	0.6001	0.7159	0.5795	0.6060	0.5671	0.5751	0.6353	0.7051	0.5805
B0026	0.4452	0.5261	0.4995	0.6173	0.5285	0.6190	0.5000	0.5988	0.5200	0.5807	0.5834	0.5671	0.4985	0.5452
B0029	0.6797	0.8073	0.9269	0.6915	0.7698	0.7858	0.7989	0.6938	0.5776	0.5803	0.7095	0.9251	0.9270	0.6076
B0030	0.7156	0.8217	0.9132	0.7158	0.7605	0.8323	0.7798	0.7177	0.5797	0.5652	0.6399	0.9329	0.9137	0.7519
B0045	0.5917	0.5838	0.8654	0.5478	0.8675	0.7368	0.8197	0.7294	0.6702	0.7475	0.6712	0.7101	0.8651	0.6938
B0046	0.7038	0.7203	0.8193	0.5366	0.8206	0.6174	0.7248	0.5626	0.6640	0.5638	0.6616	0.5898	0.8192	0.5784
B0053	0.7006	0.7483	0.6253	0.6975	0.6291	0.5429	0.5466	0.6734	0.5689	0.5429	0.5812	0.7977	0.6252	0.7038
B0054	0.7200	0.8058	0.6238	0.4764	0.7463	0.5256	0.6253	0.4829	0.5731	0.6011	0.5575	0.7297	0.6237	0.7670

Relevance degree sort result of HFs.

	Relevance:		High											Low
B0005	5	7	3	13	12	11	6	2	10	9	1	4	14	8
B0006	7	3	13	5	12	11	2	10	16	9	1	4	18	14
B0025	5	7	3	13	12	2	9	6	1	4	14	8	11	10
B0026	6	4	8	11	10	12	14	5	2	9	7	3	13	1
B0029	13	3	12	2	7	6	5	11	8	4	1	14	10	9
B0030	12	13	3	6	2	7	4	14	8	4	1	11	9	10
B0045	5	3	13	7	10	6	8	12	14	1	9	1	2	4
B0046	5	3	13	7	2	1	9	11	6	12	13	1	8	4
B0053	12	2	14	1	4	8	5	3	13	1	9	7	10	6
B0054	2	14	5	12	1	7	3	13	10	9	11	6	8	4

Number of Feature : 14 -> 8

LIB PHM implementation

Result of RVM Model
(Train/Test ratio = 7:3)

Estimation errors of single battery experiments.

	B0005	B0025	B0029	B0045	B0053
R_2	0.999958	0.999916	0.999966	0.999317	0.998693
ξ_{RMSE}	0.010222	0.016749	0.010046	0.018376	0.038024

Estimation errors of multiple battery experiments.

	B0018	B0028	B0032	B0048	B0056	B0033
R_2	0.999320	0.999710	0.999939	0.998944	0.999112	0.999376
ξ_{RMSE}	0.041133	0.030476	0.013651	0.041288	0.035195	0.042244

LIB PHM implementation

Designing Data-Driven Battery Prognostic Approaches for Variable Loading Profiles: Some Lessons Learned

Abhinav Saxena[1], José R. Celaya[2], Indranil Roychoudhury[3], Sankalita Saha[4], Bhaskar Saha[5], and Kai Goebel[6]

[1, 2, 3] *Stinger Ghaffarian Technologies Inc., NASA Ames Research Center, CA, 94035, USA*

abhinav.saxena@nasa.gov
jose.r.celaya@nasa.gov
indranil.roychoudhury@nasa.gov

[4,5] *Mission Critical Technologies Inc., NASA Ames Research Center, CA, 94035, USA*

sankalita.saha@gmail.com
bhaskar.saha@parc.com

[6] *NASA Ames Research Center, CA, 94035, USA*

kai.goebel@nasa.gov

Abstract

Among various approaches for implementing prognostic algorithms data-driven algorithms are popular in the industry due to their intuitive nature and relatively fast developmental cycle. However, no matter how easy it may seem, there are several pitfalls that one must watch out for while developing a data-driven prognostic algorithm. One such pitfall is the uncertainty inherent in the system. At each processing step uncertainties get compounded and can grow beyond control in predictions if not carefully managed during the various steps of the algorithms. This paper presents analysis from our preliminary development of data-driven algorithm for predicting end of discharge of Li-ion batteries using constant load experiment data and challenges faced when applying these algorithms to randomized variable loading profile as is the case in realistic applications. Lessons learned during the development phase are presented.

1. Introduction

The field of prognostics is steadily maturing as an important field under health management as newer algorithms are constantly being developed. Among the two main categories are data-driven and model-based algorithms with competing advantages and limitations (Schwabacher, 2005). This paper summarizes our experience from implementing a data-driven approach for a variable load discharge scenario for Lithium-ion (Li-ion) batteries using experimental data collected in controlled lab environment. An intuitive observation-based approach was initially implemented, which required considerable improvements as we learned about various shortcomings during the development process. In this paper we present our lessons learned from the exercise, as well as an analysis of various pitfalls that may be encountered in developing data-driven methods that may seem intuitive and relatively straightforward in the beginning but may not match up on expectations when actually implemented. The paper also presents a detailed description of our data-driven algorithm. Corresponding results are also compared with a model based algorithm using an empirical degradation model.

