CERC@MARC

Canada Excellence Research Chair
Laureate Program

Tesla 2170 Cell Data and SOC Estimation Blind Modeling Tool – Users Guide

Phillip Kollmeyer, *Ph.D., Member IEEE*
Senior Principal Research Engineer
McMaster University
kollmeyp@mcmaster.ca
electrification.mcmaster.ca

McMaster Automotive Resource Centre (MARC)

Motivation

- Hundreds of SOC estimation methods proposed in literature
- All tested in **<u>different</u>** ways though! (different data, testing methods, etc)

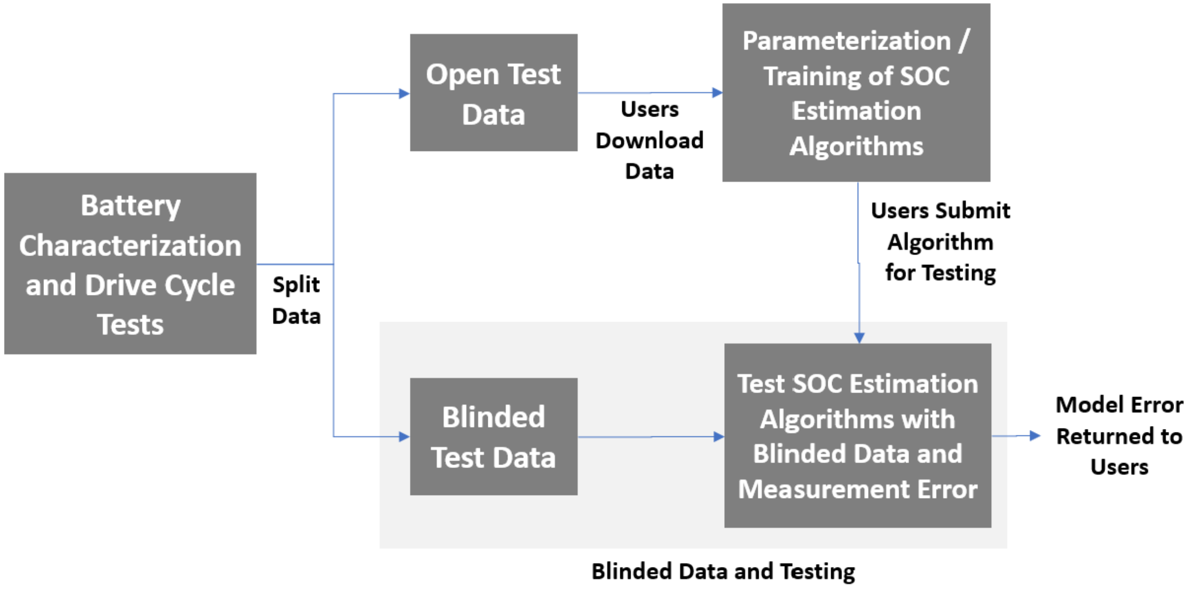
Standardized comparison method is needed

Example comparison of machine learning SOC estimators

ML Method	Lowest Error (only at 25⁰C)	Data Profiles	Battery	Multi-Temperature
FNN[11]	0.33%(RMSE)@WLTC 0.27%(MAE) @WLTC	FTP75, NEDC, US06, GUDC, Highway, WLTC	LFP (12V, 8Ah)	No
FNN[13]	0.84% $^{+++}$(MAE)@ US06 0.61% (MAE)@ HWFET	HWFET and US06	[40]Li-ion Panasonic NCR18650PF	-20℃, -10℃, 0℃, 10℃ , and 25℃
FNN w/ UKF[15]	1.4%(RMSE)@FUDS 2.5%(RMSE)@US06	FUDS, US06, DST	LFP (2.3Ah max)	0℃, 10℃, 20℃, 25℃, 40℃, and 50℃
FNN w/ECM[14]	0.33% (MAE)@FUDS	FUDS, DST	LFP (24V, 20Ah)	No
FNN w/ BSA[21]	0.81%(RMSE)@DST 0.91%(RMSE)@FUDS	FUDS, DST	NMC (3.6V, 2Ah)	0℃, 25℃, and 45℃
RBF w/ EKF[23]	~3% (RMSE)@CCC $^{+}$	Constant current charging (CCC)	Li-ion (1.2Ah)	No
RBF w/ H∞[25]	0.7%(RMSE) @CCC $^{+}$	Constant current charging (CCC)	Li-ion (1.2Ah)	No
RBF w/ UKF[26]	~3% (RMSE)@CCC $^{+}$	Constant current charging (CCC)	Li-ion (1.2Ah)	No
RBF w/ RSMO [27]	2.32%(RMSE)@UDDS $^{+}$ 2.33%(RMSE)@HWY $^{+}$	UDDS, Highway	Lithium-polymer, Turnigy (3.7V, 0.5Ah)	No
RBF w/ AEKF [29]	<2%(MAE)@DST $^{+}$	DST, HPPC	Lithium-polymer (3.7V, 32Ah)	No
RNN [38]	0.57%(MAE)@ $^{++++}$	HWFET, UDDS, LA92, and US06	[40] Li-ion Panasonic NCR18650PF (3.6V, 2.9Ah)	0℃, 10℃, and 25℃
RNN [42]	0.32%(MAE)@LA92 0.86%(MAE)@ BJDST	Panasonic: HWFET, UDDS, LA92, and US06 Samsung: FUDS, US06, BJDST	[40] Li-ion Panasonic NCR18650PF Li-ion Samsung 18650-20R	0℃, 10℃, 25℃, and 40℃ (Panasonic) 0℃, 25℃, and 45℃ (Samsung)
RNN [46]	NMC: 0.77%(MAE)@ $^{+}$ LFP: 1.72%(MAE)@ $^{+}$	DST	BAK B18650CD A123 18650 (LFP)	0℃, 25℃, 20℃, 30℃, 40℃, 50℃
RNN[50]	LFP :0.53%(RMSE) @ $^{++}$ LTO :0.70%(RMSE) @ $^{++}$	Dynamic char./discharge profile $^{++}$	LFP (3.6V) LTO(2.6V)	0℃, 10℃, 25℃, and 40℃
ELM w/ AUKF[31]	0.4%(RMSE)@ $^{++}$	Constant pulse discharging current $^{++}$	Samsung 2.6Ah	No
SVM[33]	0.4%(RMSE)@DST	DST	LFP (3.6V, 60Ah)	No

References listed in: C. Vidal, P. Malysz, P. Kollmeyer and A. Emadi, "Machine Learning Applied to Electrified Vehicle Battery State of Charge and State of Health Estimation: State-of-the-Art," in *IEEE Access*, vol. 8, pp. 52796-52814, 2020, doi: 10.1109/ACCESS.2020.2980961.

Blind Modeling Tool Concept



- Standardized dataset for SOC estimator parametrization / training
- Standardized method for testing SOC estimators with **blinded** data

Publishing Results from Tool

- Please **use** shared data and blind modeling tool **results** in your research
- **More** published studies using tool = **More** opportunities for comparison

Cite this paper when using data/tool:

P. J. Kollmeyer, M. Naguib, F. Khanum and A. Emadi, "A Blind Modeling Tool for Standardized Evaluation of Battery State of Charge Estimation Algorithms," *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2022, pp. 243-248, doi: https://doi.org/10.1109/ITEC53557.2022.9813996

A Blind Modeling Tool for Standardized Evaluation of Battery State of Charge Estimation Algorithms

Phillip J. Kollmeyer*, *Member, IEEE,* Mina Naguib, *Student Member, IEEE,* Fauzia Khanum, Ali Emadi, *Fellow, IEEE*
McMaster Automotive Resource Centre (MARC), McMaster University
*Email: kollmeyp@mcmaster

Abstract- **There are hundreds of approaches to estimating battery state of charge (SOC). It is difficult to compare results reported in different papers because each typically uses a different dataset. While some papers compare multiple SOC estimation algorithms, the author's bias, skill, or effort towards each algorithm may unintentionally skew the results. A standardized way to test and compare methodologies between authors is necessary to allow the best algorithms to stand out. An example in another application area is the National Institute of Standards (NIST) Face Recognition Vendor Test, which compares facial recognition software using a standardized dataset. A similar approach is proposed here for batteries, where data is provided for users to parameterize and train their algorithms. An online tool is provided to subject the algorithms to a wide range of blinded test cases. A high-quality dataset is prepared using battery cells from a prevalent electric vehicle. A total of sixty-four drive cycles are performed at each of six temperatures ranging from -20 °C to 40 °C. The blind modelling tool is demonstrated for one SOC estimation algorithm. It will be made available for researchers to benchmark and compare their algorithms.**

I. INTRODUCTION

Lithium-ion (Li-ion) batteries have been widely adopted for powering electric vehicles (EVs). Accurate battery state of

