REPORT ON DIFFERENT OCV MODELING OF LI-ION BATTERY

Advanced Energy Storage Systems

ELEC-8900-30-R-2022F

Prepared By: Arpanbhai Patel
(110088169)

Submitted To: Dr. Balasingam

Date: Dec 5, 2022

Contents

Abstract:

The report indicated the various models of linear batteries and makes a contrasting the characteristics of the State of Charge (SOC), Open Circuit Voltage(OCV) with the other four battery cells. Different errors were also eliminated along with the SOC and OCV. According to that data and results, the raking table and the metrics table were estimated. For finalize the best model, errors and the linear models were computed. SOC for different models is studied along with the properties of the OCV, this makes it easy to determine the remaining charge of the batterie and the electrical potential capability.

1. Data Plotting for the SOC and OCV Data:

Open circuit voltage (OCV) and state of charge of the cell are not linearly correlated (SOC). The voltage across battery terminals decreases with its usage as the battery's charge is decreased. So, as a result, the graph of the battery's state of charge (SOC) and the voltage between its terminals (OCV) appears non-linear.

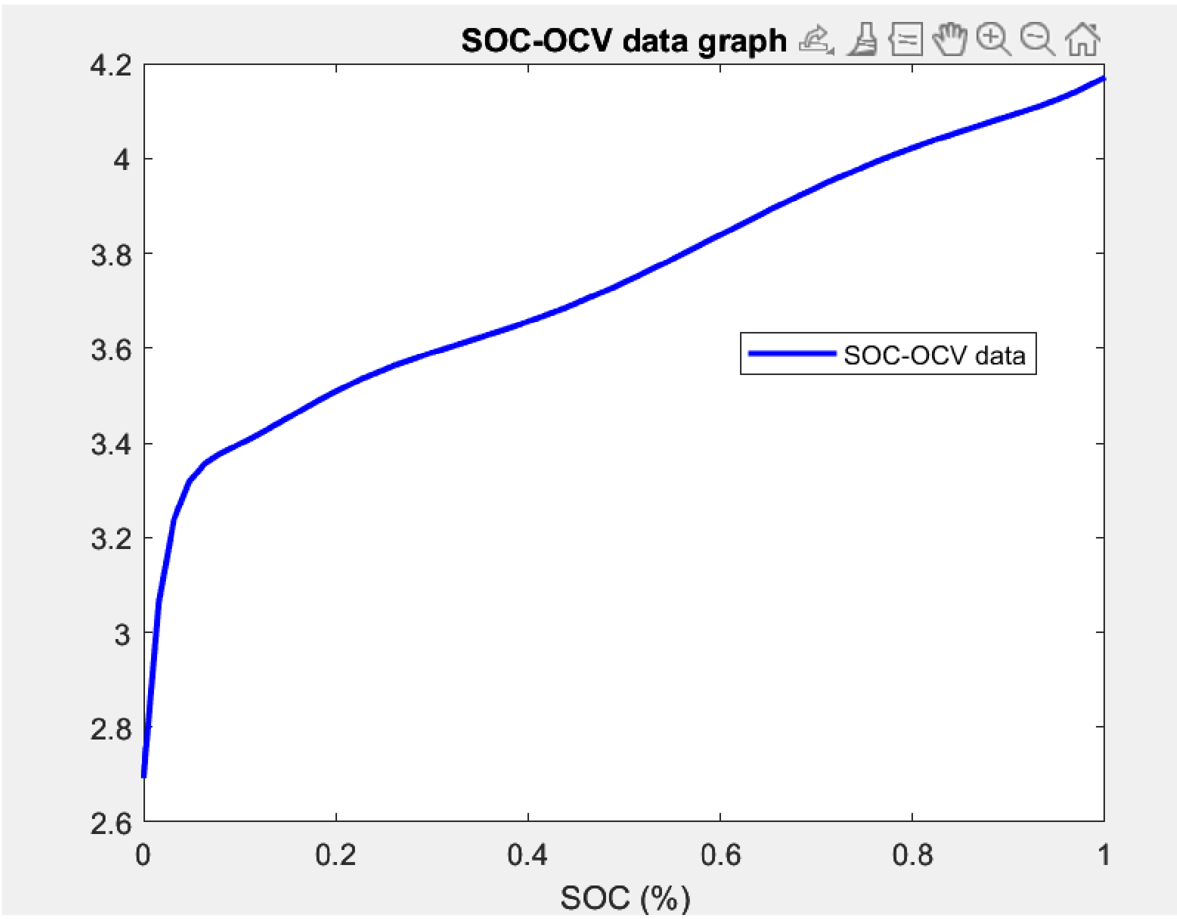


Figure 1 OCV-SOC Graph of Battery.

2. Design of Different Linear OCV models:

A linear OCV model satisfies,

$$v = V_0(s) = p(s)^T k \qquad (1)$$

where $V_0(s)$ denotes the OCV $s \in [0, 1]$ denotes the SOC
$p(s)^T$ = row vector of linear and non-linear functions of S
k = OCV parameter vector.

The data of SOC and OCV values have been given to us in the excel sheet. First, we need to plot the graph for each OCV linear model with the help of MATLAB programming.

<u>These are the different linear OCV models, which will be discussed in the further report.</u>

- Unnewehr universal model (or simply, linear model)

$$\mathbf{p}(s)^T = [1 \quad s] \tag{2}$$

- Shepherd model

$$\mathbf{p}(s)^T = \begin{bmatrix} 1 & \frac{1}{s} \end{bmatrix} \tag{3}$$

- Nernst model

$$\mathbf{p}(s)^T = [1 \quad \ln(s) \quad \ln(1-s)] \tag{4}$$

- Combined model

$$\mathbf{p}(s)^T = \begin{bmatrix} 1 & \frac{1}{s} & s & \ln(s) & \ln(1-s) \end{bmatrix} \tag{5}$$

- Combined+3 model

$$\mathbf{p}(s)^T = \begin{bmatrix} 1 & \frac{1}{s} & s & \frac{1}{s^2} & \frac{1}{s^3} & \frac{1}{s^4} & \ln(s) & \ln(1-s) \end{bmatrix} \tag{6}$$

- Polynomial model

$$\mathbf{p}(s)^T = [1 \quad s \quad \dots \quad s^m \qquad s^{-1} \quad \dots \quad s^{-n}] \tag{7}$$

- Exponential model

$$\mathbf{p}(s)^T = [1 \quad e^s \quad \dots \quad e^{s^m} \qquad e^{-s} \quad \dots \quad e^{-s^n}] \tag{8}$$

First, for ease of the MATLAB program, I need to import the data of SOC and OCV from the excel sheet. So, the battery number can be entered to get the desired data from an excel sheet. '**xlsread command'** is used to take the data of 4 batteries to MATLAB.

Import data into MATLAB from Excel Sheet:

For getting the graph for the SOC-OCV data model, error metrics, and providing data ranking for different linear OCV models, we need proper MATLAB code along with the data of SOC-OCV for all mentioned batteries.

So, first, we must load all the data for each battery into the MATLAB software from the already given excel file. We can use either the xlsread or readmatrix commands to import the data from the cells we desire for the equations.

This code has input string in which the user needs to enter the battery cell number and the code will fetch the data from the excel file as per the battery cell number and give it to the output to the user.

MATLAB code for importing data:

Input:

```
data = input ('Enter the battery number: ', 's')
if     data == 'C1202'
    soc=xlsread('C:\Users\Chandrakant\Desktop\Sem 3\AES\Project\Project sample\Data.csv','A3:A34')
    ocv=xlsread('C:\Users\Chandrakant\Desktop\Sem 3\AES\Project\Project sample\Data.csv','B3:B34')
elseif data == 'C1203'
    soc=xlsread('C:\Users\Chandrakant\Desktop\Sem 3\AES\Project\Project sample\Data.csv','D3:D34')
    ocv=xlsread('C:\Users\Chandrakant\Desktop\Sem 3\AES\Project\Project sample\Data.csv','E3:E34')
elseif data == 'C1204'
    soc=xlsread('C:\Users\Chandrakant\Desktop\Sem 3\AES\Project\Project sample\Data.csv','G3:G34')
    ocv=xlsread('C:\Users\Chandrakant\Desktop\Sem 3\AES\Project\Project sample\Data.csv','H3:H34')
else
    SOC=xlsread('C:\Users\Chandrakant\Desktop\Sem 3\AES\Project\Project sample\Data.csv','J3:J34')
    OCV=xlsread('C:\Users\Chandrakant\Desktop\Sem 3\AES\Project\Project sample\Data.csv','K3:K34')
end
```

Output:

```
Command Window
  >> code1
fx Enter the battery number:
```

We must have the scaled value of the SOC for plotting the linear models of OCV.
Here, we shall assume the value epsilon (E) as 0.175

zs = soc*(1-2*eps) + eps;
zs acts as a scaled SOC.

```
%Importing data from Excel(avoiding first 2 row)
A=readmatrix('Data.csv','NumHeaderLines',2);
soc=A(:,1); % SOC data From Battery C1202
v=A(:,2); % OCV data of Battery C1202
eps = 0.175;
zs = soc*(1-2*eps)+eps; % Scaled value of SOC in terms of zs
```

For plotting the graph for different models for every four batteries, we must calculate the K and OCV for all 8 models.