1.1. Motivation

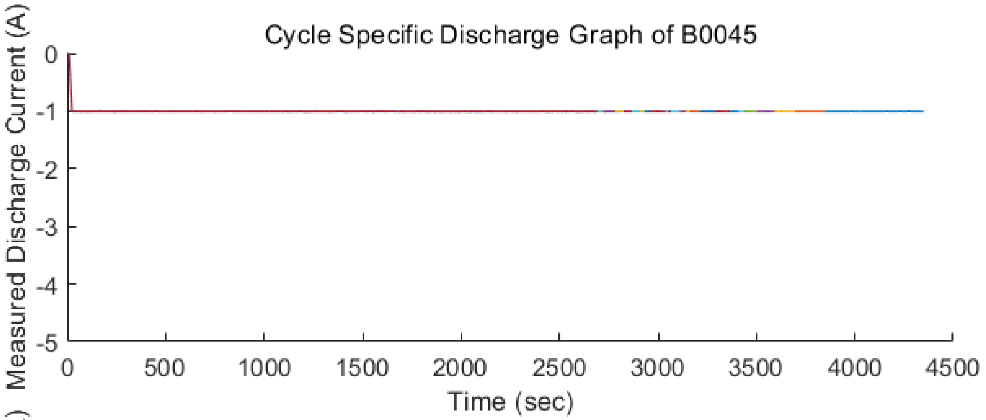
The motivation for this works stems primarily from two sources. First, it is of growing interest to develop prognostic health management solutions for Li-ion batteries as the use of power storage technologies is gaining momentum in energy intensive industries. While several efforts have focused on relevant topics, an accurate way of estimating battery capacity during realistic load profiles with variable and/or random operational loading still deserves attention. This paper describes the results of our efforts towards developing a generic data-driven approach for developing prognostic algorithms for randomized variable loading scenarios. It is generally assumed that data-driven methods typically require large amounts of training data in the initial development phase, but wherever possible, allow a much rapid, easy to implement, and computationally inexpensive developments compared to model-based approaches. This however, comes at a cost of a significant data processing effort upfront and still does not guarantee a successful implementation. More often than not it calls for re-evaluation of the initial hypothesis and may require significant changes adding to complexity as problems become more realistic. In this effort we exemplify a process

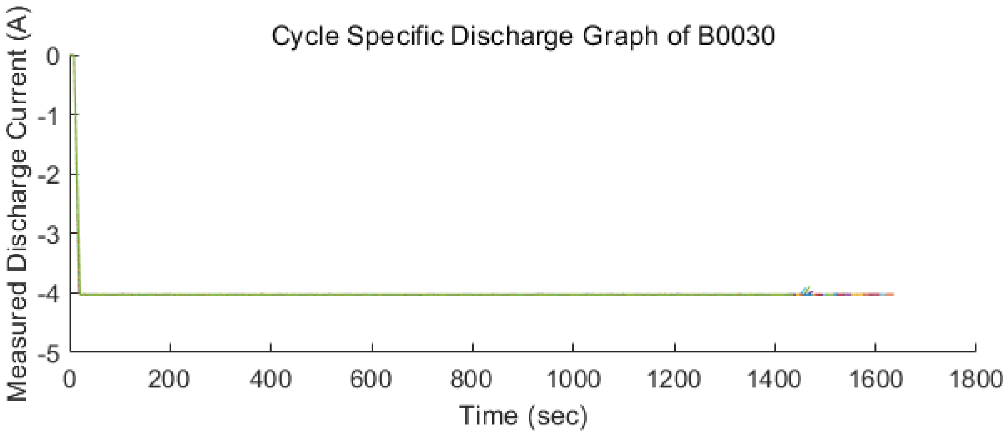
Predict the end of discharge (EoD) time of battery

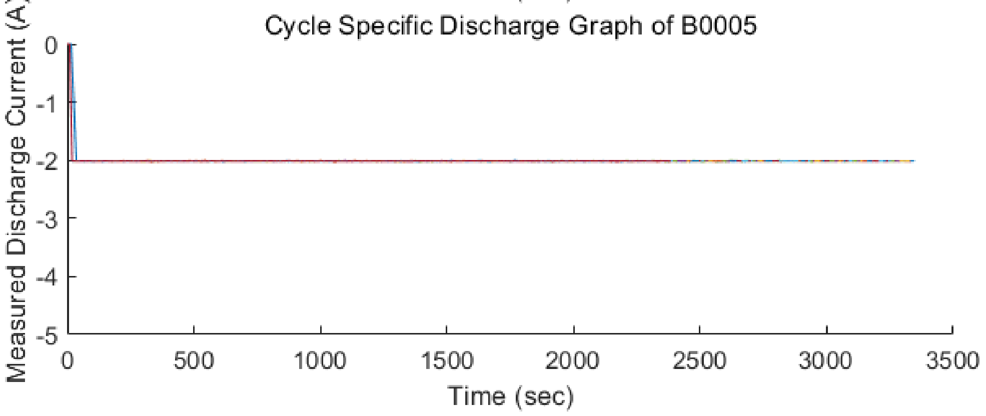
We only dealt with Constant Load Discharge Scenario