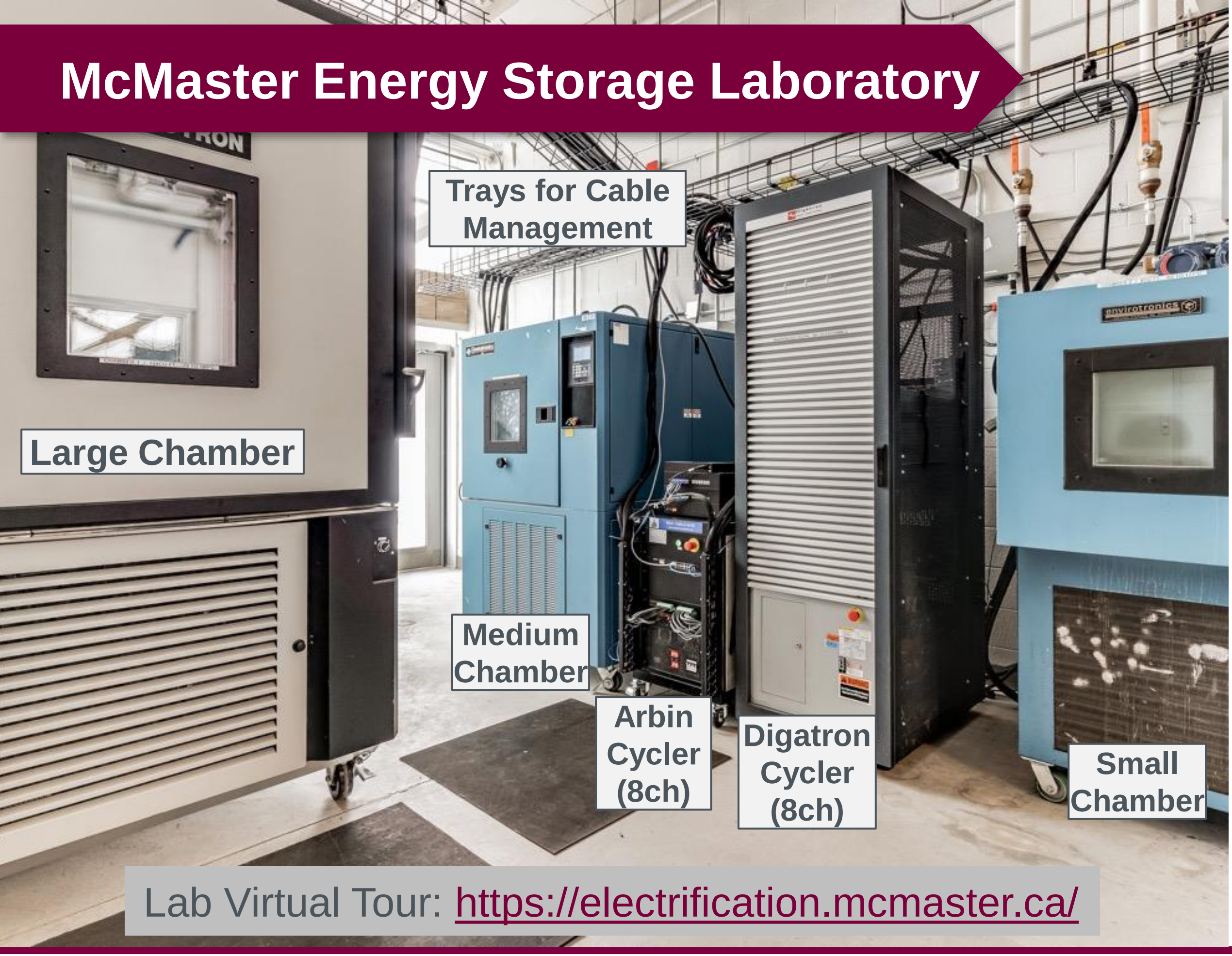
The experimental results were kept blind from the contestants, so their modelling techniques were not influenced by awareness of the correct solution. Another example is the National Institute of Standards (NIST) Face Recognition Vendor Test (FRVT) [6], which uses actual images taken at immigration checkpoints to benchmark the accuracy of various facial recognition algorithms. A ranking for the tested algorithms is published, and each algorithm is given a score based on the face recognition accuracy.

To address the challenge of fairly comparing battery SOC estimation techniques, a blind modelling tool is proposed in this work. Battery characterization and drive cycle data are provided to the users to develop and test their SOC estimation algorithm. The users then submit their algorithm via a web portal, and the algorithm is tested with a variety of drive cycles and test cases to which they are blind. The error for each test case is returned to the user, and these error values can be used as a standardized metric to compare algorithms proposed by different researchers. A specialized reference dataset is created for the modelling tool by testing four Tesla/Panasonic 2170, 4.5Ah cells, with drive cycles modelled for the Tesla Model 3 EV at a range of temperatures. A total of 384 unique drive

Presentation Summary

Table of Contents

McMaster Energy Storage Laboratory

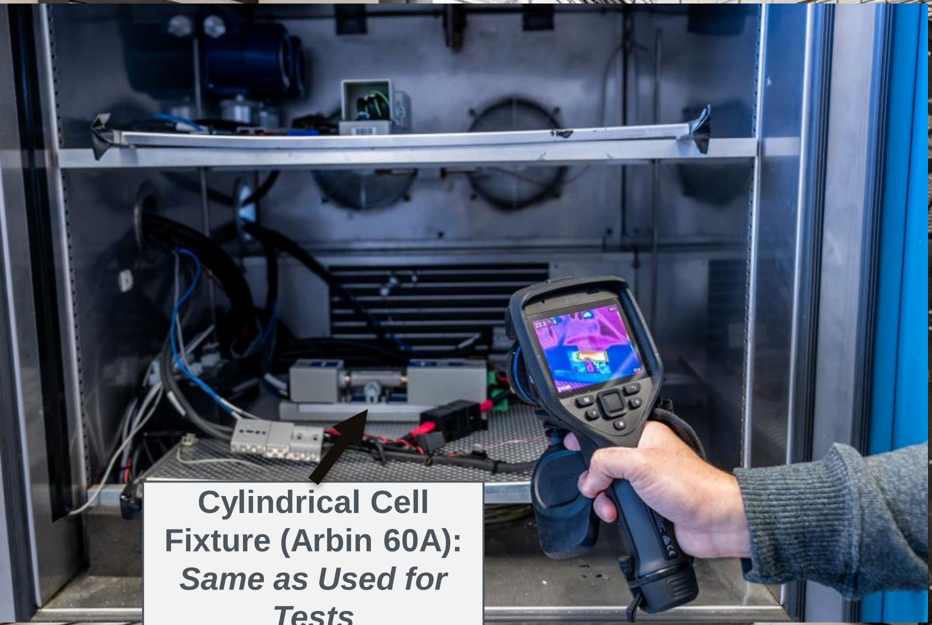


Lab Virtual Tour: https://electrification.mcmaster.ca/

- Cell cycling, 17 channels, up to 600A
- Pack cycling: up to 1,000V/400A or 500V/800A
- Three thermal chambers, one chiller
- Integrated explosive gas sensing and fire suppression

Tests Performed in McMaster Laboratory

Example Battery Test Setups



See *“3-McMaster Battery Lab Overview”* for more lab details

Battery Cycler and Thermal Chamber used for Tests

Envirotronics SH16 Thermal Chamber

- 16 ft^3, -30 to 177 ^oC
- +/- 0.3 ^oC accuracy
- 1.5kW cooling @ -30 ^oC
- Serial control interface

Arbin LBT Cell Tester

- 8 x 60A, 0-5V channels
- Parallel channels for 480A
- 0.04% of range accuracy, error low as +/-8µA

Arbin Cell Tester Specifications

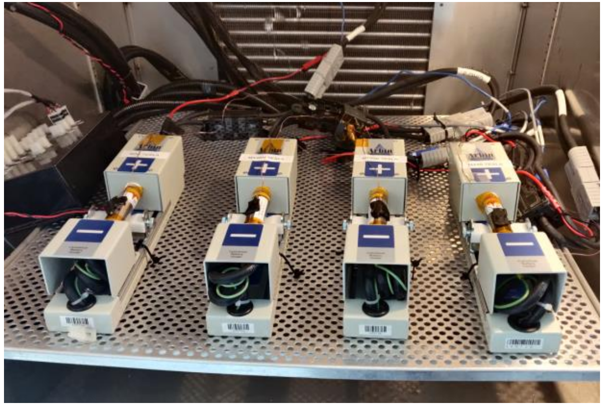
Voltage	0V to 5V
Current	60A per channel
# of channels	8
Parallel operation	2 to 8 sequential channels can be operated in parallel
Series operation	Series connections cannot be made
Input impedance	50 MΩ
Current Range	+/- 60A, 5A, 500mA, 20mA
Control Accuracy	+/- 24mA, 2mA, 200µA, 8µA (0.04%) +/- 2mV (0.04%)
Max command rate	5ms
Max system log rate	2000 samples per second
Temperature Sensing	16 channels, type T thermocouples
Control Software	Arbin MITS 8.0

Description of Cells Tested

- Source: Tesla Model 3 Battery Pack
- Form Factor: Cylindrical, 2170
- Capacity at Start of Tests: Around 4.7 Ah

Tesla Model 3

Tesla cells in chamber



Tests Performed on Cells

Four battery cells tested

- Drive cycle power profiles created using vehicle model of Tesla Model 3 EV standard range (50kWh pack)
- Each cell has different combination of payload mass and HVAC on/off condition

Cell	Payload Mass (kg)	HVAC Enabled	Referenced Name
1	80 kg	Yes	m80
2	448 kg	Yes	m448
3	448 kg	No	m448-N
4	1000 kg	Yes	m1000

Description of Battery Experiments

	Battery Cell Test Case			
Tests performed at -20, -10, 0, 10, 25, 40 °C	m80	m448	m448-N	m1000
C/20 discharge capacity (always at 40 °C)	Open	Blind	Open	Open
C/3, C/2, 1C discharge	Open	Blind	Open	Open
C/20 charge and discharge	Open	Blind	Open	Open
HPPC test with four pulse magnitudes	Open	Blind	Open	Open
UDDS, HWFET, LA92, US06 drive cycles	Blind	Blind	Blind	Blind
Eight reordered drive cycles	Open	Blind	Open	Open
Two HWCUST drive cycles	1 Blind, 1 Open	Blind	1 Blind, 1 Open	1 Blind, 1 Open
Two HWGRADE drive cycles	1 Blind, 1 Open	Blind	1 Blind, 1 Open	1 Blind, 1 Open

Open data shared

Blind data not shared, only used for testing

For complete details, see ITEC 2022 paper for description of test cases: P. J. Kollmeyer, M. Naguib, F. Khanum and A. Emadi, "A Blind Modeling Tool for Standardized Evaluation of Battery State of Charge Estimation Algorithms," *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2022, pp. 243-248, https://doi.org/10.1109/ITEC53557.2022.9813996

Tests on Blinded Data for Validating Model Performance

- Tool provided for submitting and testing models (see section 4 of presentations)
- Following tests performed on submitted models:

Test	Description	Data
1	All cells	All cells, all drive cycles
2	Blinded cell	Blinded m448 cell, all drive cycles
3	Non-Blinded cells	Non-blinded m80, m448-N, m1000 cells, all drive cycles
4	Charging	Charge data, m80 cell
5	Range of Vehicle Masses	Compare m80, m448, m1000 test cases, all drive cycles
6	HVAC On/Off	m448 cell (HVAC On) compared to m448-N cell (HVAC Off)
7	Standard Drive Cycles	UDDS, HWFET, LA92, US06 for m1000 cell

Test	Description	Data
8	Non-Standard Drive Cycles	HWCUST, HWGRADE for m1000 cell
9	Range of Temperatures	Compare -20, -10, 0, 25, 40 °C test cases, m80cell
10	Initial SOC Error	Emulate initial SOC error via test case starting at SOC other than 100%
11	Current Sensor Offset	Current sensor offset error added for multiple cases
12	Robustness	Gain, offset, noise error added to voltage, current, or temperature measurement for multiple cases