2.1. Unnewehr Universal Model:

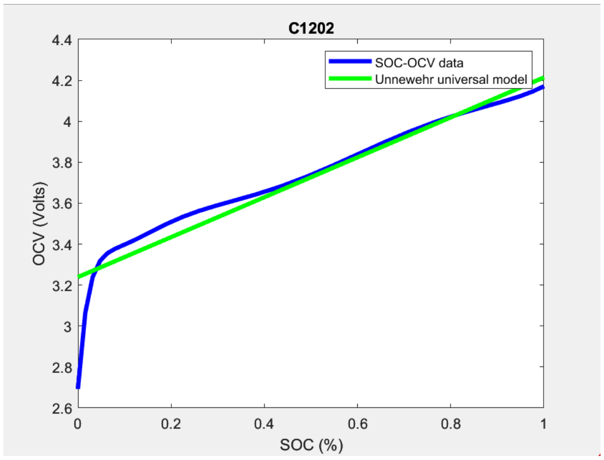
MATLAB code for plotting Unnewehr Universal Model:

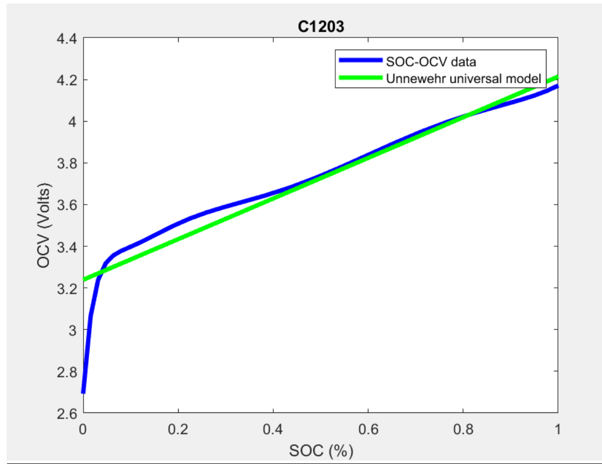
```
% Equation and graph for Unnewehr Universal model
p_u=[ones(length(soc),1) soc];
u=p_u';
k_u=inv(u*p_u)*(u*v); % Value of K
ocv_u= p_u*k_u; % calculating OCV for Unnewehr model
```

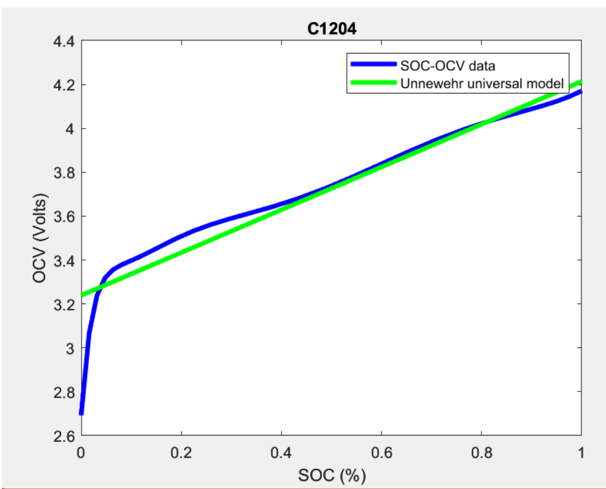
From the above MATLAB code, we can observe that, for plotting the Unnewehr model graph first, the equation of the Unnewehr model is written and then K also is calculated. The main purpose of writing this code is to calculate the OCV for Unnewehr universal model.

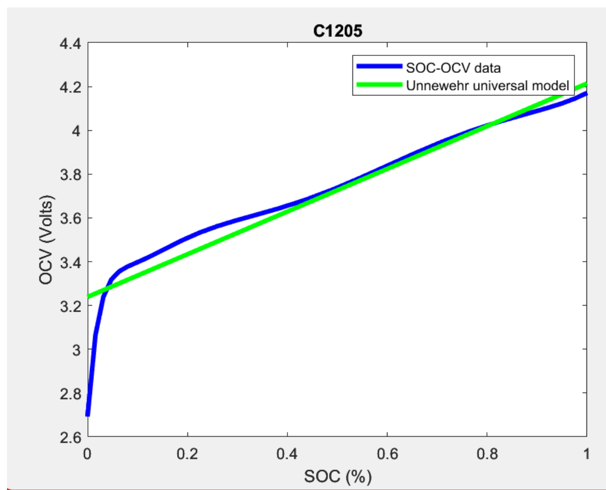
Below is the code to calculate the OCV and plot the graph for the Shepherd model.

Following is the graph of the Unnewehr Universal Model for different 4 batteries.









OCV Predication Error Metrics and Model Evaluation Metrics:

Predication of V is done using the error metrics values plus total number of model parameters.

Metrics for evaluating models consider the trade-off between the quantity number of points and the number of model parameters. According to model evaluation, the least squares equivalent of AIC are employed only when the models are fitted.

MATLAB code for Error Metrics:

```
% calculation for Error Metrics for Unnewehr Model
n_u = numel(ocv_u);
Vbar_u = (1/n_u)*(norm(v));
BF_u = (1-((norm(ocv_u-v))/(norm(v-Vbar_u))))*100; %bestfit value
R_squared_u = (1 - ((norm(ocv_u-v)).^2/(norm(v-Vbar_u).^2)))*100; %R square value
Merc_u= v-ocv_u;
Maxerror_u = max(Merc_u); %Max error value
error_u = immse(v,ocv_u);
rms_u = sqrt(error_u);
```

MATLAB code for Model Evaluation Metrics:

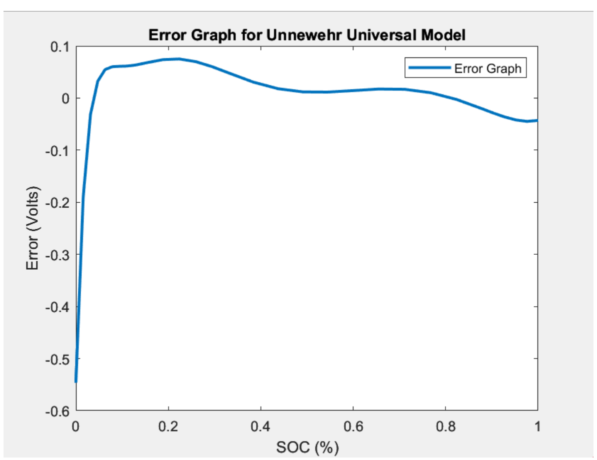
```
%Model evalution metrics
z_u = sum(Merc_u.^2);
M_u = numel(k_u);
AIC_u = n_u*log(z_u/n_u) + 2*(M_u + 1);
```

The following table indicates the value of error metrics and model evaluation metrics.

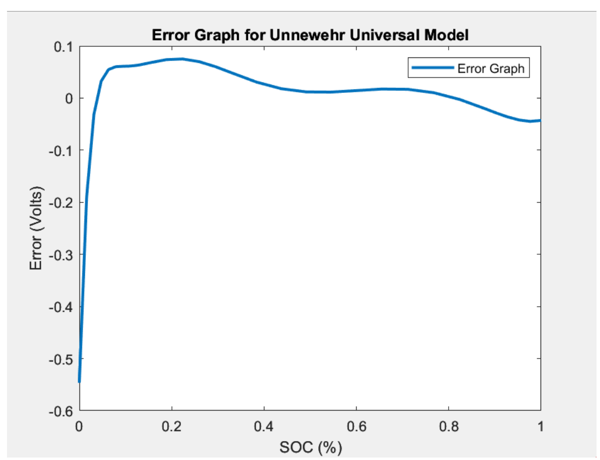
Metrics Table:

Battery No:	BF	R^2	Max Error	RMSE	AIC
C1202	96.2961	99.8628	0.0750	0.1118	-134.2380
C1203	96.2854	99.8655	0.0747	0.1121	-134.0416
C1204	96.8666	99.8666	0.0741	0.1102	-135.1353
C1205	96.5177	99.8786	0.0690	0.1054	-137.9861

Error Graph for Unnewehr model:

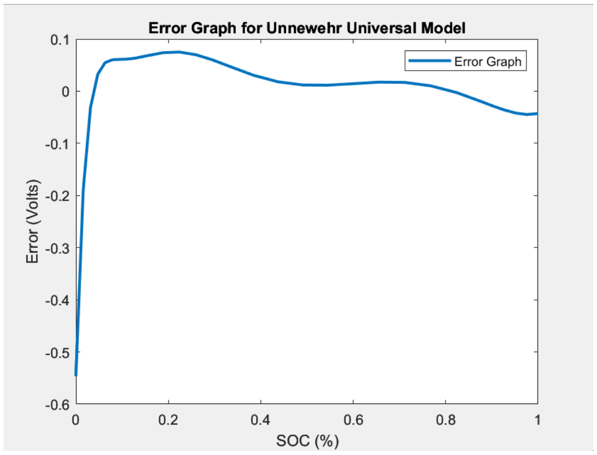


C1202

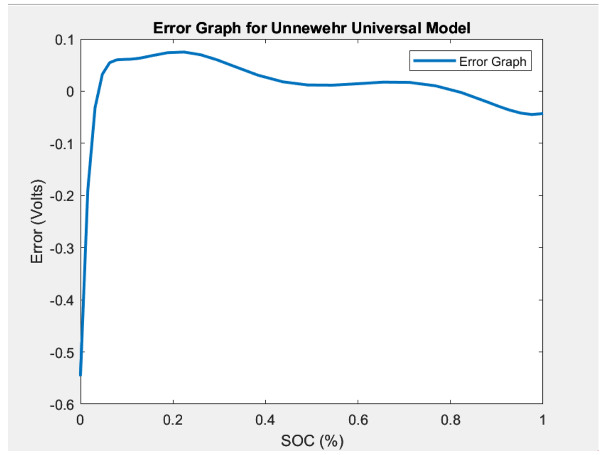


C1203

C1204



C1205

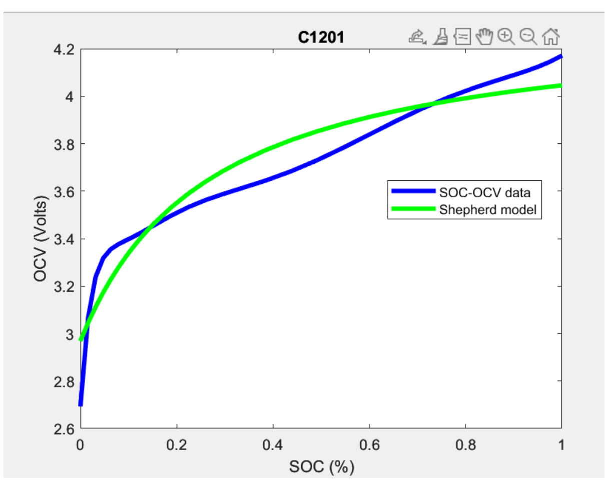


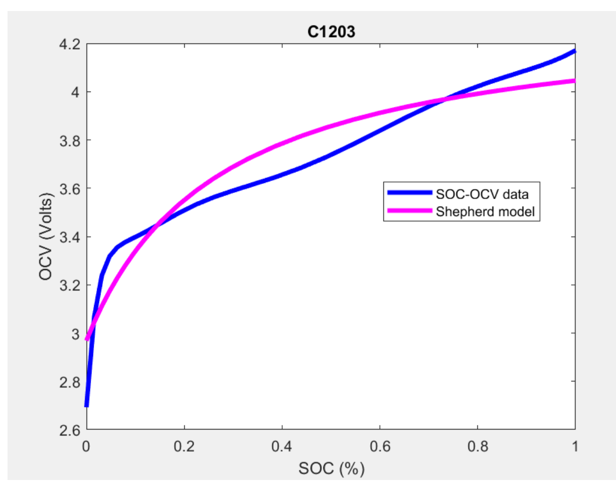
2.2. Shepherd Model:

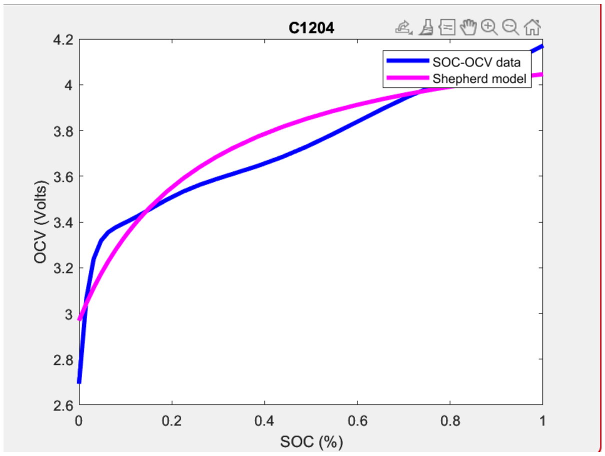
Below is the code to calculate the OCV and plot the graph for the Shepherd model.