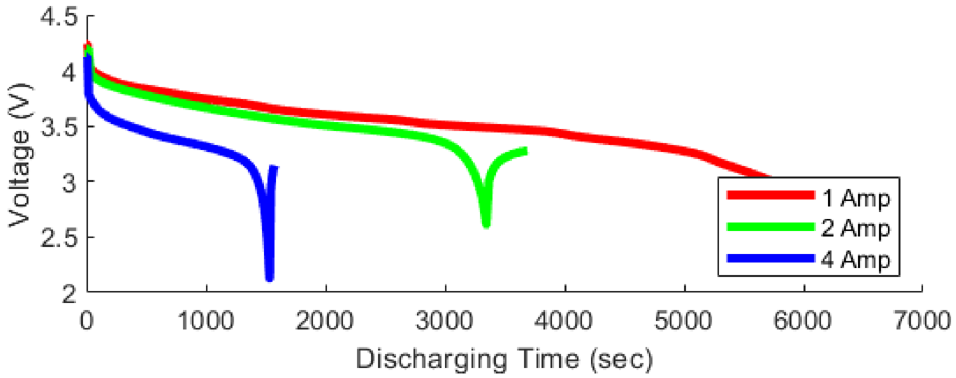
LIB PHM implementation

Data Analysis – Discharge Current & Voltage









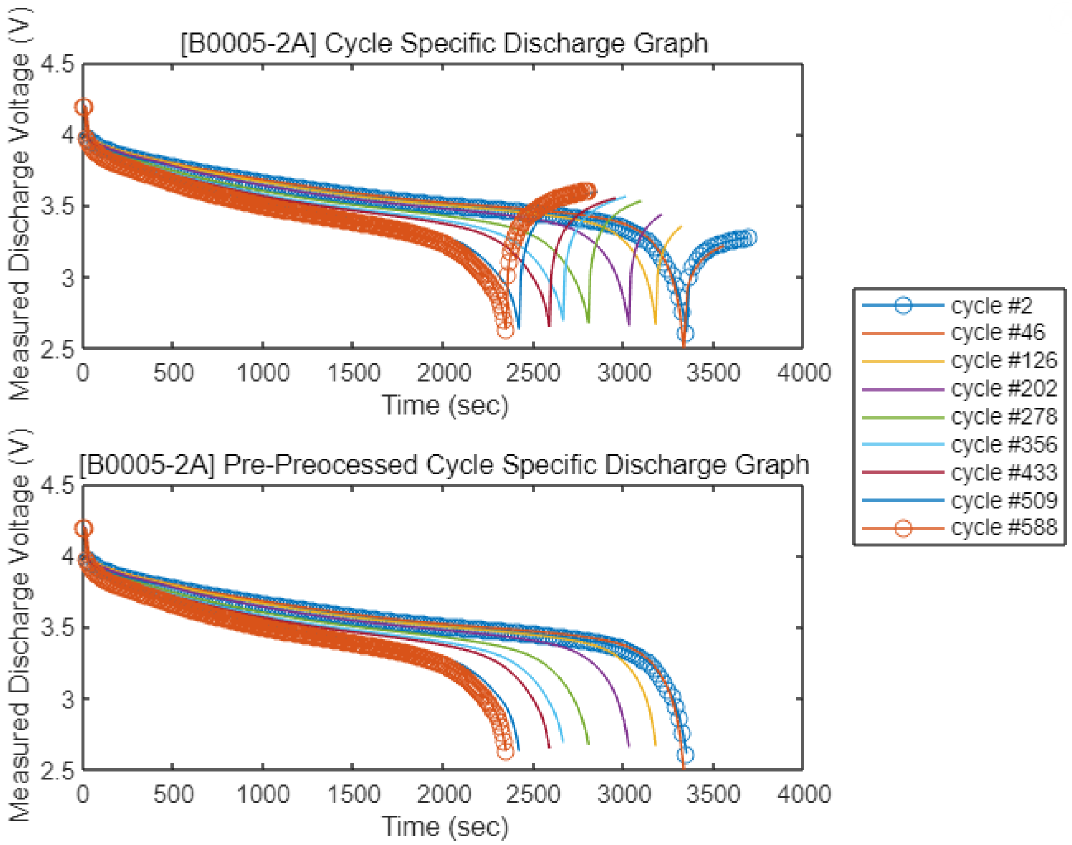
LIB PHM implementation

Flow Chart

Data Pre-Processing → Feature Extraction & Construct feature Table → Normalize Using Standard Scaler → Feature Selection & PCA → RUL