Data Structure

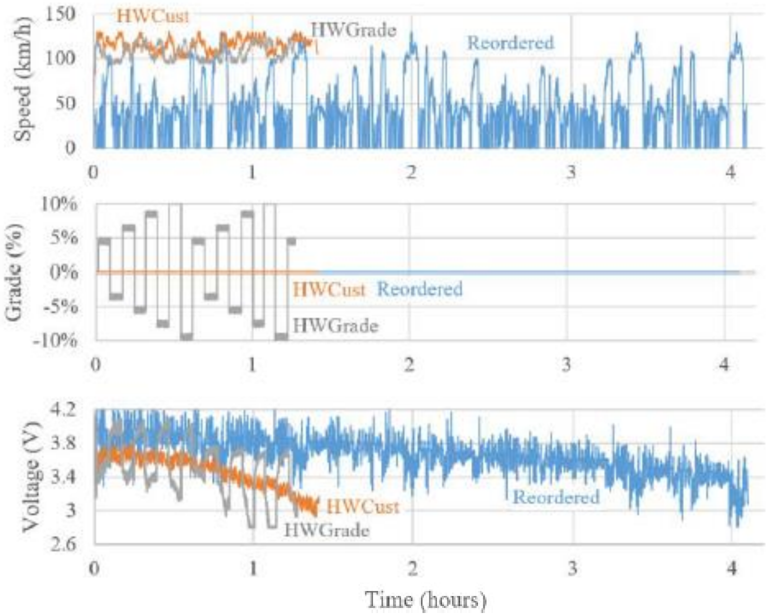
- Open data provided in “1-Open Data” folder
- Organized by cell (m80, m448n, m1000, see earlier slide for description)
- Separate folders for each temperature (40C, 25C, etc) and for characterization and drive cycle data

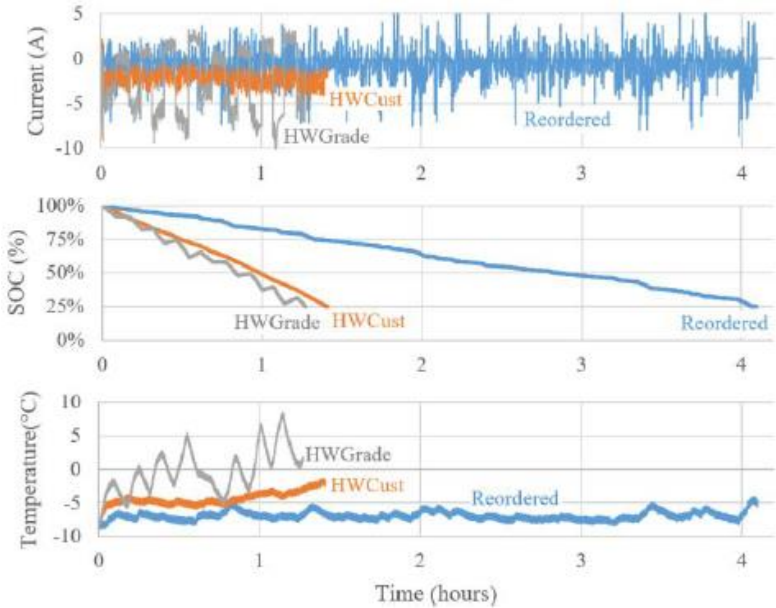
Example Data Files

Name	Status	Date modified	Type	Size
9_CC_CV_charge_10-11-21_01.30 Tesla_0C_DriveCycles_Ch4	✓	3/18/2022 8:56 PM	MATLAB Data	797 KB
10_REORDERED1_10-10-21_20.11 Tesla_0C_DriveCycles_Ch4	✓	3/18/2022 8:56 PM	MATLAB Data	986 KB
11_CC_CV_charge_10-11-21_01.30 Tesla_0C_DriveCycles_Ch4	✓	3/18/2022 8:56 PM	MATLAB Data	797 KB
12_REORDERED2_10-11-21_12.10 Tesla_0C_DriveCycles_Ch4	✓	3/18/2022 8:56 PM	MATLAB Data	947 KB
13_CC_CV_charge_10-11-21_17.17 Tesla_0C_DriveCycles_Ch4	✓	3/18/2022 8:56 PM	MATLAB Data	725 KB
14_REORDERED3_10-12-21_02.37 Tesla_0C_DriveCycles_Ch4	✓	3/18/2022 8:56 PM	MATLAB Data	945 KB

See “readme.txt” for full description of files and naming conventions

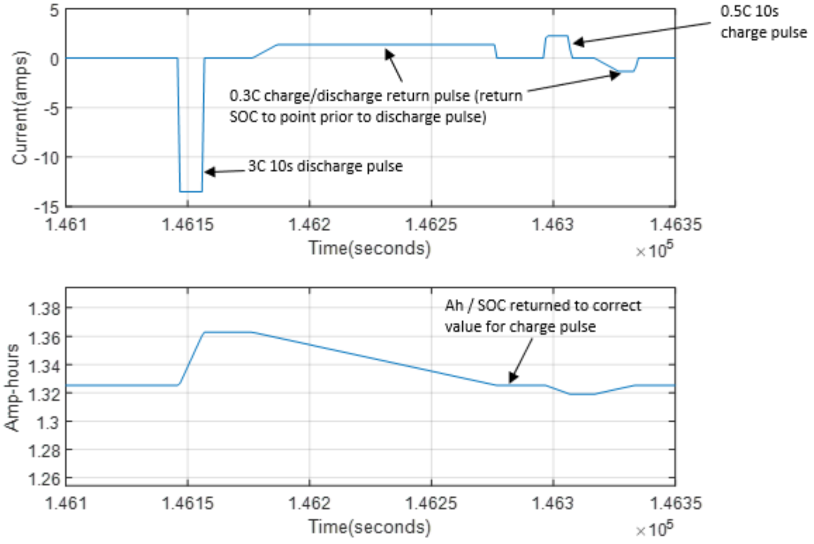
Example Data: Drive cycles for m80 cell at -10°C





HPPC Test Explanation

One HPPC charge and discharge pulse



Description of complete procedure

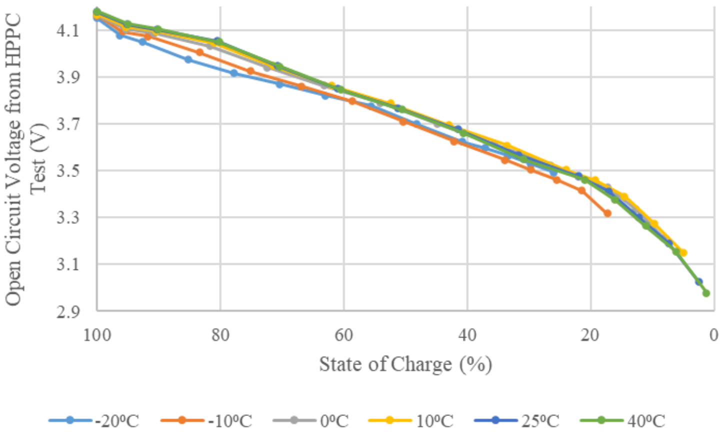
The HPPC tests consist of four ten second discharge and charge pulses in the following order, with 0.3C pulses in between to return SOC to prior value. The 0.3C return pulses are to ensure that each 10s charge and discharge pulse is performed at the exact same SOC:

1. Discharge cell at 0.3C until SOC reaches specified value (i.e. 95% SOC)
2. Pause one hour
3. C/2 10s discharge pulse
   - Pause 20s
   - 0.3C charge until SOC returned to value for this step (i.e .95% SOC)
   - Pause 20s
4. C/8 10s charge pulse
   - Pause 10s
   - 0.3C discharge until SOC returned to value for this step (i.e .95% SOC)
   - Pause 10 minutes
5. 1C 10s discharge pulse
   - Pause 20s
   - 0.3C charge until SOC returned to value for this step (i.e .95% SOC)
   - Pause 20s
6. C/4 10s charge pulse
   - Pause 10s
   - 0.3C discharge until SOC returned to value for this step (i.e .95% SOC)
   - Pause 10 minutes
7. 2C 10s discharge pulse
   - Pause 20s
   - 0.3C charge until SOC returned to value for this step (i.e .95% SOC)
   - Pause 20s
8. C/3 10s charge pulse
   - Pause 10s
   - 0.3C discharge until SOC returned to value for this step (i.e .95% SOC)
   - Pause 10 minutes
9. 3C 10s discharge pulse
   - Pause 20s
   - 0.3C charge until SOC returned to value for this step (i.e .95% SOC)
   - Pause 20s
10. C/2 10s charge pulse
   - Pause 10s
   - 0.3C discharge until SOC returned to value for this step (i.e .95% SOC)
   - Pause 10 minutes

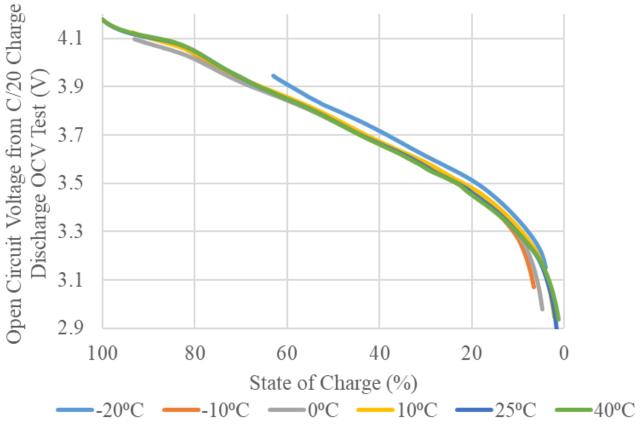
Open Circuit Voltage (OCV)

- OCV often needed for Kalman filter and other equivalent circuit based SOC estimation methods
- OCV is derived from two different tests and provided in “Characterization Test Plots_m80.xlsx”, as shown in the following plots:

OCV after 1 hour wait in HPPC test



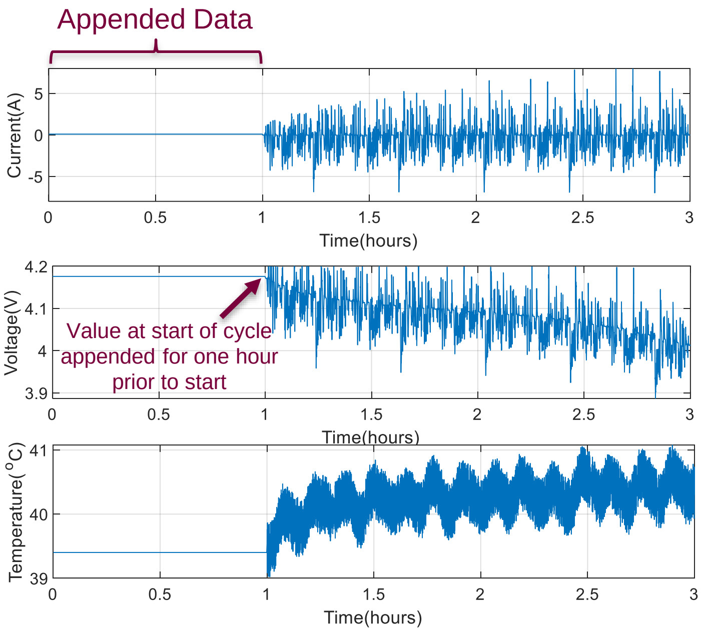
OCV from average of C/20 charge / discharge voltage