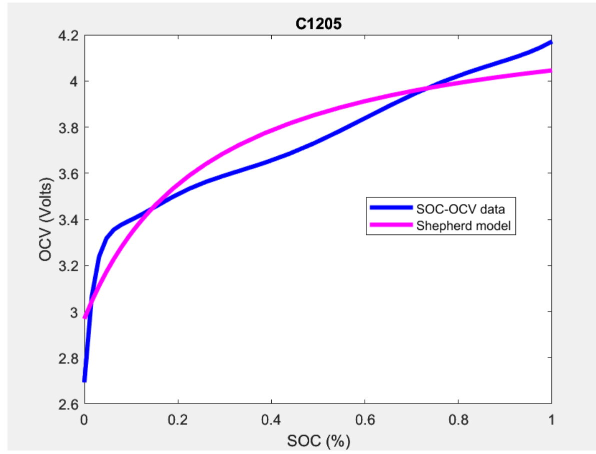
```
%Equation of Shepherd Model.
p_s=[ones(length(soc),1) 1./zs];
u=p_s';
k=inv(u*p_s)*(u*v);
ocv_s= p_s*k; % OCV value for Shepherd model.
```

Graph of Shepherd model of 4 battery cells.









MATLAB code for Error Metrics:

```
% calculation for Error Metrics for Shepherd Model
n_s = numel(ocv_s);
Vbar_s = (1/n_s)*(norm(v));
BF_s = (1-((norm(ocv_s-v))/(norm(v-Vbar_s))))*100;
R_squared_s = (1 - ((norm(ocv_s-v)).^2/(norm(v-Vbar_s).^2)))*100;
Merc_s= v-ocv_s;
Maxerror_s = max(Merc_s);
error_s = immse(v,ocv_s);
rms_s = sqrt(error_s);
```

MATLAB code for Model Evaluation Metrics:

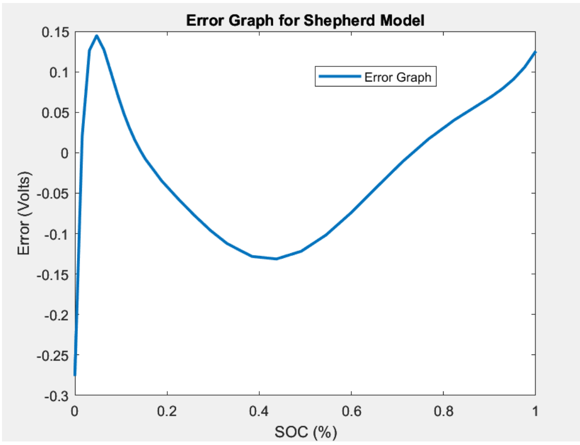
```
%Model evalution metrics
z_s = sum(Merc_s.^2);
M_s = numel(k);
AIC_s = n_s*log(z_s/n_s) + 2*(M_s + 1);
```

Metrics Table:

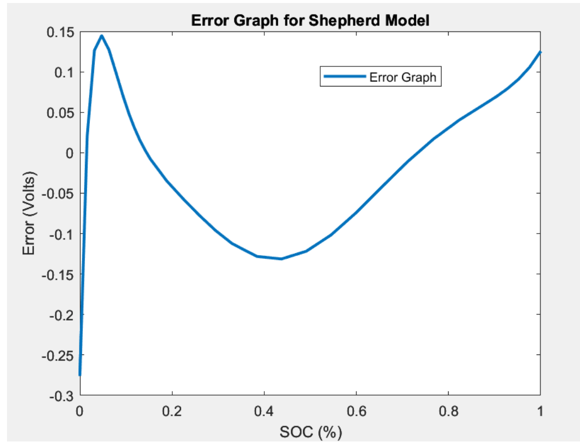
Battery No:	BF	R^2	Max Error	RMSE	AIC
C1202	96.8014	99.8977	0.1447	0.0965	-143.6238
C1203	96.8015	99.8976	0.1453	0.0965	-143.6151
C1204	96.8161	99.8986	0.1442	0.0962	-143.9101
C1205	96.8593	99.9014	0.1414	0.0951	-144.5950

Graph of Error Metrics:

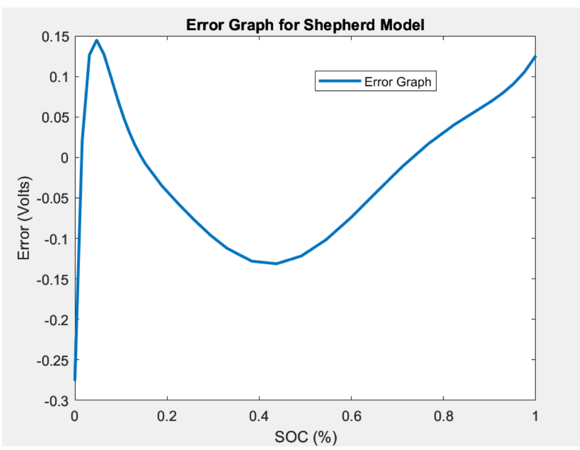
C1202



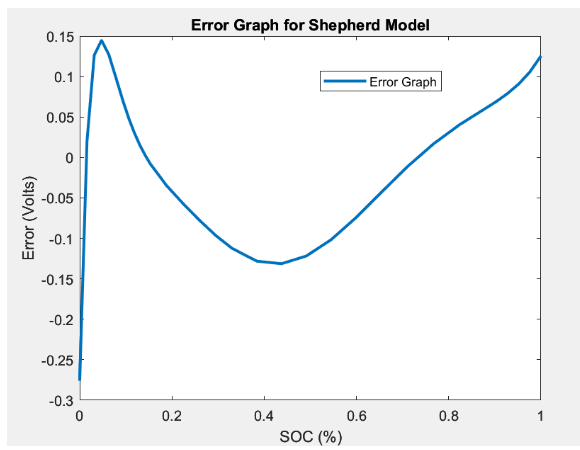
C1203



C1204



C1205

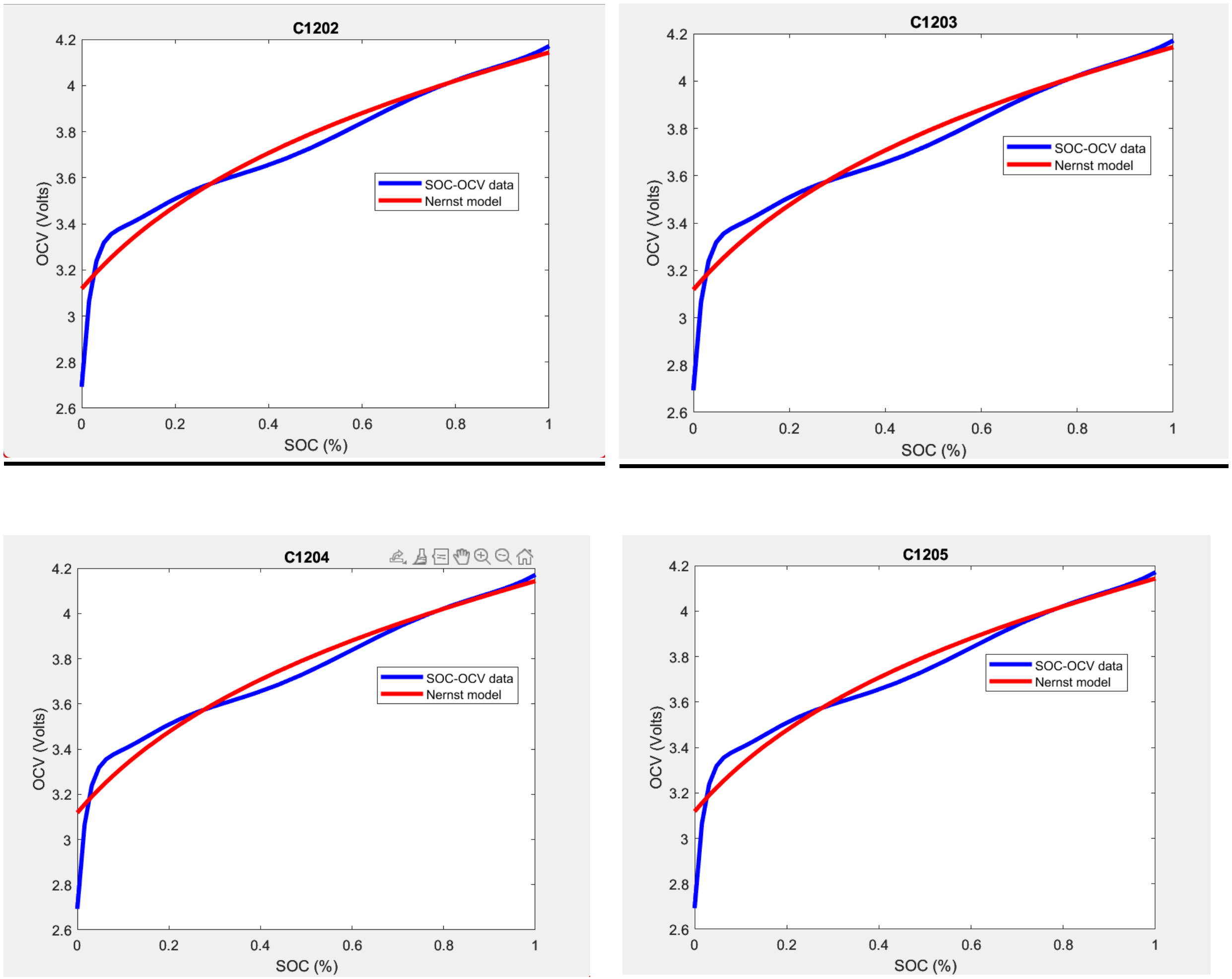


2.3. Nernst Model:

Below is the MATLAB code to calculate the OCV and plot the graph for the Nernst model.

```
%Equation of Nernst Model
p_n=[ones(length(soc),1) log(zs) log(1-zs)];
u=p_n';
k=inv(u*p_n)*(u*v);
ocv_n= p_n*k; %OCV value of Nernst Model
```

Graph of Nernst model of 4 battery cells.