LIB PHM implementation

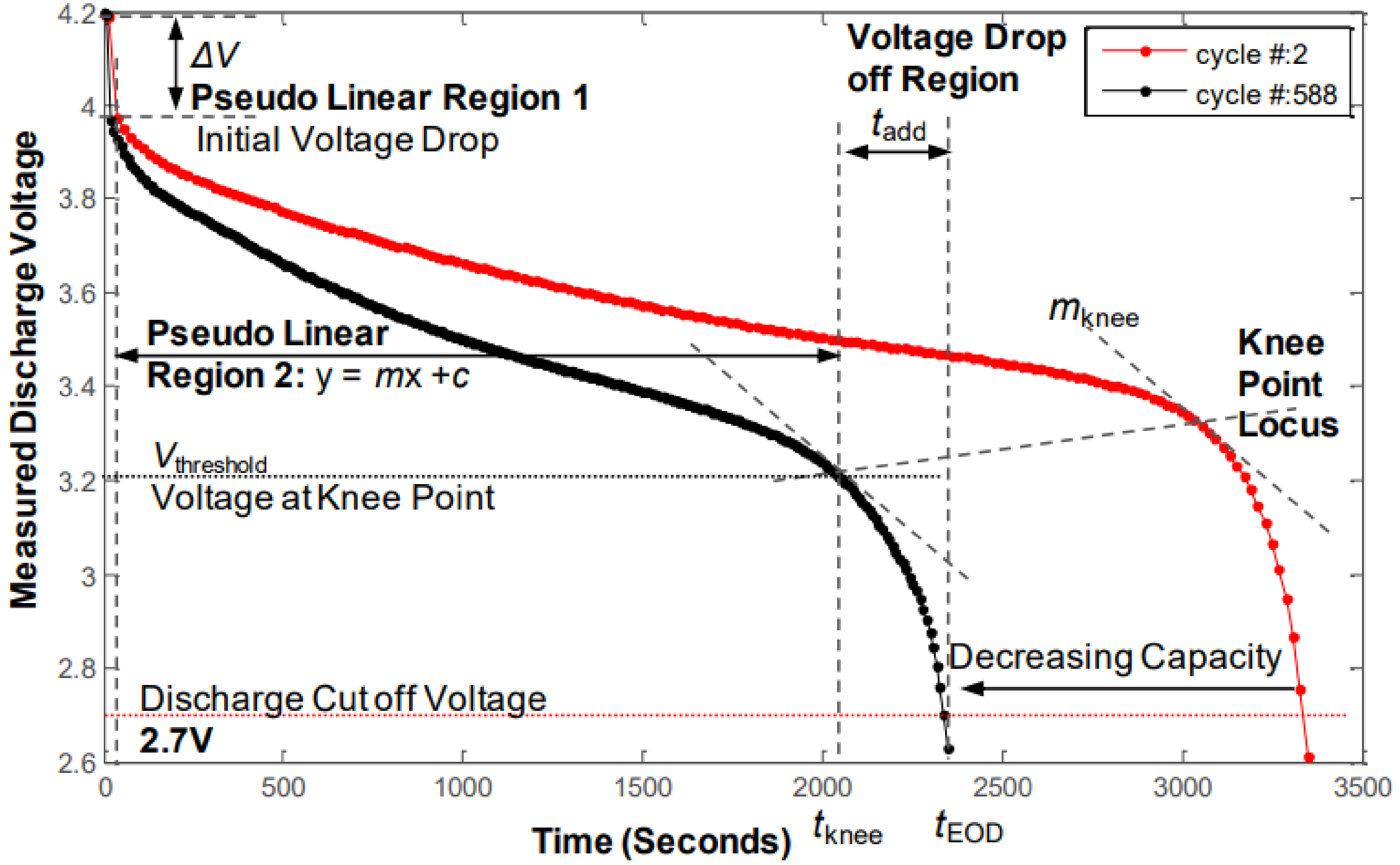
Data Analysis & Data Preprocessing



$\boldsymbol{t_{EoD}}$ **decreases with number of cycle**

Only use the data when $\mathbf{t} \leq \boldsymbol{t_{EoD}}$

LIB PHM implementation



Features

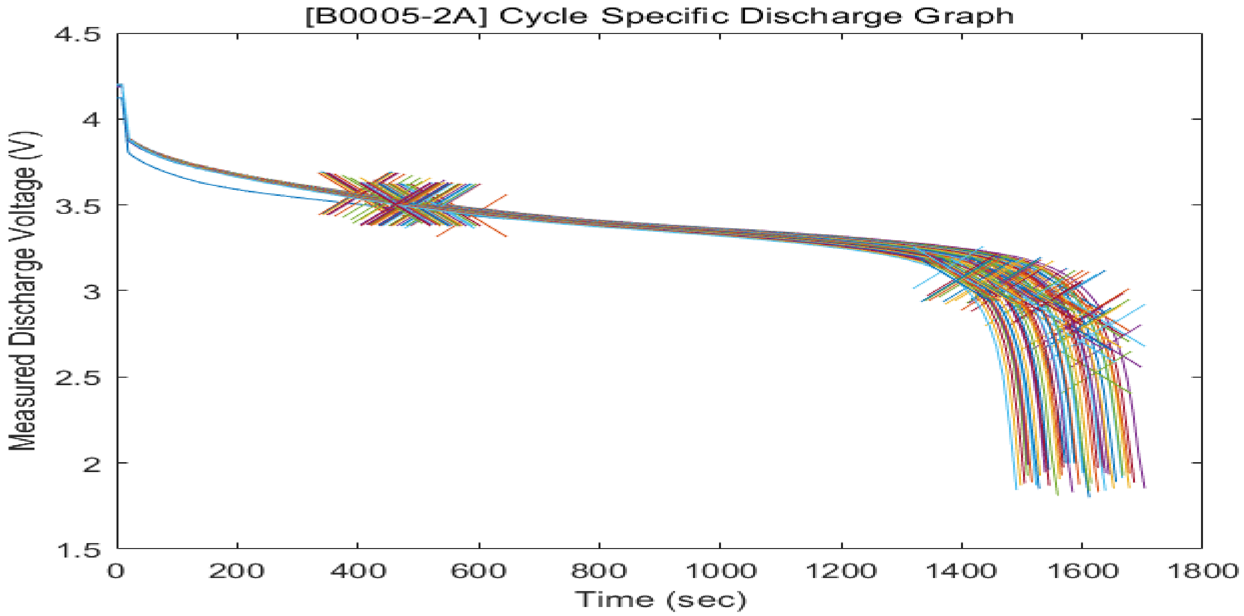
1. The battery SOH -> R_{meas}
2. The slope, m
3. The knee point time, t_{knee}
4. The remaining time, t_{add}
   $(t_{EoD} = t_{knee} + t_{add})$
5. The voltage discharging time