Users my find they are better off deriving OCV from the test data in other ways, this OCV data is just provided as an example

Padding of Test Data

- **Testing tool adds one hour of padded data prior to all test cycles applied to algorithms**
- **Ensures that algorithms are not impacted by lack of initial data**
  - For example, recurrent neural networks or algorithms which filter voltage need initial data

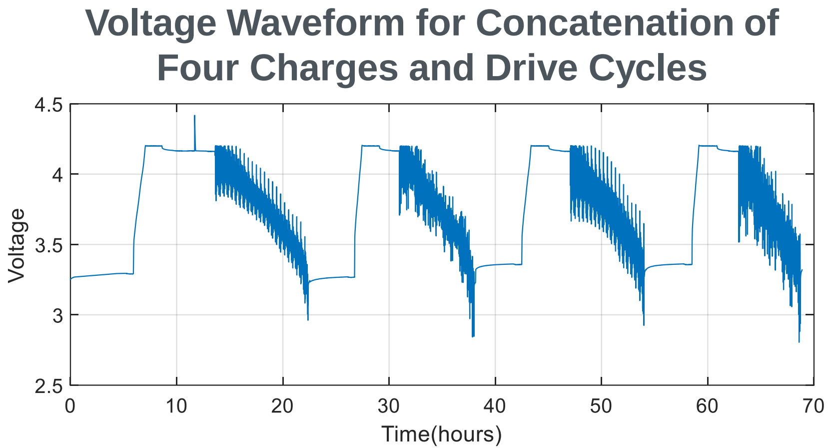


Presentation Summary

Table of Contents

File Concatenation Tool

- Concatenates multiple individual test files into a single file –
  - Tool can be helpful to create larger files of contiguous data for training and testing machine learning algorithms or for testing Kalman filter type algorithms
- Provided in “2-File Concatenation Tool” folder
- To use tool, open folder in Matlab, open and run “ConcatenateFilesByName.m”



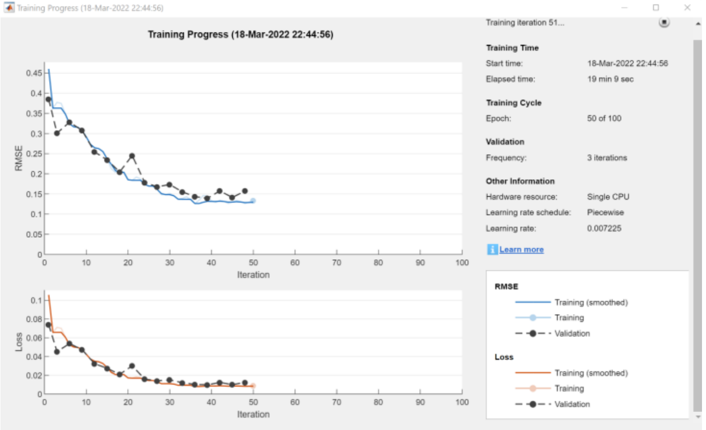
Presentation Summary

Table of Contents

LSTM SOC Estimator Example

- MATLAB script for training LSTM SOC estimator, in “3-Example SOC Estimation Training Algorithm” folder
- **To run example training:**
  1. Open folder in MATLAB
  2. Paste “LSTM_Training_Algorithm(10,5,0.01,20,0.85,1,1,1,1)” into command window and press enter

Window Which Appears When Running Training



10 epochs training should take less than 10 minutes

See “readme.txt” in “3-Example SOC Estimation Model” folder for full description of LSTM example

- **Note:** The example LSTM was not found to achieve very good accuracy, may not be a good option for estimating SOC of these battery cells

Presentation Summary

Table of Contents

Steps for Submitting Model for Blind Testing

- Step 1: Format SOC estimator as *Model.m* or *Model.p* Matlab function
- Step 2: Create *Settings.xlsx* file
- Step 3: Zip files into *Blind_Model.zip*
- Step 4: Test *Blind_Model.zip* using *SOC_Estimator_Test_Script.m*
- Step 5: After testing zip file with script from Step 4, upload zip file via online form
- Step 6: Email sent to author confirming submission
- Step 7: Email sent to author with results
- Step 8: View results sent in *Error Summary.zip*

Step 1: Format model as *Model.m* or *Model.p* Matlab function

- **Input Data (measured):** *X*
  - T x 3 array, where T is length of test data in seconds
  - Current, voltage, and temperature
  - 1 Hz sample frequency
  - Each row is 1 second

- **Output Data (estimated):** *Y*
  - T x 1 array, where T is length of test data in seconds
  - SOC, from 0 (0%) to 1 (100%)
  - 1 Hz sample frequency
  - Each row is 1 second

X

15503x3 double

	1	2	3
5580	0	4.1301	26.8951
5581	0	4.1299	25.7370
5582	2.2939	4.1686	25.7370
5583	4.4447	4.3633	25.7939

Current Column 1 | **Voltage** Column 2 | **Temperature** Column 3

→ *.m* or *.p* SOC estimator function →

Must be named Model.m or Model.p

Y

15503x1 double

	1	2
5580	0.9591	
5581	0.9591	
5582	0.9591	
5583	0.9594	

Estimated SOC Column 1

What is a p-code (*.p*) file?

- A p-code file is a ***content obscured executable file***: https://www.mathworks.com/help/matlab/ref/pcode.html
- Create ***.p*** version of a ***.m file*** by typing *pcode model.m* for example in the Command Window
- Use ***.p*** files if you want to obscure your source code
- All code submitted to the blind model tool will be kept confidential and will only be used to determine the results

Step 1: Example Coulomb Counting Function

```
%Coulomb Counting SOC Estimator - McMaster University 2022
function [Y_est] = Model(X)

%Input X: Measured current, voltage, and temperature values
%X: 3 columns, T rows, where T is length of input data in seconds
Current = X(:,1); %Amps, column 1
%Current: negative-discharging, positive-charging
Voltage = X(:,2); %Volts, column 2
Temperature = X(:,3); %degrees Celsius, column 3
Time = (0:1:(length(Current)-1))'; %seconds

%Coulomb Counting SOC Estimator
SOC_Init = 1; %Assume battery always starts fully charged
Capacity = 4.6; %Ah, Nominal capacity of new Tesla 21700 NMC/NCA cell

%Coulomb counting: SOC = integral of current
for i=1:length(Time)...

%Output Y: Estimated SOC
%Y: 1 columns, T rows, where T is length of input data in seconds
Y_est=SOC'; %Transpose SOC from columns to rows
end
```

See folder:

Example 1 - Coulomb Counter

Step 1: Example LSTM Machine Learning Function

```
%LSTM Neural Network SOC Estimator - McMaster University 2022
function [Y_est] = Model(X)

%Normalize input data X to normalization used for trained network
MAX =    [15,     4.5,     51 ];
MIN =    [-19,    2.5,     -27];
X(:,1) = ((X(:,1) - MIN(1))./(MAX(1)-MIN(1)));
X(:,2) = ((X(:,2) - MIN(2))./(MAX(2)-MIN(2)));
X(:,3) = ((X(:,3) - MIN(3))./(MAX(3)-MIN(3)));

%Reorder and transpose X data to match neural network format
X_reordered = [X(:,2), X(:,1), X(:,3)]';

%Load trained network parameters
load("Trained_LSTM_Network_Parameters.mat");

%Estimate SOC
[updatedNet,Pr] = predictAndUpdateState(NETS{1,1},X_reordered(:,1:100));
Y_est = predict(updatedNet, X_reordered(:,:))';
end
```

See folder:

Example 2 - Recurrent Neural Network

Step 2: Create *Settings.xlsx* file

- Save file as "Settings.xlsx"
- **Data in file includes**
  - Author Name: *Cell B1*
  - Author Affiliation: *Cell B2*
  - Author Email Address: *Cell B3*
  - Model Name: *Cell B4*

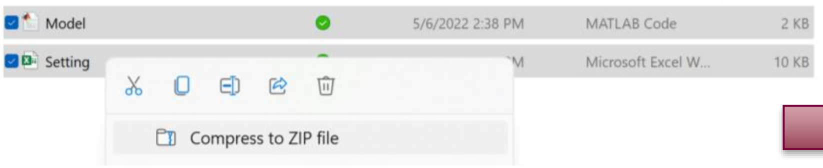
Example file – *cells must be the same, all fields must be occupied*

	A	B
1	Author Name	Phillip Kollmeyer
2	Author Affiliation	McMaster University
3	Author Email	kollmeyp@mcmaster.ca
4	Model Name	Coulomb Counter

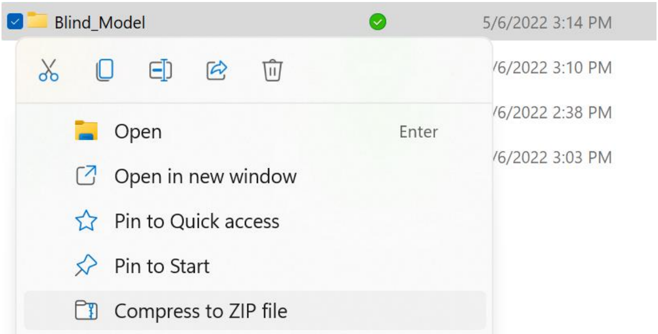
Step 3: Zip files into *Blind_Model.zip*

Zip files directly to *Blind_Model.zip*

Zip files directly



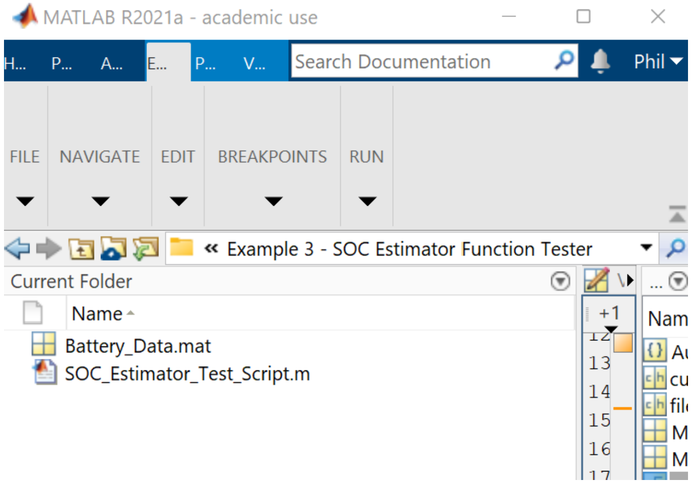
Do not place files in folder and zip folder