<u>MATLAB code for Error Metrics:</u>

```
% calculation for Error Metrics for Nernst Model
n_n = numel(ocv_n);
Vbar_n = (1/n_n)*(norm(v));
BF_n = (1-((norm(ocv_n-v))/(norm(v-Vbar_n))))*100;
R_squared_n = (1 - ((norm(ocv_n-v)).^2/(norm(v-Vbar_n).^2)))*100;
Merc_n= v-ocv_n;
Maxerror_n = max(Merc_n);
error_n = immse(v,ocv_n);
rms_n = sqrt(error_n);
```

MATLAB code for Model Evaluation Metrics:

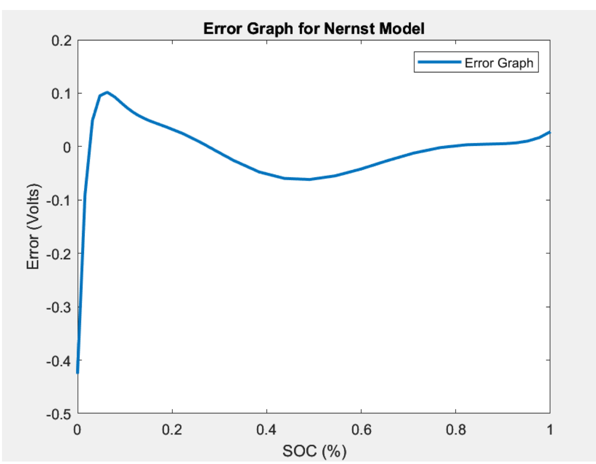
```
%Model evalution metrics
z_n = sum(Merc_n.^2);
M_n = numel(k);
AIC_n = n_n*log(z_n/n_n) + 2*(M_n + 1);
```

Metrics Table:

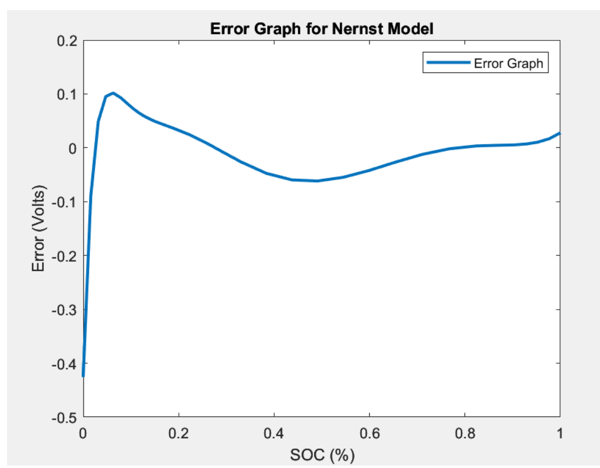
Battery No:	BF	R^2	Max Error	RMSE	AIC
C1202	96.9961	99.9098	0.1447	0.0907	-145.6438
C1203	96.9924	99.9095	0.1453	0.0908	-145.5532
C1204	97.0403	99.9124	0.1442	0.0893	-146.5850
C1205	97.1935	99.9212	0.1414	0.0850	-149.7950

Graph of Error Metrics:

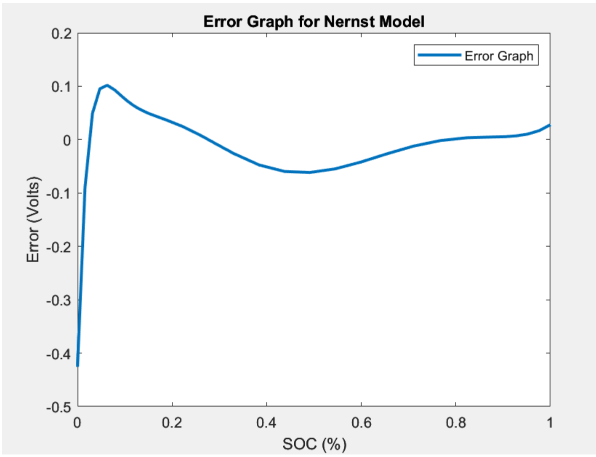
C1202



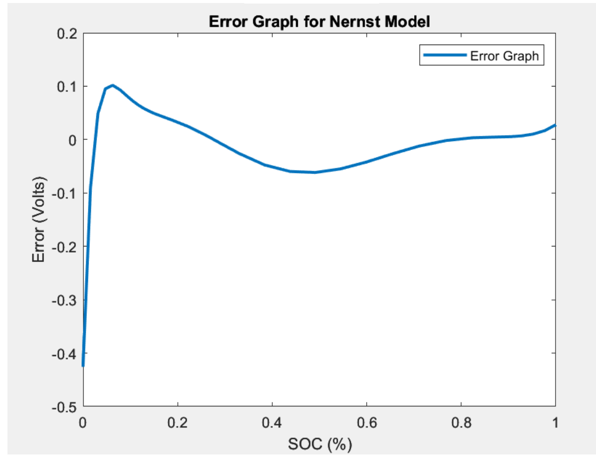
C1203



C1204



C1205

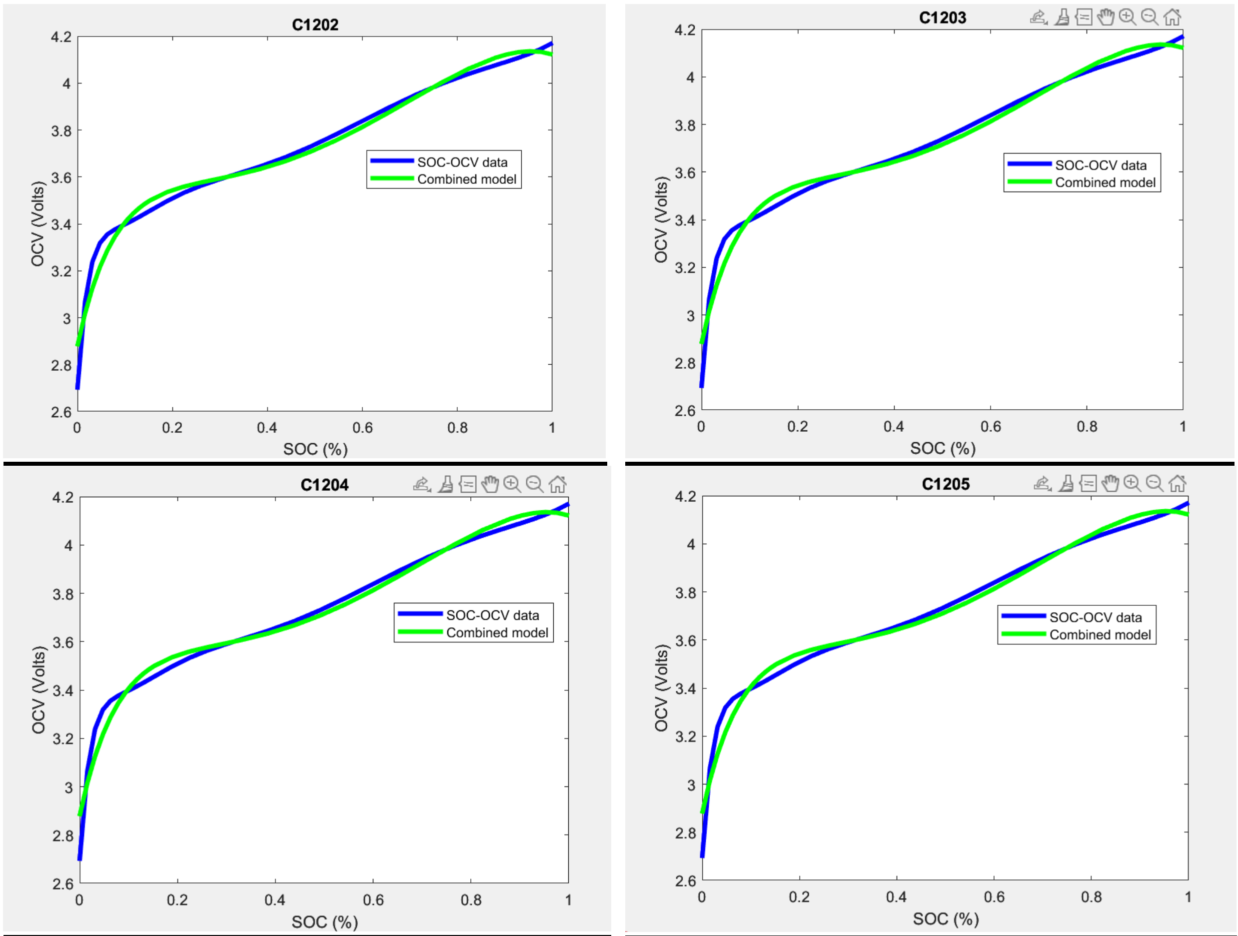


2.4. Combined Model:

Below is the code to calculate the OCV and plot the graph for the Combined model.

```
%Equation of Combined Model
p_c=[ones(length(soc),1) 1./zs zs log(zs) log(1-zs) ];
u=p_c';
k=inv(u*p_c)*(u*v);
ocv_c= p_c*k; %OCV value of combined model
plot(soc, v,'b',soc, ocv_c,'g','linewidth',3 );
xlabel("SOC (%)");
ylabel("OCV (Volts)");
title("C1202")
legend("SOC-OCV data","Combined model");
```

Graph of Combined model of 4 battery cells.