Feature Extraction

1. The battery SOH -> R_{meas}

- In each cycle, extract when data values change rapidly near the start point
- Since V = IR -> R = V/I, let's find R_meas = dV/dI.

2~4. The slope, m & The knee point time, t_{knee} & The remaining time, t_{add}



Identify the section of the pseudolinear region 2 by extracting the point at which the slope changes rapidly

Construct Feature Table

```
[thresh_cycle] = getThreshCycle(B0005)
```

thresh_cycle = 147

```
featureTable = featureTable(1 : thresh_cycle, :)
```

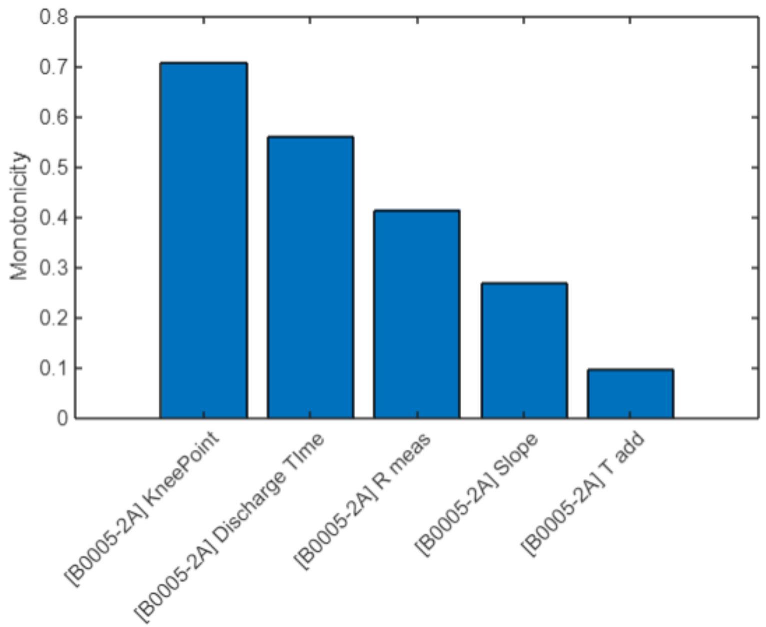
featureTable = 147x5 timetable

	Time	[B0005-2A] R meas	[B0005-2A] Slope	[B0005-2A] KneePoint	[B0005-2A] T add	[B0005-2A] Discharge Time
1	2008-04-02 15:25:41	0.1073	-1.6654e-04	3.0534e+03	293.5620	3.6902e+(
2	2008-04-02 19:43:48	0.1041	-1.6263e-04	3.0351e+03	293.7340	3.6723e+(
3	2008-04-03 00:01:06	0.1029	-1.5734e-04	2.9963e+03	313.1250	3.6516e+(
4	2008-04-03 04:16:37	0.1025	-1.6236e-04	3.0162e+03	293.5310	3.6316e+(
5	2008-04-03 08:33:25	0.1018	-1.6743e-04	3.0339e+03	273.8130	3.6292e+(
6	2008-04-03 12:55:10	0.1016	-1.6746e-04	3.0349e+03	274.3120	3.6523e+(
7	2008-04-03 17:17:16	0.1008	-1.7318e-04	3.0537e+03	254.7350	3.6508e+(
8	2008-04-03 21:28:14	0.1004	-1.6423e-04	3.0176e+03	273.8900	3.5725e+(
9	2008-04-04 01:38:15	0.0998	-1.7042e-04	3.0355e+03	254.4380	3.5506e+(

Construct Feature Table

To determine monotonicity,
data just before the time of failure were used.

```
% Since moving window smoothing is already done, set 'WindowSize' to 0 to
% turn off the smoothing within the function
featureImportance = monotonicity(featureTable( floor(breakpoint*0.6) : breakpoint, : ), 'WindowSize', 0);
helperSortedBarPlot(featureImportance, 'Monotonicity');
```