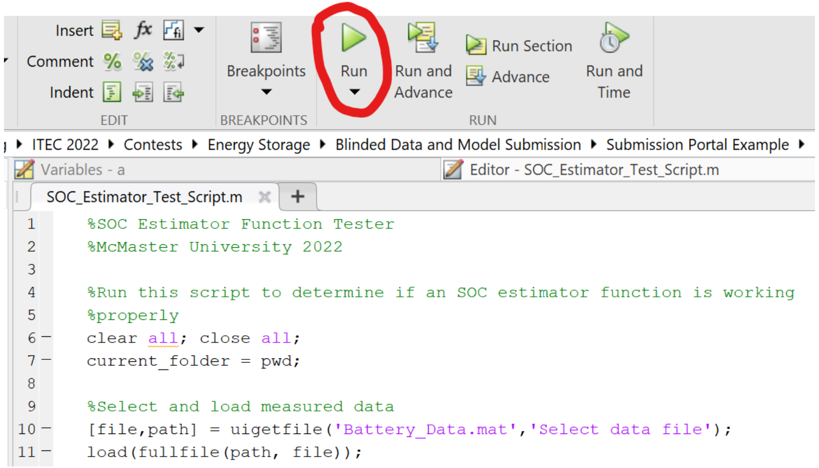
- Zipped folder **cannot** include any sub folders
- Must include files named *Model.m* and *Settings.xlsx*
- May include other files needed by *Model.m*
- .zip file name does not have to be "Blind_Model"

Step 4: Test *Blind_Model.zip* using *SOC_Estimator_Test_Script.m*

a. In Matlab open folder containing *SOC_Estimator_Test_Script.m*

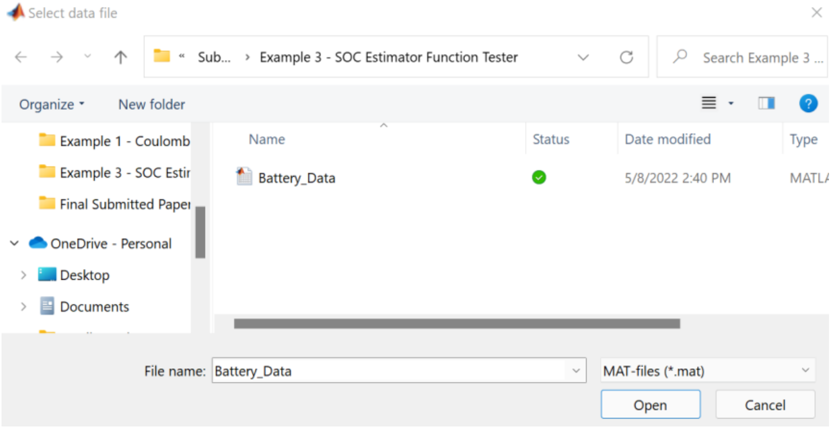


b. Open and run *SOC_Estimator_Test_Script.m*

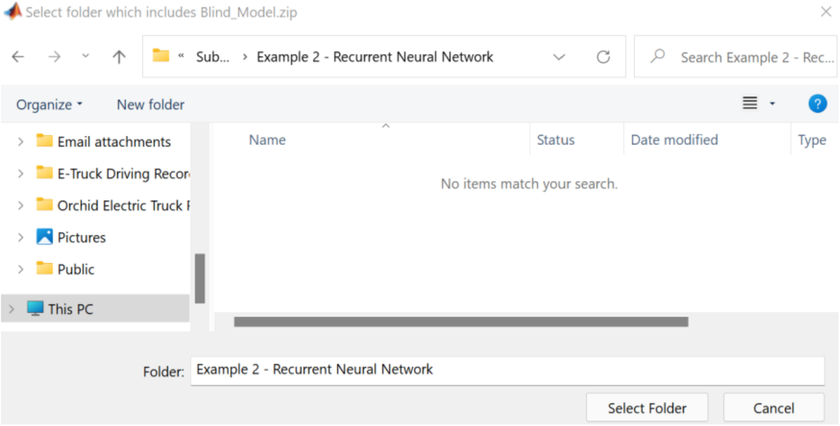


Step 4: Test *Blind_Model.zip* using *SOC_Estimator_Test_Script.m*

c. Select *.mat* file with battery data
(*Default: Battery_Data.mat)*



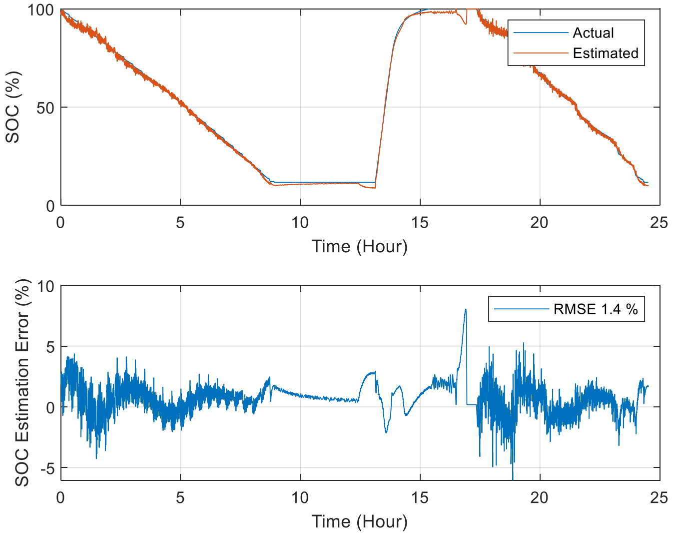
d. Select folder containing *Blind_Model.zip*



- Two working example folders provided:
  - Example 1 - Coulomb Counter
  - Example 2 - Recurrent Neural Network

Step 4: Test *Blind_Model.zip* using *SOC_Estimator_Test_Script.m*

e. If *Blind_Model.zip* and subfiles are configured correctly, results figure will display



- Results for following folder:

Example 2 - Recurrent Neural Network

f. Typical errors

Blind_Model.zip not present in selected folder

```
Error using checkfilename>validateFilename (line 157)
Function UNZIP was unable to find file ''Blind_Model.zip''.
```

Settings.xlsx not present in unzipped contents

```
Error using xlsread (line 136)
Unable to open file 'Settings.xlsx'
```

Model.m not present in unzipped contents

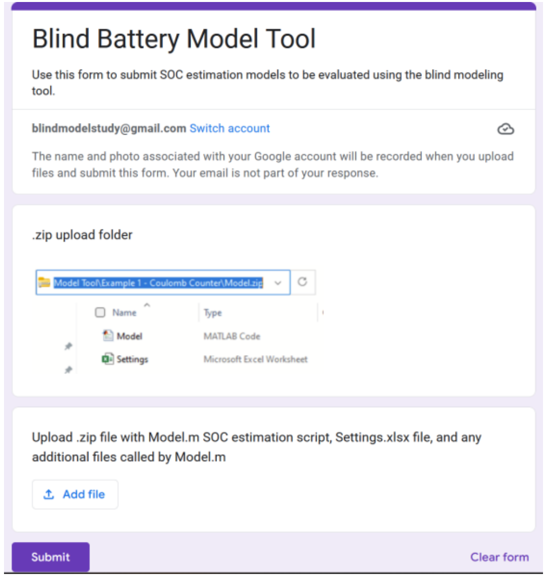
```
Undefined function 'Model' for input arguments of type 'double'.

Error in SOC_Estimator_Test_Script (line 31)
Y_est = Model(X);
```

Important: Model must work with *SOC_Estimator_Test_Script,* otherwise online submission will fail to return a result

Step 5: After testing zip file with script from Step 4, upload zip file using the Google Form

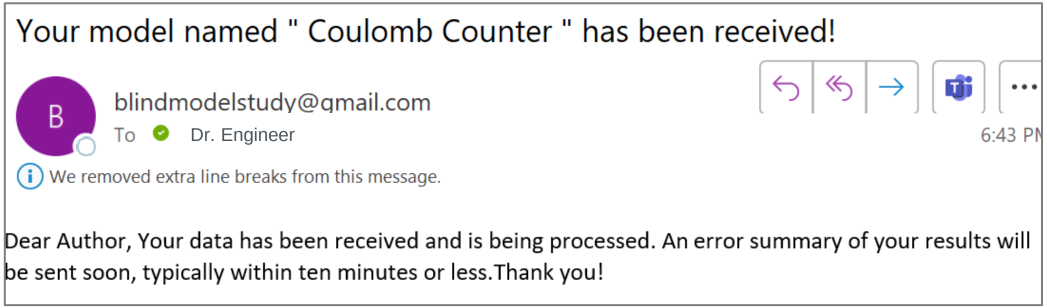
Google Form for Uploading Model



https://forms.gle/qvJHDzVfQaYgEFdg8

Step 6: Email sent to author confirming submission

System Automatically Sends Email Confirming Submission


Your model named " Coulomb Counter " has been received!

blindmodelstudy@gmail.com
To Dr. Engineer
6:43 PM

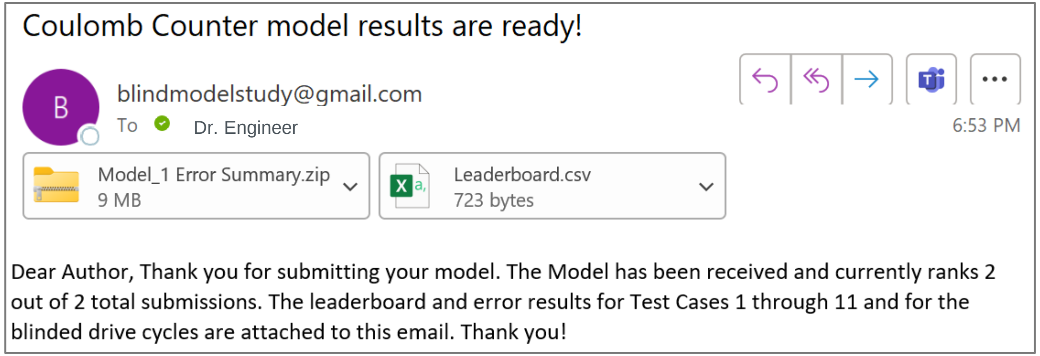
We removed extra line breaks from this message.

Dear Author, Your data has been received and is being processed. An error summary of your results will be sent soon, typically within ten minutes or less.Thank you!

- If email is not received within one hour, test system is likely down
- Send email to kollmeyp@mcmaster.ca (Phillip Kollmeyer) requesting support

Step 7: Email sent to author with results

System Automatically Sends Email with Results


Coulomb Counter model results are ready!

blindmodelstudy@gmail.com
To Dr. Engineer
6:53 PM

Model_1 Error Summary.zip
9 MB

Leaderboard.csv
723 bytes

Dear Author, Thank you for submitting your model. The Model has been received and currently ranks 2 out of 2 total submissions. The leaderboard and error results for Test Cases 1 through 11 and for the blinded drive cycles are attached to this email. Thank you!