MATLAB code for Error Metrics:

```
% calculation for Error Metrics for Combined Model
n_c = numel(ocv_c);
Vbar_c = (1/n_c)*(norm(v));
BF_c = (1-((norm(ocv_c-v))/(norm(v-Vbar_c))))*100;
R_squared_c = (1 - ((norm(ocv_c-v)).^2/(norm(v-Vbar_c).^2)))*100;
Merc_c= v-ocv_c;
Maxerr_c = max(Merc_c);
error_c = immse(v,ocv_c);
rms_c = sqrt(error_c);
```

MATLAB code for Model Evaluation Metrics:

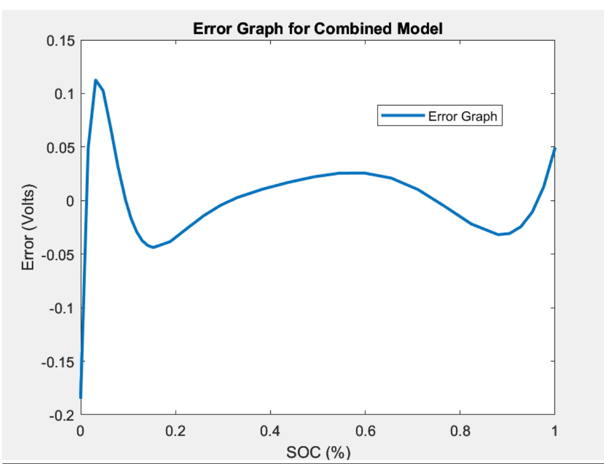
```
% Model Evaluation metrics
z_c = sum(Merc_c.^2);
M_c = numel(k);
AIC_c = n_c*log(z_c/n_c) + 2*(M_c + 1);
plot(soc,Merc_c,'c','linewidth',2);
```

Metrics Table:

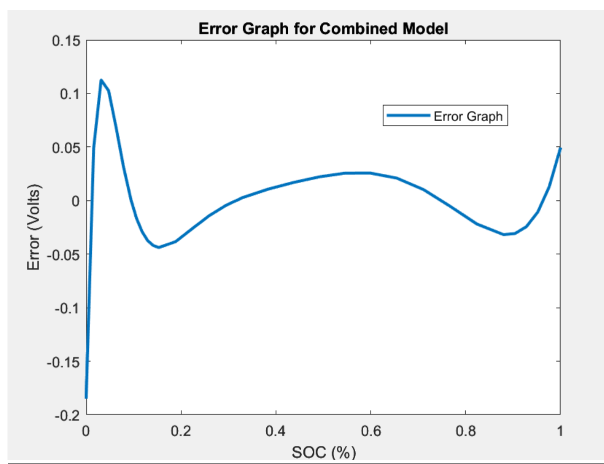
Battery No:	BF	R^2	Max Error	RMSE	AIC
C1202	98.3165	99.9718	0.1126	0.0507	-178.7006
C1203	98.3168	99.9717	0.1130	0.0508	-178.7011
C1204	98.3402	99.9725	0.1106	0.0501	-179.6014
C1205	98.4789	99.9769	0.1056	0.0460	-184.9944

Graph of Error Metrics:

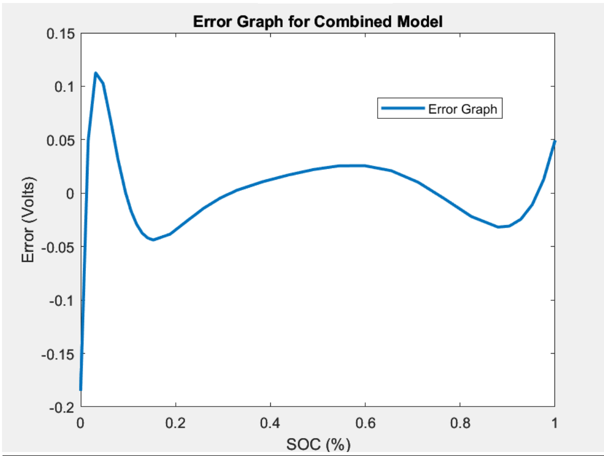
C1202



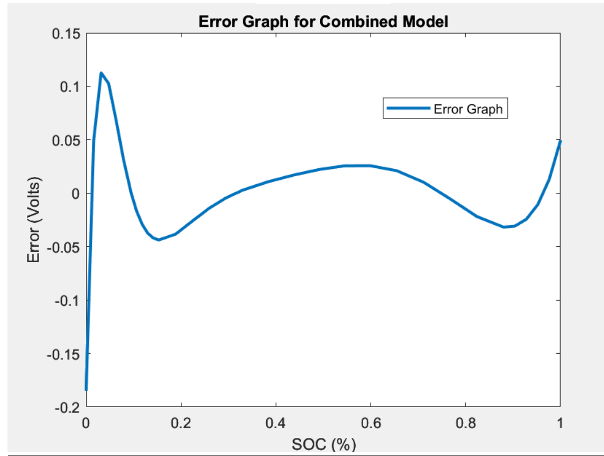
C1203



C1204



C1205

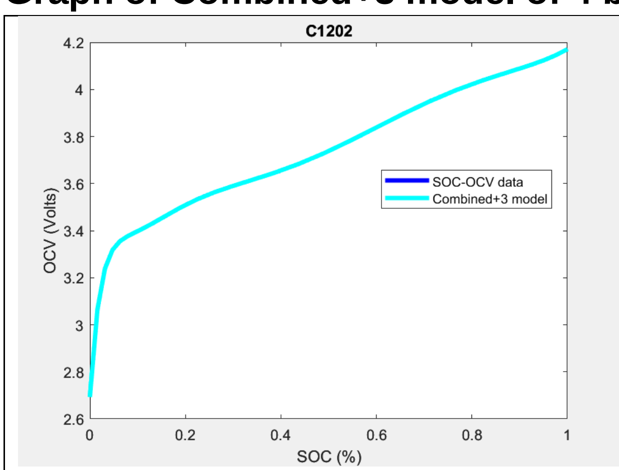


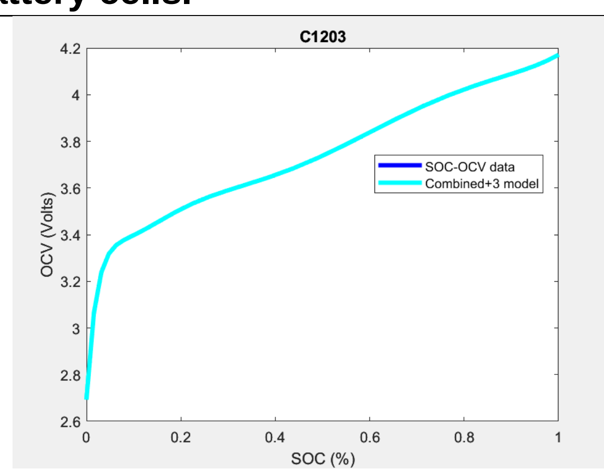
2.5. Combined+3 model:

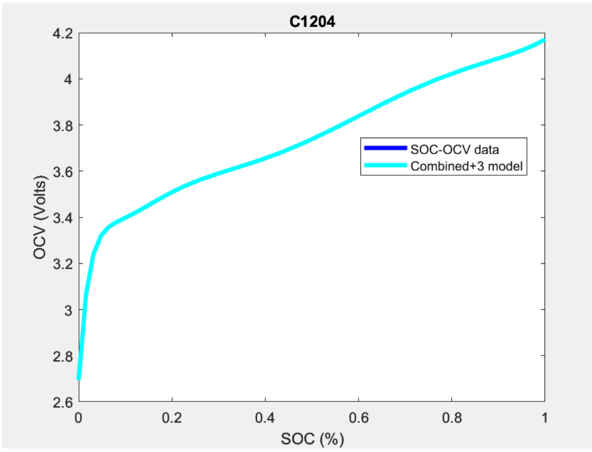
Below is the code to calculate the OCV and plot the graph for the Combined+3 model.

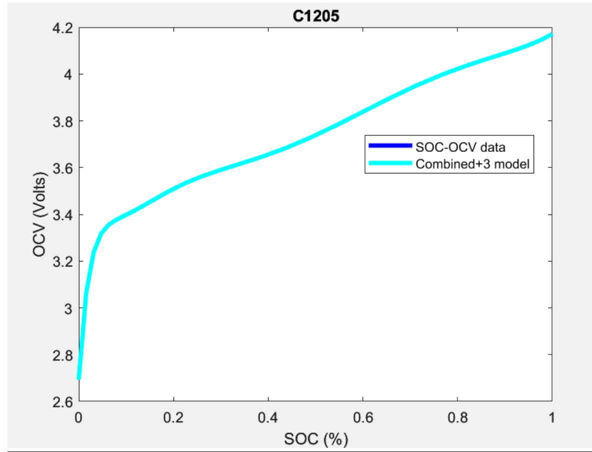
```
%Equation of combined+3 model.
p_c3=[ones(length(soc),1) 1./zs 1./zs.^2 1./zs.^3 1./zs.^4 zs log(zs) log(1 -zs)];
u=p_c3';
k=inv(u*p_c3)*(u*v);
ocv_c3= p_c3*k; %OCV value of combines+3 Model
plot(soc, v,'b',soc, ocv_c3,'c','linewidth',3 );
xlabel("SOC (%)");
ylabel("OCV (Volts)");
title("C1202")
legend("SOC-OCV data","Combined+3 model");
```

Graph of Combined+3 model of 4 battery cells.









MATLAB code for Error Metrics:

```
% calculation for Error Metrics for Combined+3 Model
n_c3 = numel(ocv_c3);
Vbar_c3 = (1/n_c3)*(norm(v));
BF_c3 = (1-((norm(ocv_c3-v))/(norm(v-Vbar_c3))))*100;
R_squared_c3 = (1 - ((norm(ocv_c3-v)).^2/(norm(v-Vbar_c3).^2)))*100;
Merc_c3= v-ocv_c3;
Maxerror_c3 = max(Merc_c3);
error_c3 = immse(v,ocv_c3);
rms_c3 = sqrt(error_c3);
```

MATLAB code for Model Evaluation Metrics:

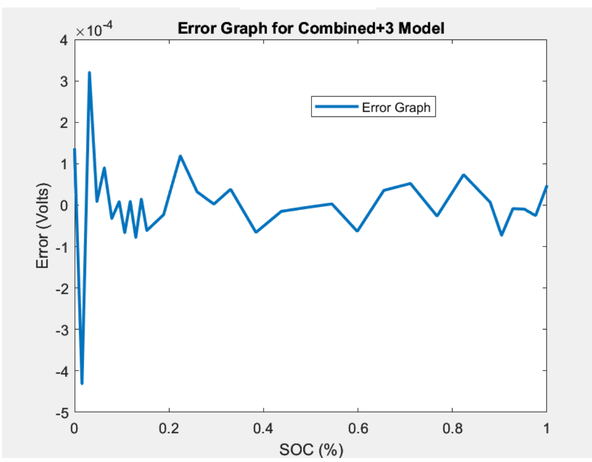
```
% Model Evaluation metrics
z_c3 = sum(Merc_c3.^2);
M_c3 = numel(k);
AIC_c3 = n_c3*log(z_c3/n_c3) + 2*(M_c3 + 1);
```

Metrics Table:

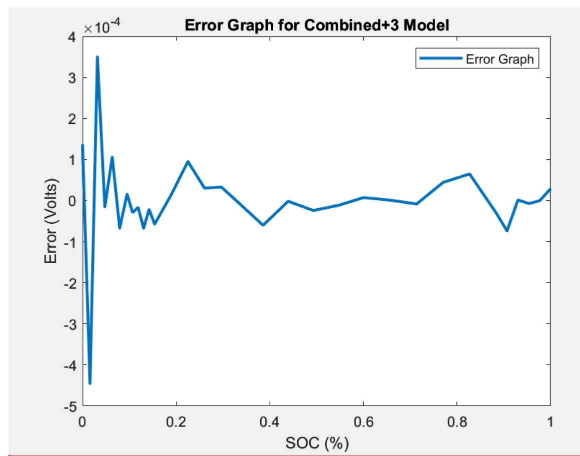
Battery No:	BF	R^2	Max Error	RMSE	AIC
C1202	99.9964	100.000	$3.1998*10^{-4}$	$1.0843*10^{-4}$	-566.2818
C1203	99.9963	100.000	$3.5093*10^{-4}$	$1.1115*10^{-4}$	-564.6973
C1204	99.9966	100.000	$3.8869*10^{-4}$	$1.1099*10^{-4}$	-570.1977
C1205	99.9988	100.000	$7.5180*10^{-5}$	$3.6698*10^{-5}$	-635.6177