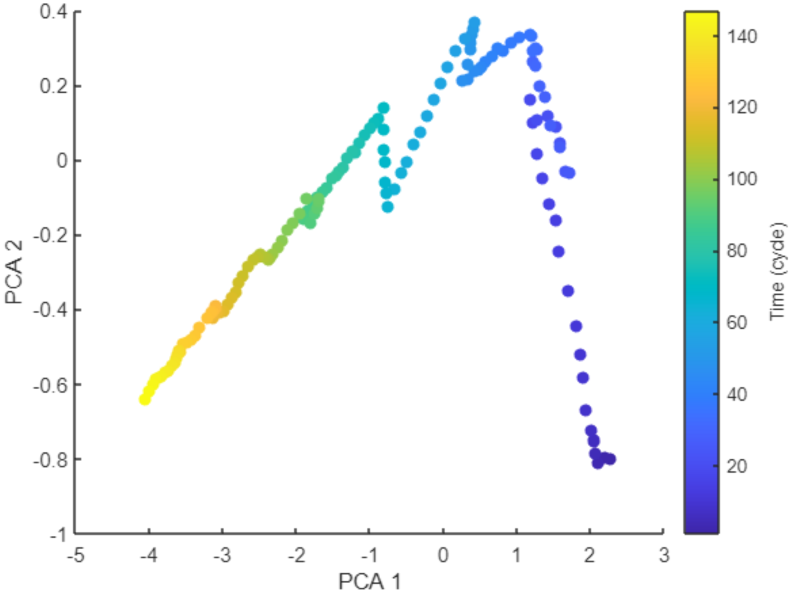

```
trainDataSelected  = trainData(:, featureImportance{:,:} > 0.5);
featureSelected    = featureTableSmooth(:, featureImportance{:,:} > 0.5)
```

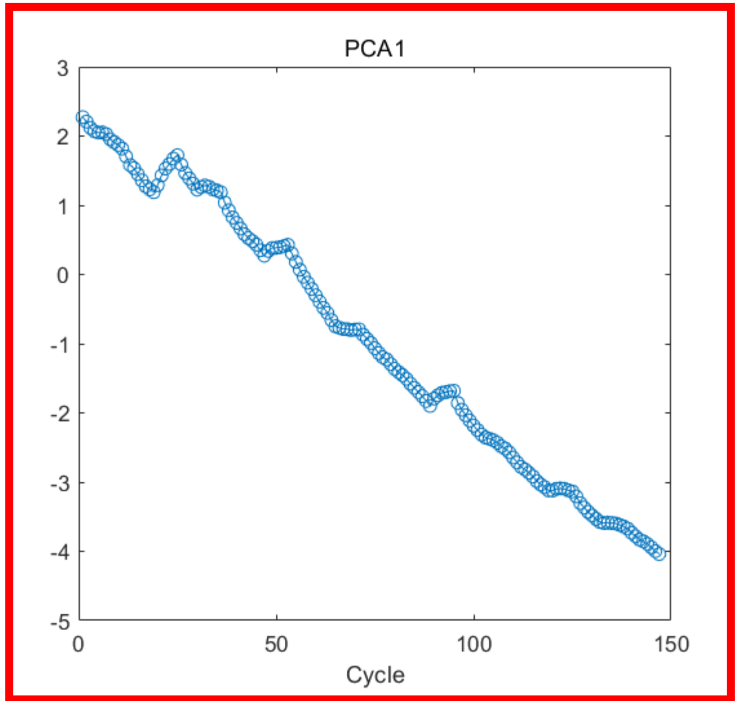
featureSelected = 147×2 timetable

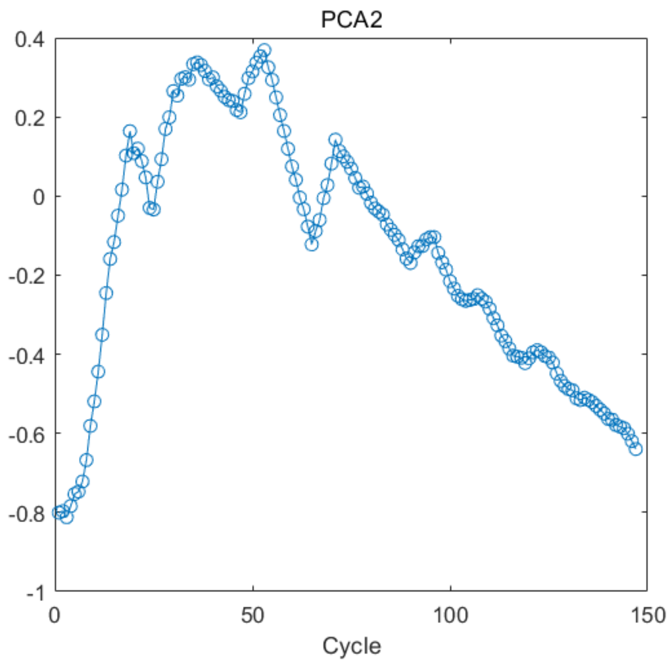
	Time	[B0005-2A] KneePoint	[B0005-2A] Discharge Time
1	2008-04-02 15:25:41	3.0534e+03	3.6902e+03
2	2008-04-02 19:43:48	3.0442e+03	3.6813e+03
3	2008-04-03 00:01:06	3.0283e+03	3.6714e+03
4	2008-04-03 04:16:37	3.0252e+03	3.6614e+03
5	2008-04-03 08:33:25	3.0270e+03	3.6550e+03
6	2008-04-03 12:55:10	3.0283e+03	3.6545e+03
7	2008-04-03 17:17:16	3.0283e+03	3.6480e+03
8	2008-04-03 21:28:14	3.0254e+03	3.6313e+03
9	2008-04-04 01:38:15	3.0320e+03	3.6145e+03