- If email is not received within one hour **and** confirmation email (step 6) was received, then the submitted model has an error
- Investigate and correct cause of error and re-submit (see step 4)
- If issue cannot be resolved, send email to kollmeyp@mcmaster.ca (Phillip Kollmeyer) requesting support

Step 8: View results sent in *Leaderboard.csv* and *Error Summary.zip*

***Error Summary.zip* Folder contains results for multiple test cases**

Error Summary

Name	Type
Drive_Cycle_Error_Summary_Table	MATLAB Data
Test 01 to 08 RMS Error_Bar Plot	MATLAB Figure
Test 02 and 03_Blind vs Non-Blind Cells_Time Domain Plot	MATLAB Figure
Test 04 Charging and Cycle_Time Domain Plot	MATLAB Figure
Test 07 and 08 Standard vs Non-Standard Cycles_Time Domain Plot	MATLAB Figure
Test 09 Error vs Temperature_Time Domain Plot	MATLAB Figure
Test 09 RMS Error vs Temperature_Bar Plot	MATLAB Figure
Test 10 Initial SOC Error_Time Domain Plot	MATLAB Figure
Test 11 Current Sensor Offset Error_Time Domain Plot	MATLAB Figure

- See ITEC 2022 paper for complete description of test cases: *A Blind Modeling Tool for Standardized Evaluation of Battery State of Charge Estimation Algorithms*

Step 8: View results sent in *Leaderboard.csv*

Leaderboard ranks all submissions by their "weighted" error

Results for Coulomb Counter Example

Test Case		Weighting	RMSE	Weighted Error
2	Blinded Cell	10%	1.4%	0.1%
3	Non-Blinded Cells	10%	2.4%	0.2%
4	Charging (m80)	10%	94.1%	9.4%
5	1000kg Payload (m1000)	10%	2.3%	0.2%
8	Non-Standard Cycles HWGRADE, HWCUST (m1000)	10%	2.8%	0.3%
9	-20°C (m80)	10%	3.3%	0.3%
10	Initial SOC offset 0°C (m80)	10%	45.5%	4.6%
10	Initial SOC offset -20°C (m80)	10%	23.2%	2.3%
11	0.3A offset 10°C (m1000)	10%	24.5%	2.5%
11	-0.1A offset -10°C (m1000)	10%	1.4%	0.1%
			Weighted error	20.1%

Step 8: View results sent in *Leaderboard.csv*

Leaderboard includes:

- **Error values for all tests which make up "Weighted Error"**
- **For all blinded drive cycles: rms, mae, max error**

	A	B	C	D	E	F	G
1	**Submission_Time**	**Author**	**Affiliation**	**Model Name**	**Weighted Error**	**Test 2**	**Test 3**
2	12/13/2022 18:40	Phillip Kollmeyer	McMaster University	LSTM Neural Network	3.28	1.45	2.87
3	12/13/2022 21:00	Junran Chen	McMaster University	Feedforward Neural Network	4.13	1.29	3.28
4	1/7/2023 17:52	Phillip Kollmeyer	McMaster University	Coulomb Counter	20.14	1.4	2.44

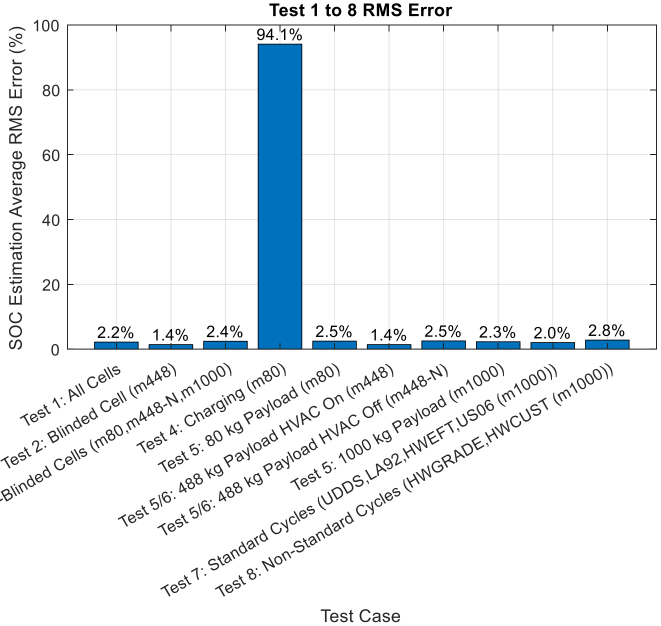
H	I	J	K	L	M	N	O	P	Q	R
Test 4	**Test 5**	**Test 8**	**Test 9**	**Test 10 0degC**	**Test 10 -20degC**	**Test 11 0.3 A offset**	**Test 11 - 0.1A offset**	**All drive cycles RMSE**	**All drive cycles MAE**	**All drive cycles max error**
0.92	2.97	3.94	2.17	1.83	6.14	4.58	5.96	2.75	2.17	10.58
1.04	3.5	6.29	5.42	2.41	9.91	4.05	4.08	3.04	2.46	9.69
94.08	2.29	2.8	3.32	45.51	23.23	24.5	1.88	2.38	2.05	3.97

Weighted Error (row 2) = 0.1*sum(F2:O2)

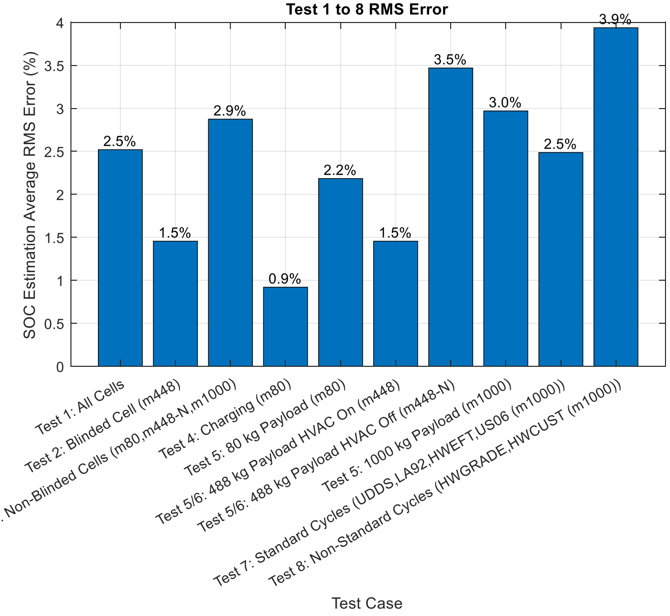
Step 8: View results sent in *Error Summary.zip*

Test 01 to 08 RMS Error_Bar Plot.fig

Coulomb Counting Example



LSTM Example

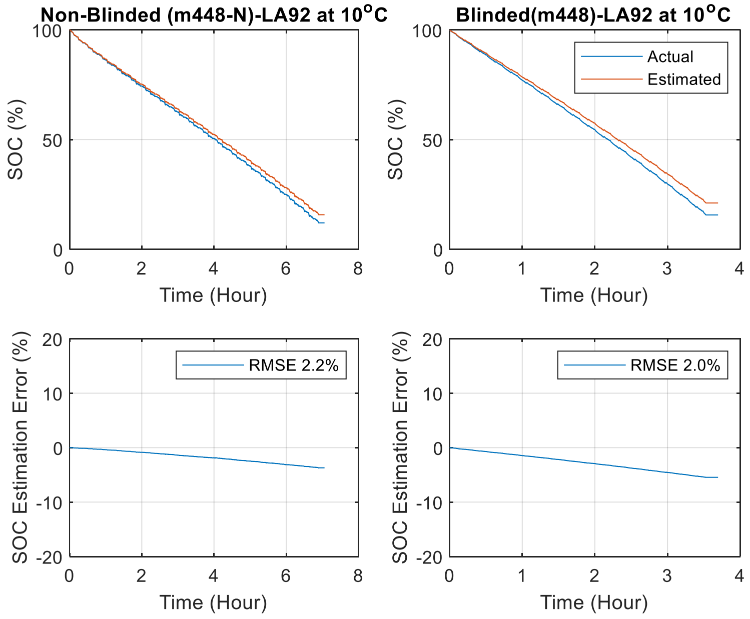


Charging error high for coulomb counting because charge starts around 0% SOC, and coulomb counting code assumes all test cycles start at 100% SOC

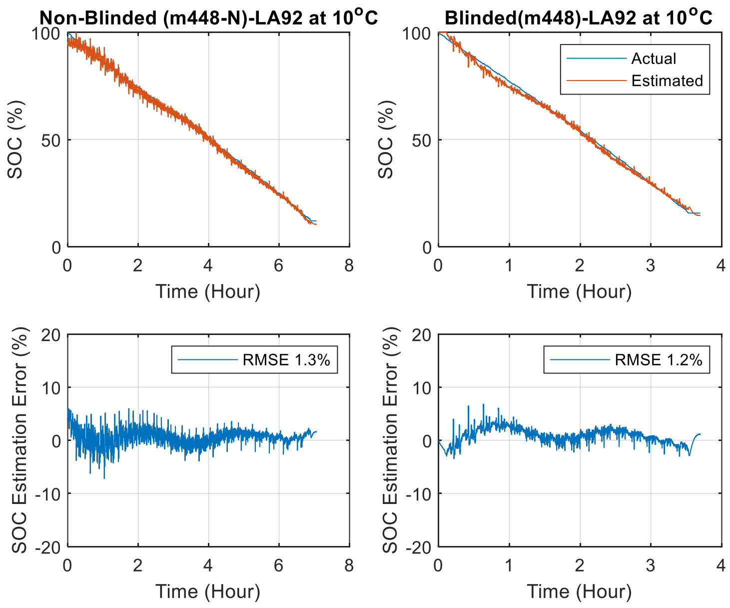
Step 8: View results sent in *Error Summary.zip*