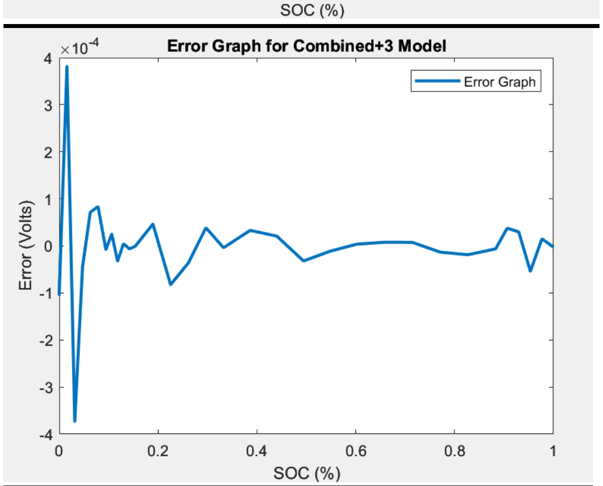
Graph of Error Metrics:

C1202

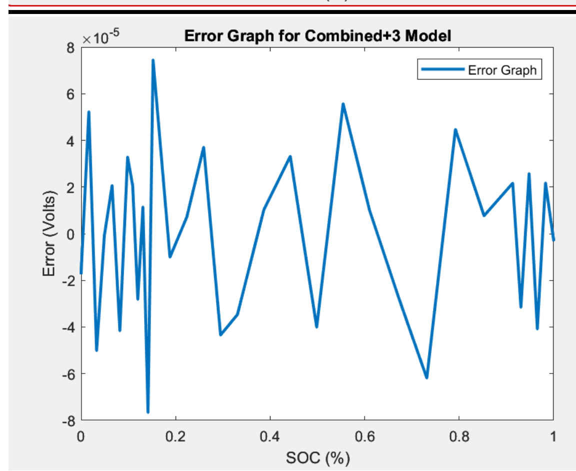


C1203





C1204



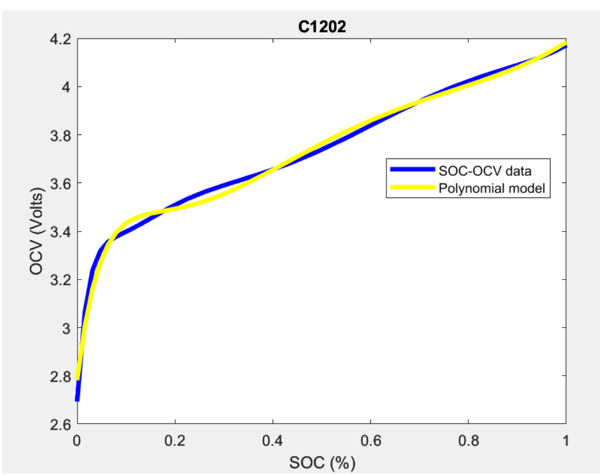
C1205

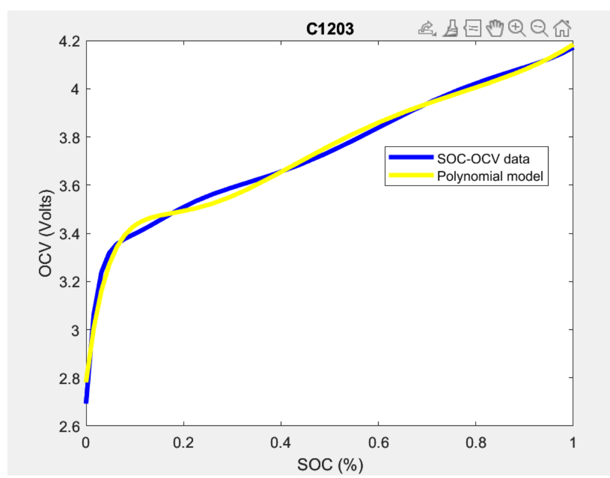
2.6. Polynomial Model:

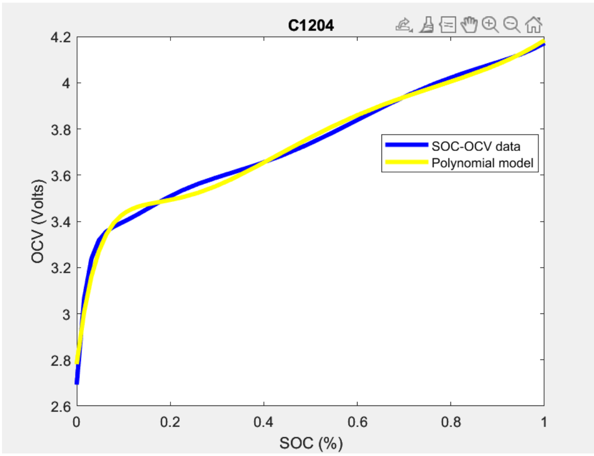
Below is the code to calculate the OCV and plot the graph for the Polynomial model.

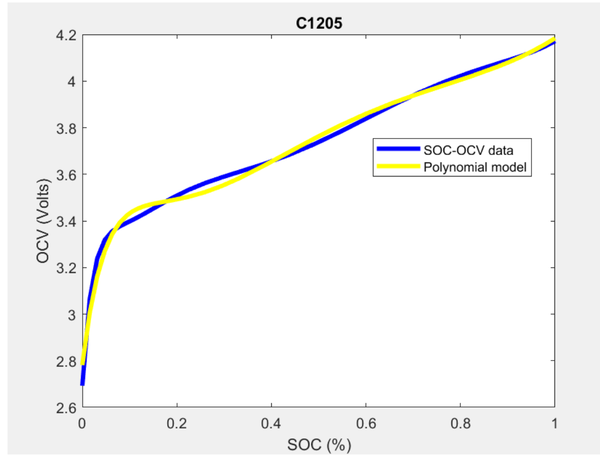
```
%Equation for Polynomial value
p_poly=[ones(length(soc),1) zs zs.^2 zs.^3 zs.^-1 zs.^-2];
u=p_poly';
k=inv(u*p_poly)*(u*v); %value of K for polynomial model
ocv_poly= p_poly*k; %OCV value of polynomial model
plot(soc, v,'b',soc, ocv_poly,'y','linewidth',3 );
xlabel("SOC (%)");
ylabel("OCV (Volts)");
title("C1202")
legend("SOC-OCV data","Polynomial model");
```

Graph of Polynomial model of 4 battery cells.









MATLAB code for Error Metrics:

```
% calculation for Error Metrics for Polynomial Model
n_poly = numel(ocv_poly);
Vbar_poly = (1/n_poly)*(norm(v));
BF_poly = (1-((norm(ocv_poly-v))/(norm(v-Vbar_poly))))*100;
R_squared_poly = (1 - ((norm(ocv_poly-v)).^2/(norm(v-Vbar_poly).^2)))*100;
Merc_poly= v-ocv_poly;
Maxerror_poly = max(Merc_poly);
error_poly = immse(v,ocv_poly);
rms_poly = sqrt(error_poly);
```

MATLAB code for Model Evaluation Metrics:

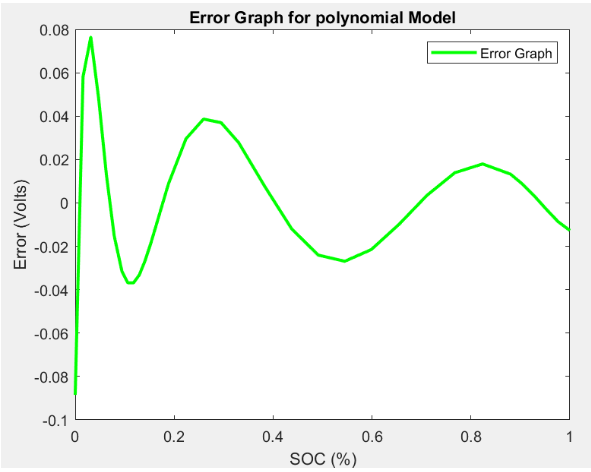
```
% Model Evaluation metrics
z_poly = sum(Merc_poly.^2);
M_poly = numel(k);
AIC_poly = n_poly*log(z_poly/n_poly) + 2*(M_poly + 1);
```

Metrics Table:

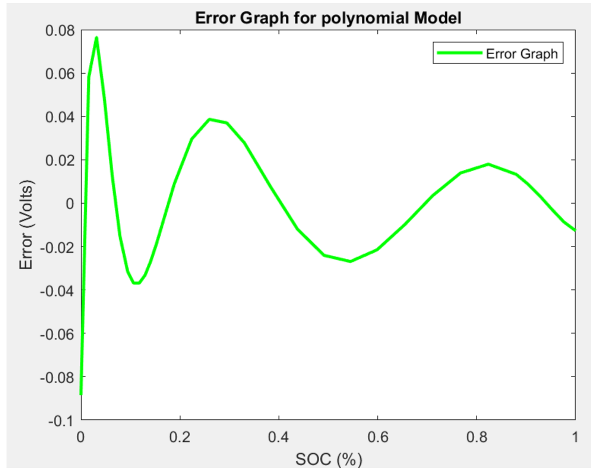
Battery No:	BF	R^2	Max Error	RMSE	AIC
C1202	98.9309	99.9886	0.0764	0.0323	-205.7623
C1203	98.9325	99.9886	0.0763	0.0322	-205.8450
C1204	98.9401	99.9888	0.0750	0.0320	-206.3090
C1205	99.0182	99.9904	0.0699	0.0297	-211.0159