LIB PHM implementation

Normalize & Apply PCA







Health Indicator

RUL

Linear Degradation Model

The linearDegradationModel object implements the following continuous-time linear degradation model [1]:

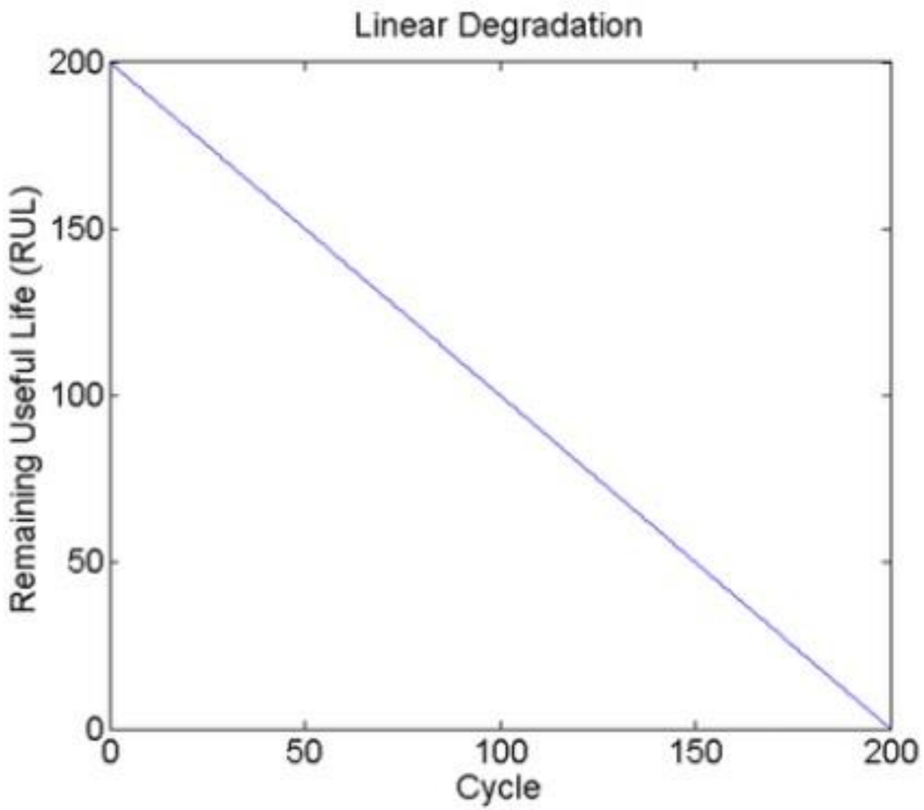
$$S(t) = \phi + \theta(t)t + \varepsilon(t)$$

where:

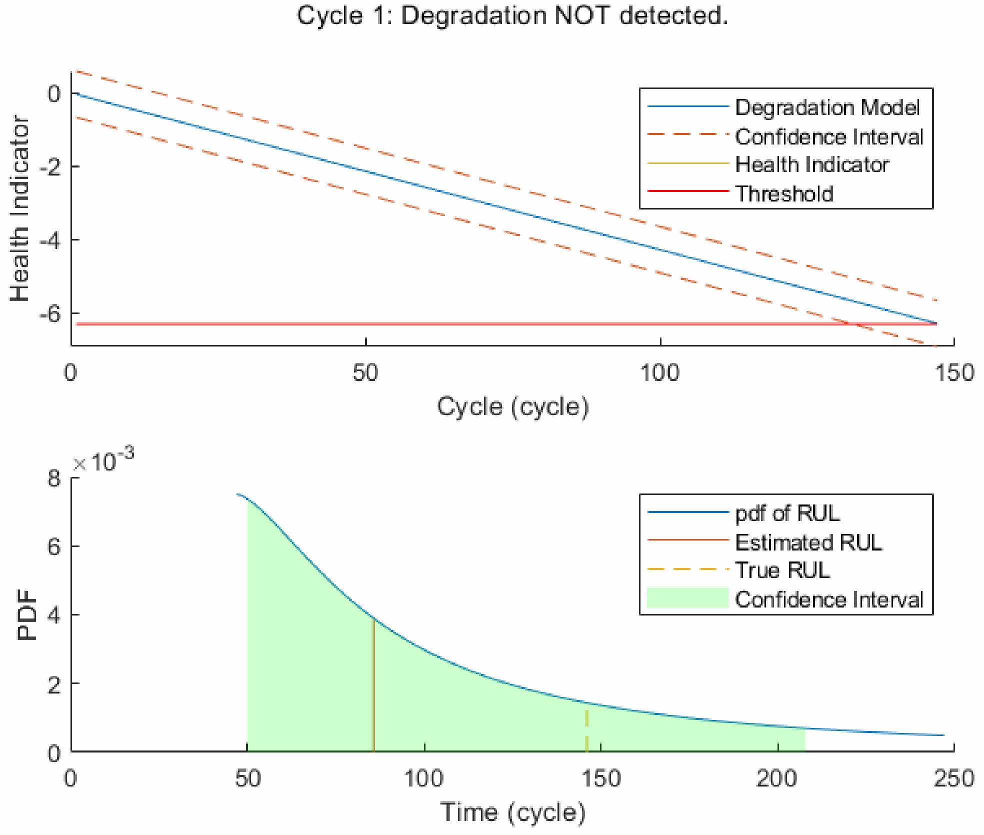
- ϕ is the model intercept, which is constant. You can initialize ϕ as the nominal value of the degradation variable using Phi.
- $\theta(t)$ is the model slope and is modeled as a random variable with a normal distribution with mean Theta and variance ThetaVariance.
- $\varepsilon(t)$ is the model additive noise and is modeled as a normal distribution with zero mean and variance NoiseVariance.