Test 02 and 03_Blind vs Non-Blind Cells_Time Domain Plot.fig

Coulomb Counting Example



LSTM Example

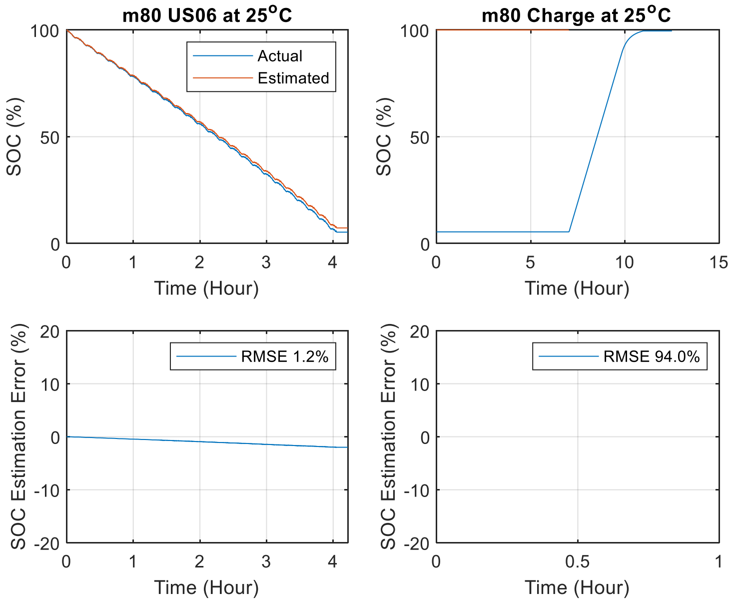


LSTM more accurate, but somewhat noisy

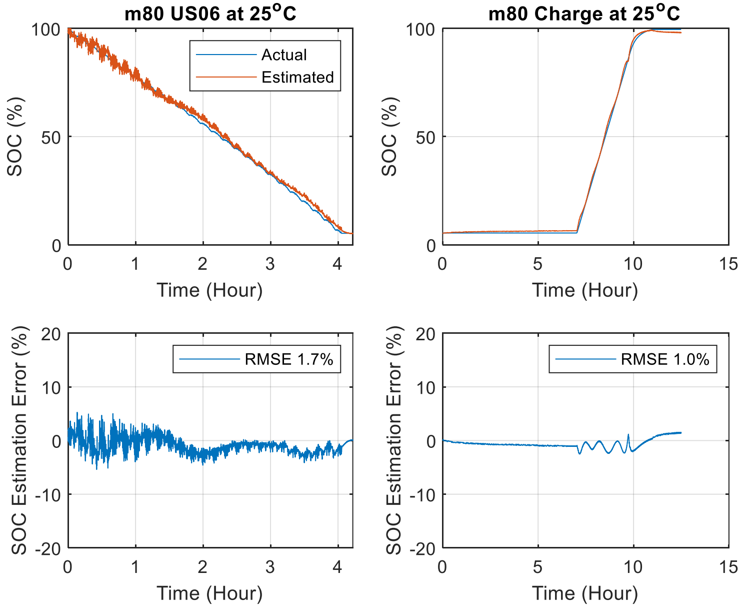
Step 8: View results sent in *Error Summary.zip*

Test 04 Charging and Cycle_Time Domain Plot.fig

Coulomb Counting Example



LSTM Example

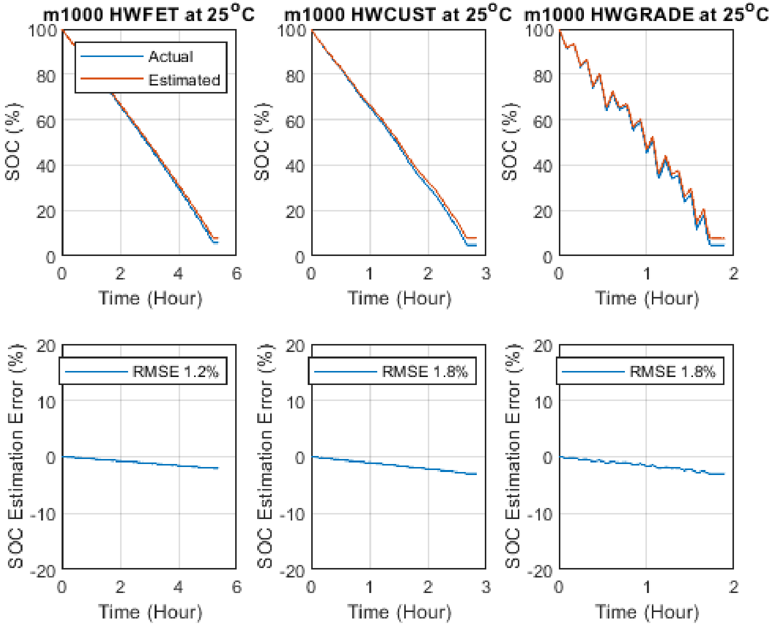


Charging case starts at 0% SOC – causing high error for coulomb counting which assumes 100% SOC at start of file

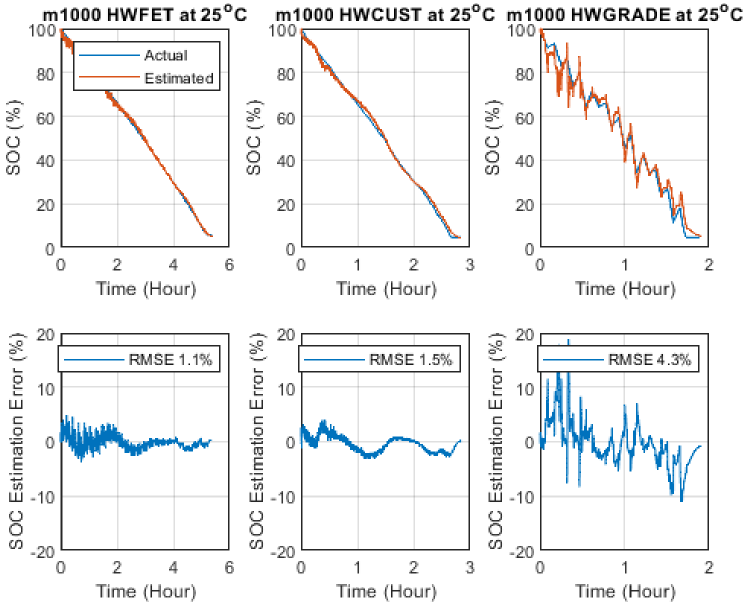
Step 8: View results sent in *Error Summary.zip*

Test 07 and 08 Standard vs Non-Standard Cycles_Time Domain Plot.fig

Coulomb Counting Example



LSTM Example

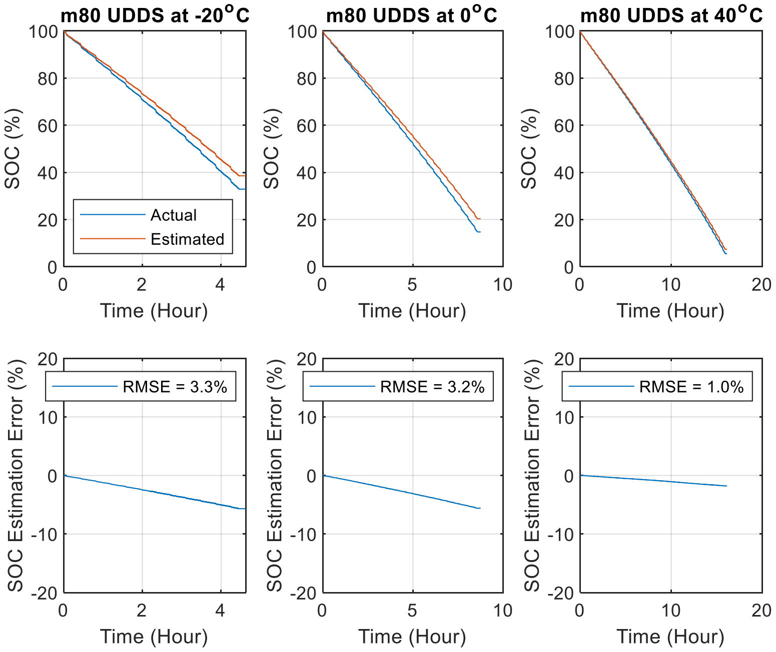


LSTM performs well, except for HWGRADE where high power results in high error

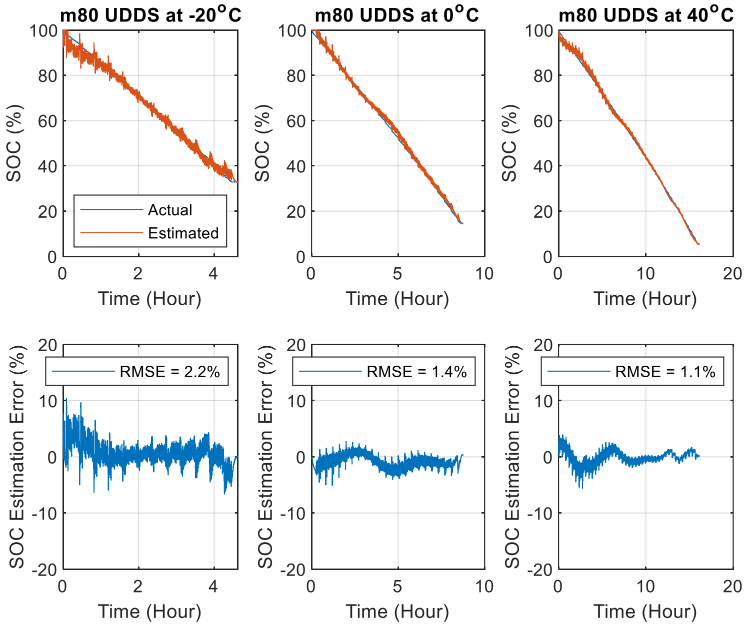
Step 8: View results sent in *Error Summary.zip*

*Test 09 Error vs Temperature_**Time Domain** Plot.fig*

Coulomb Counting Example



LSTM Example

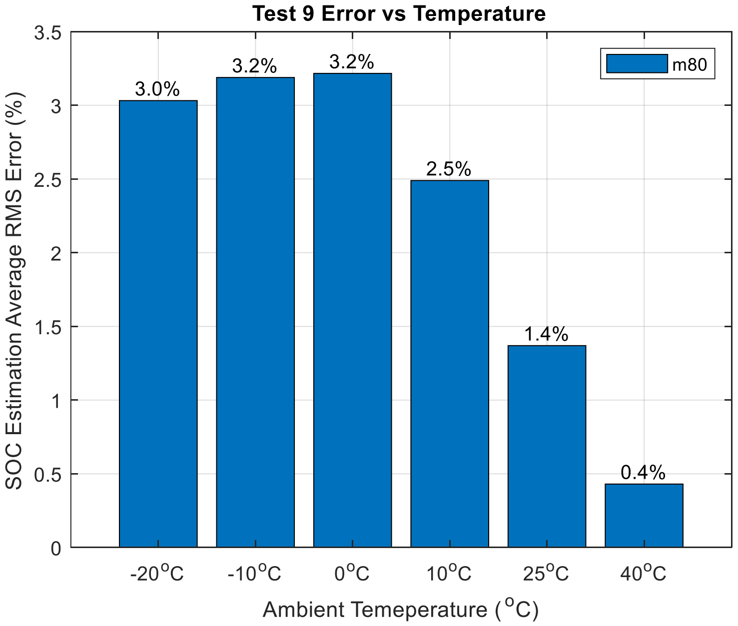


Coulomb counting error higher at low temperature due to cell aging (cold temperature tests done last)
LSTM error higher at low temperature due to nonlinear, higher resistance of cell at low temperatures

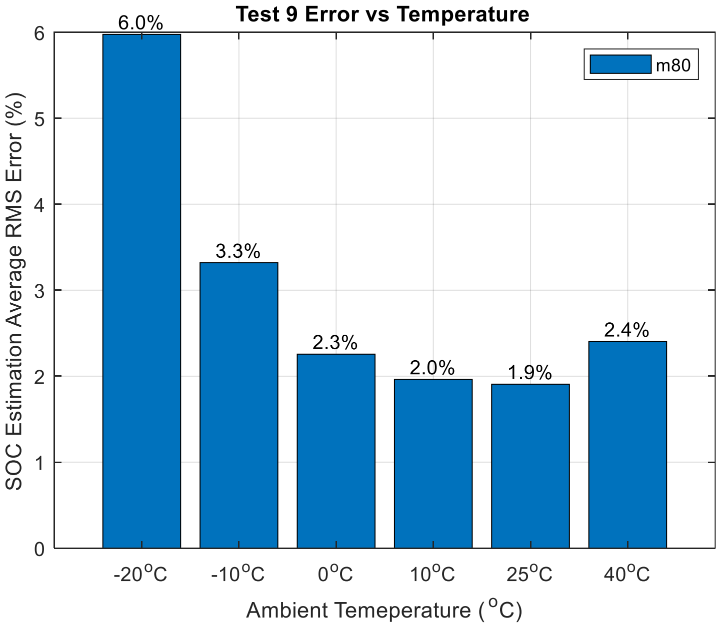
Step 8: View results sent in *Error Summary.zip*

*Test 09 RMS Error vs Temperature_**Bar** Plot.fig*

Coulomb Counting Example



LSTM Example

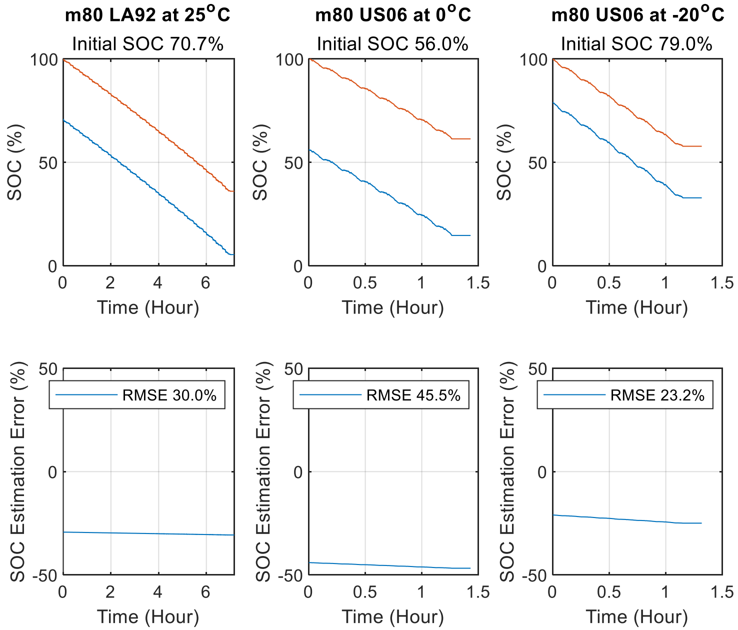


Coulomb counting error higher at low temperature due to cell aging (cold temperature tests done last)
LSTM error higher at low temperature due to nonlinear, higher resistance of cell at low temperatures

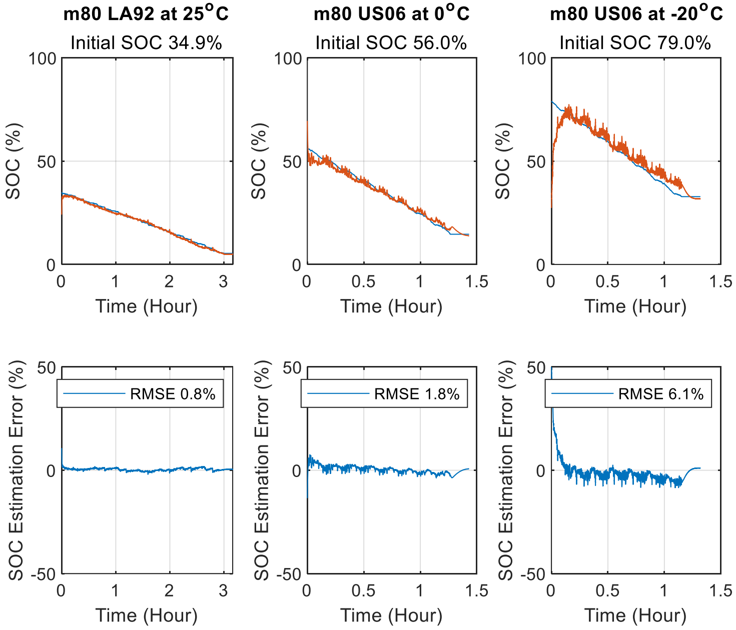
Step 8: View results sent in *Error Summary.zip*

Test 10 Initial SOC Error_Time Domain Plot.fig

Coulomb Counting Example



LSTM Example

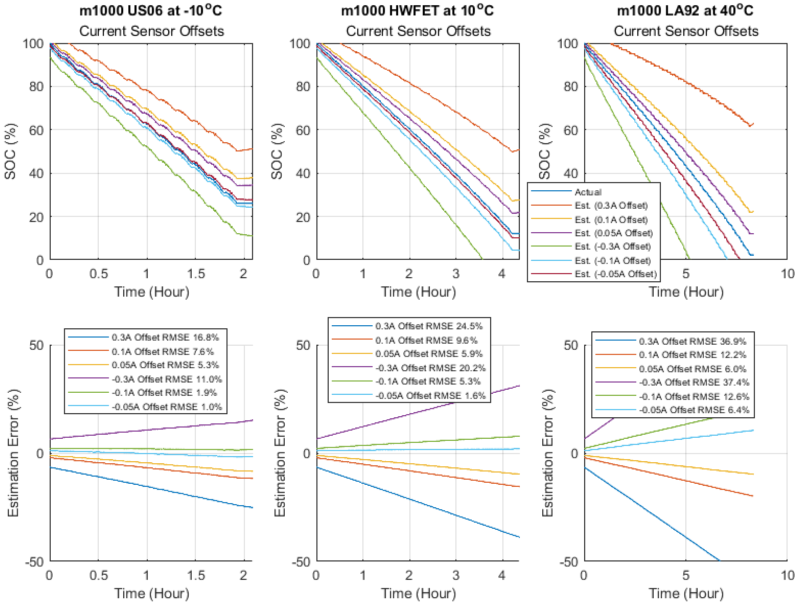


Coulomb counting can't adjust to initial SOC (assumes 100% at start of file)
LSTM is able estimate the correct SOC value

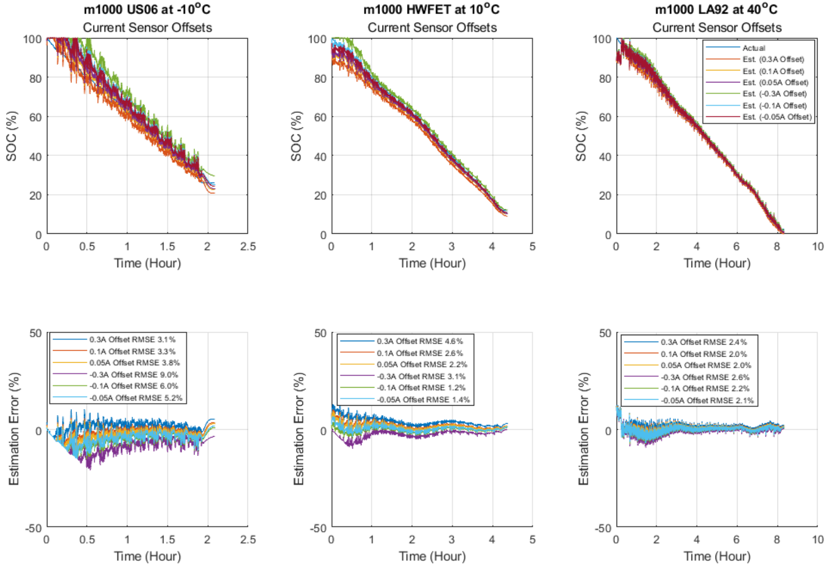
Step 8: View results sent in *Error Summary.zip*

Test 11 Current Sensor Offset Error_Time Domain Plot.fig

Coulomb Counting Example



LSTM Example



Coulomb counter integrates current sensor error, LSTM handles offset fairly well at higher temperatures

Step 8: View results sent in *Error Summary.zip*

"Error Summary Table" Figure

Coulomb Counting Example

Drive_Cycle_Error_Summary_Table

132x4 table

	1 Label	2 RMSE	3 MAE	4 MAXE
1	'UDDS _ -10 ^oC _ M80'	3.2260	2.7744	5.5621
2	'HWFET _ -10 ^oC _ M80'	3.3858	2.9177	5.6672
3	'LA92 _ -10 ^oC _ M80'	3.2243	2.7581	5.5813
4	'US06 _ -10 ^oC _ M80'	3.4232	2.9428	5.7002
5	'HWCUST2 _ -10 ^oC _ M80'	3.8628	3.3559	6.1761
6	'HWGRADE2 _ -10 ^oC _ M80'	3.8796	3.3524	6.2650
7	'UDDS _ -20 ^oC _ M80'	3.3221	2.8653	5.6698

LSTM Example

Drive_Cycle_Error_Summary_Table

132x4 table

	1 Label	2 RMSE	3 MAE	4 MAXE
1	'UDDS _ -10 ^oC _ M80'	1.3161	1.0187	6.5343
2	'HWFET _ -10 ^oC _ M80'	2.7513	2.3746	9.0097
3	'LA92 _ -10 ^oC _ M80'	1.4657	1.1132	7.6461
4	'US06 _ -10 ^oC _ M80'	1.9980	1.5584	7.3933
5	'HWCUST2 _ -10 ^oC _ M80'	2.2954	2.0961	4.4061
6	'HWGRADE2 _ -10 ^oC _ M80'	2.8434	2.2003	12.6412
7	'UDDS _ -20 ^oC _ M80'	2.1709	1.5759	10.4202

Drive cycle error values can be used for making general comparisons and finding trouble cases (high max error, etc)

Phillip Kollmeyer, *Ph.D., Member IEEE*
Senior Principal Research Engineer
McMaster University

kollmeyp@mcmaster.ca
electrification.mcmaster