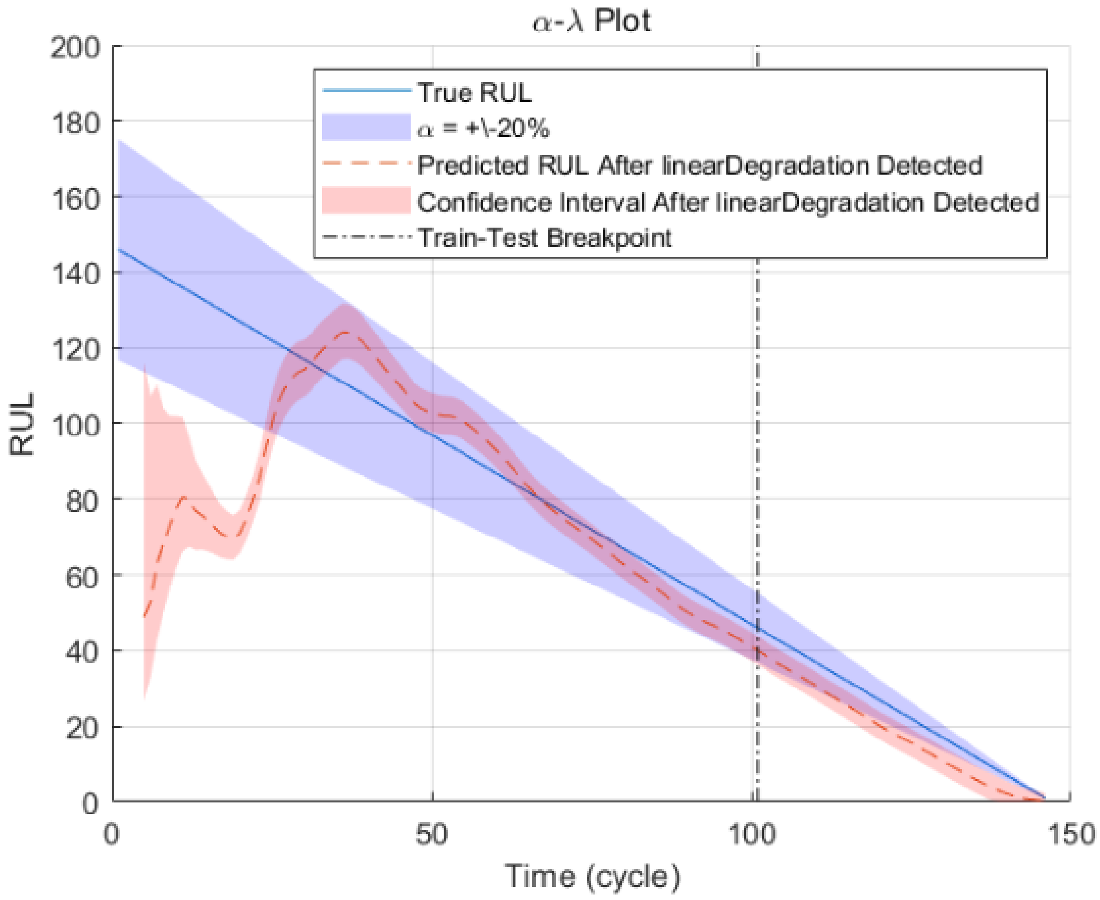
Therefore, Now create an linear degradation model.

```
mdl = linearDegradationModel(...
    'Theta', threshold/thresh_cycle, ...
    'ThetaVariance', 1/thresh_cycle, ...
    'Phi', healthIndicator(1), ...
    'NoiseVariance', 0.01);
```



RUL


Cycle 1: Degradation NOT detected.
Health Indicator
Cycle (cycle)
Degradation Model
Confidence Interval
Health Indicator
Threshold
0
-2
-4
-6
0
50
100
150
×10^-3
PDF
Time (cycle)
pdf of RUL
Estimated RUL
True RUL
Confidence Interval
8
6
4
2
0
0
50
100
150
200
250


α-λ Plot
RUL
Time (cycle)
True RUL
α = +\-20%
Predicted RUL After linearDegradation Detected
Confidence Interval After linearDegradation Detected
Train-Test Breakpoint
200
180
160
140
120
100
80
60
40
20
0
0
50
100
150

Thank you ☺

Q & A

Reference

[1] Peiyao Guo, Ze Cheng, Lei Yang. (2019). A data-driven remaining capacity estimation approach for lithium-ion batteries based on charging health feature extraction

[2] A Saxena, JR Celaya, I Roychoudhury, S Saha, B Saha, K Goebel (2012).
Designing Data-Driven Battery Prognostic Approaches for Variable Loading Profiles: Some Lessons Learned