Graph of Error Metrics

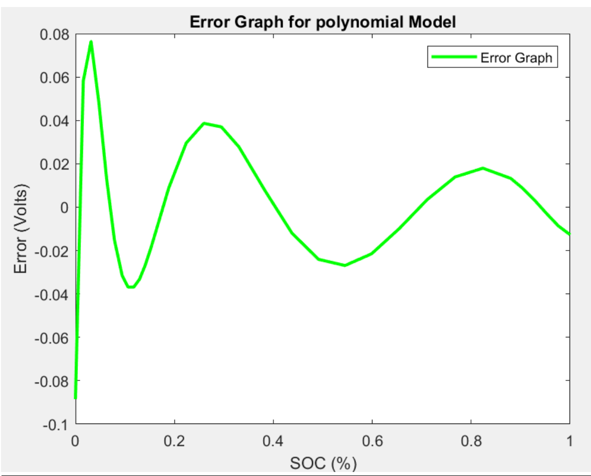
C1202



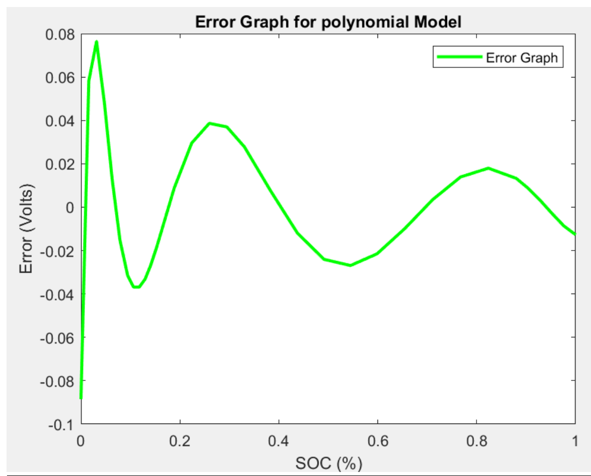
C1203



C1204



C1205

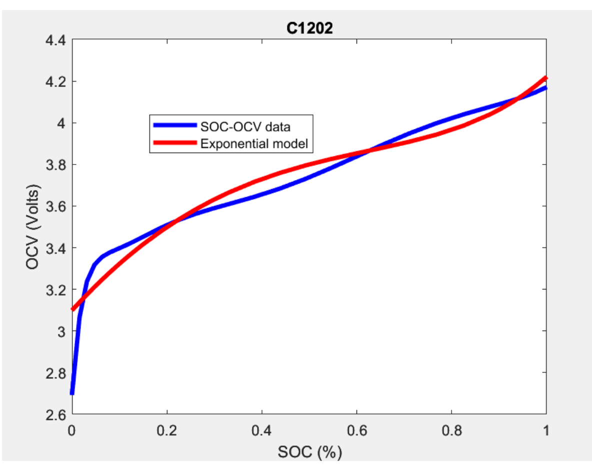


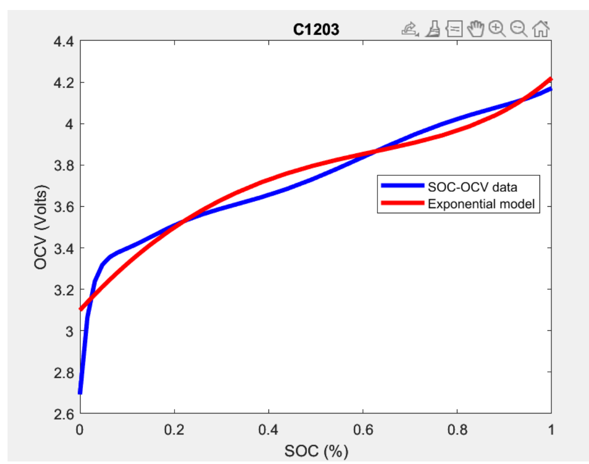
2.7. Exponential Model:

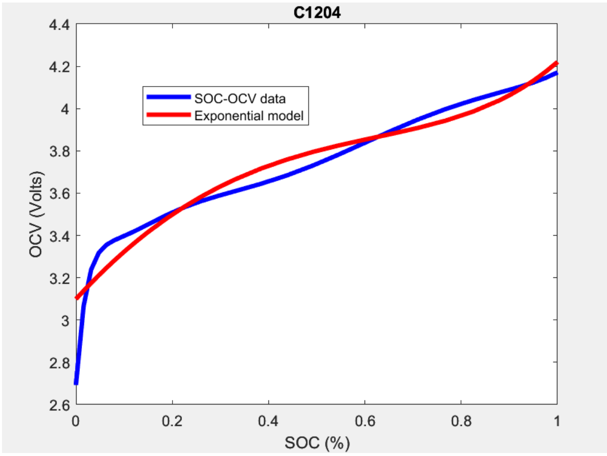
Below is the code to calculate the OCV and plot the graph for the Exponential model.

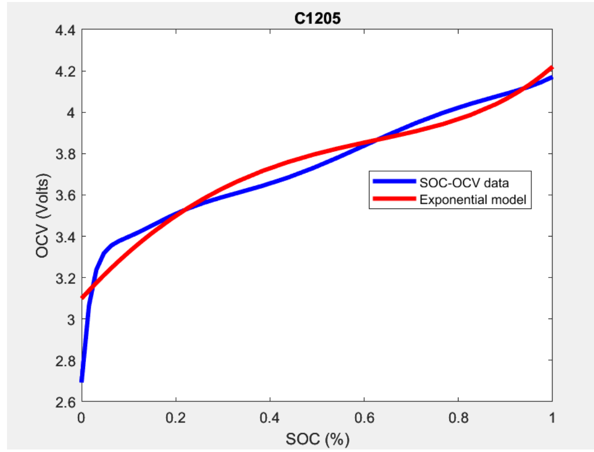
```
%equation for exponential value
p_expo=[ones(length(soc),1) exp(zs) exp(zs.^2) exp(-zs)];
u=p_expo';
k=inv(u*p_expo)*(u*v); %value of K for exponential value
ocv_expo= p_expo*k; %value of OCV for exponential model
plot(soc, v,'b',soc, ocv_expo,'k','linewidth',3 );
xlabel("SOC (%)");
ylabel("OCV (Volts)");
title("C1202")
legend("SOC-OCV data","Exponential model");
```

Graph of Exponential model of 4 battery cells.









MATLAB code for Error Metrics:

```
% calculation for Error Metrics for Exponential Model
n_expo = numel(ocv_expo);
Vbar_expo = (1/n_expo)*(norm(v));
BF_expo = (1-((norm(ocv_expo-v))/(norm(v-Vbar_expo))))*100;
R_squared_expo = (1 - ((norm(ocv_expo-v)).^2/(norm(v-Vbar_expo).^2)))*100;
Merc_expo= v-ocv_expo;
Maxerror_expo = max(Merc_expo);
error_expo = immse(v,ocv_expo);
rms_expo = sqrt(error_expo);
```

MATLAB code for Model Evaluation Metrics:

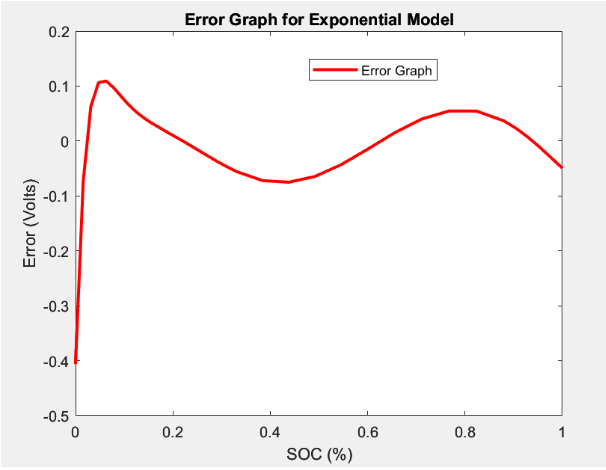
```
% Model Evaluation metrics
z_expo = sum(Merc_expo.^2);
M_expo = numel(k);
AIC_expo = n_expo*log(z_expo/n_expo) + 2*(M_expo + 1);
```

Metrics Table:

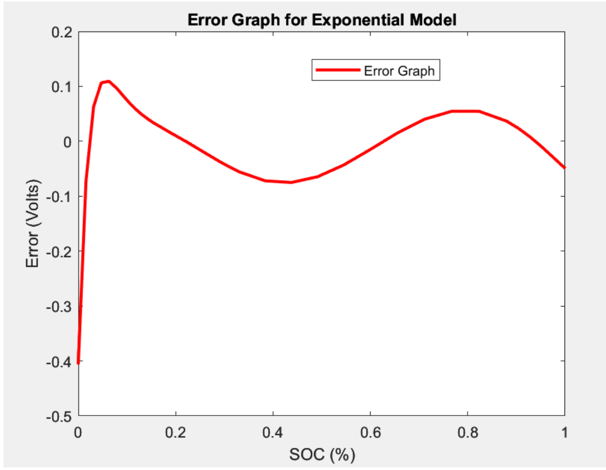
Battery No:	BF	R^2	Max Error	RMSE	AIC
C1202	96.9983	99.9099	0.1093	0.0906	-143.6905
C1203	96.9943	99.9097	0.1098	0.0907	-143.5926
C1204	97.0418	99.9125	0.1082	0.0893	-144.6173
C1205	97.1725	99.9201	0.1020	0.0856	-147.3194

Graph of Error Metrics:

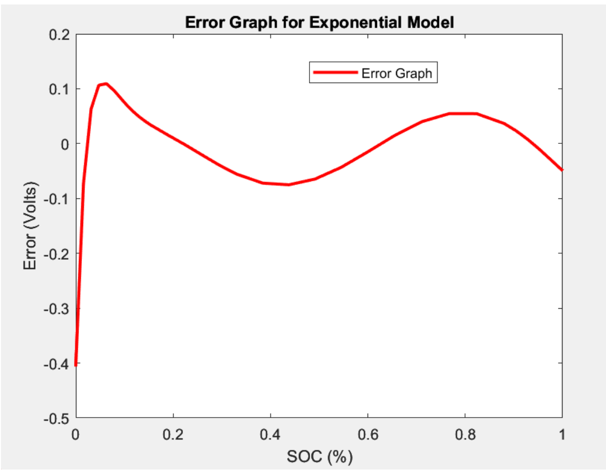
C1202



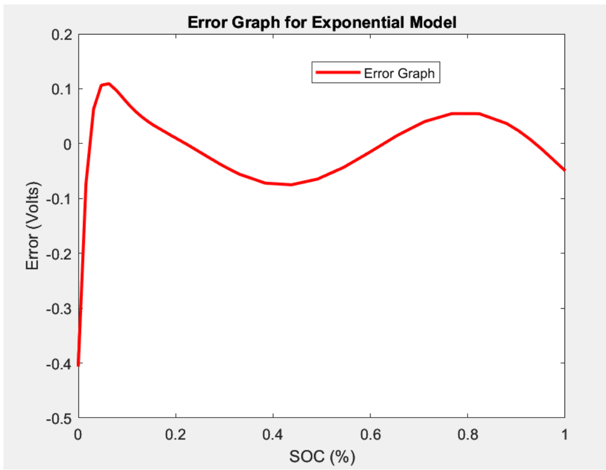
C1203



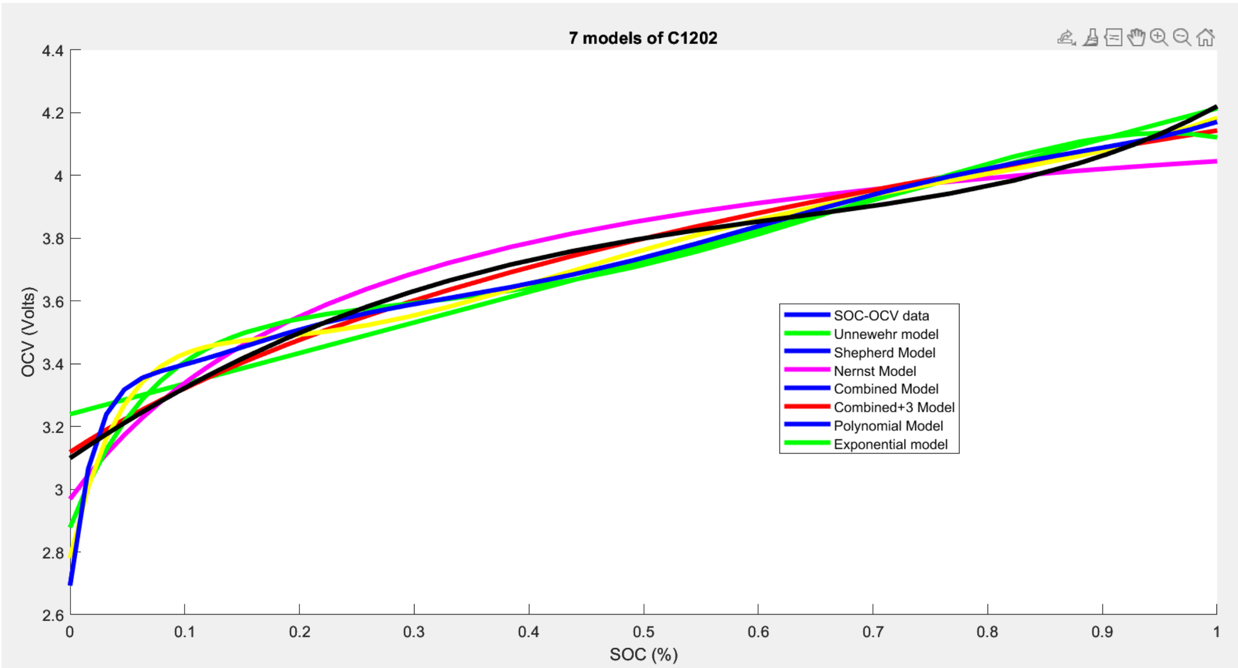
C1204

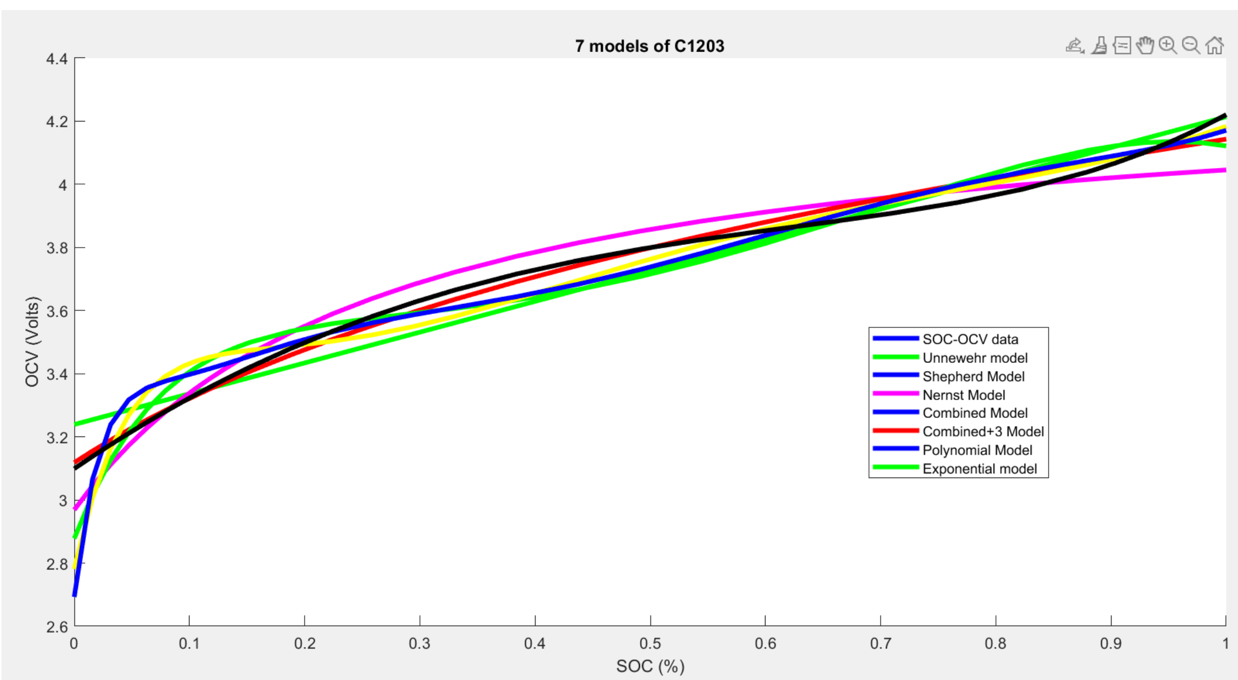


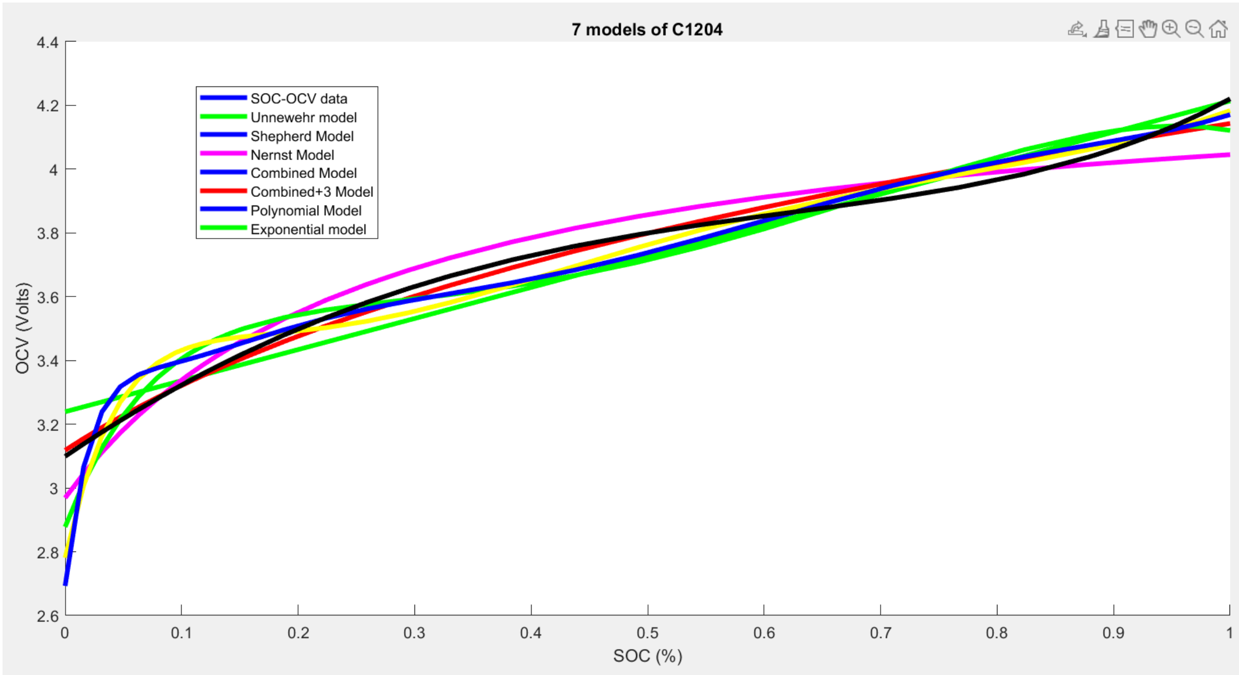
C1205

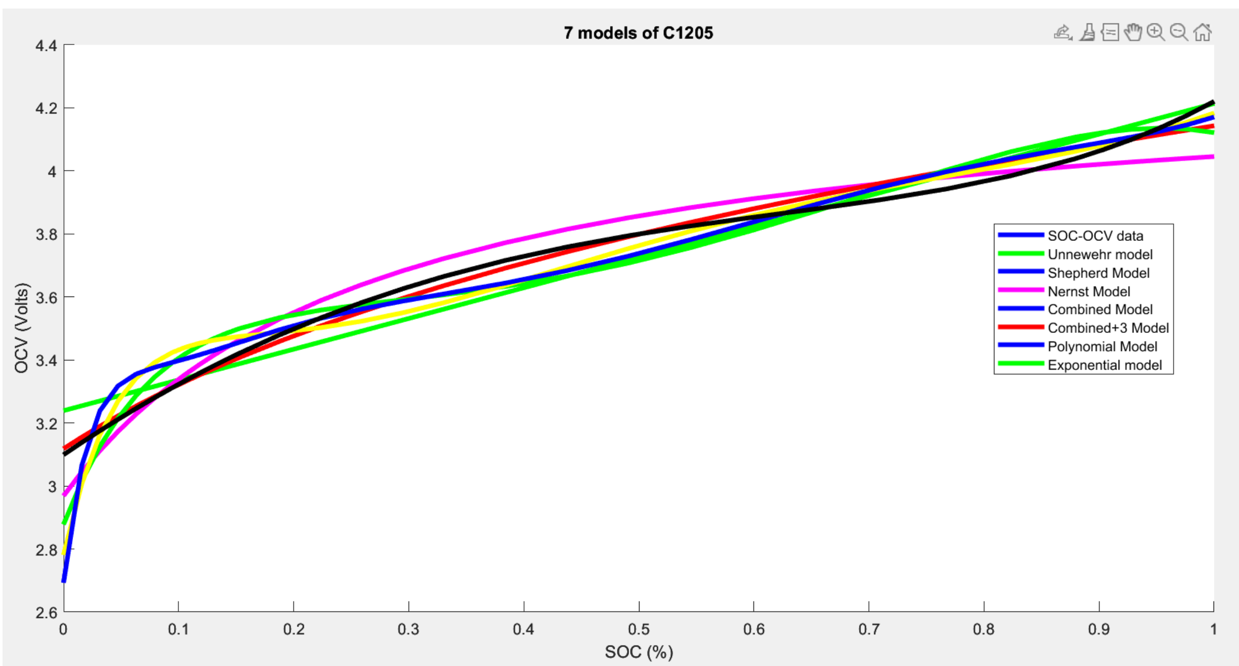


3. Graph of 7 Models for each battery:

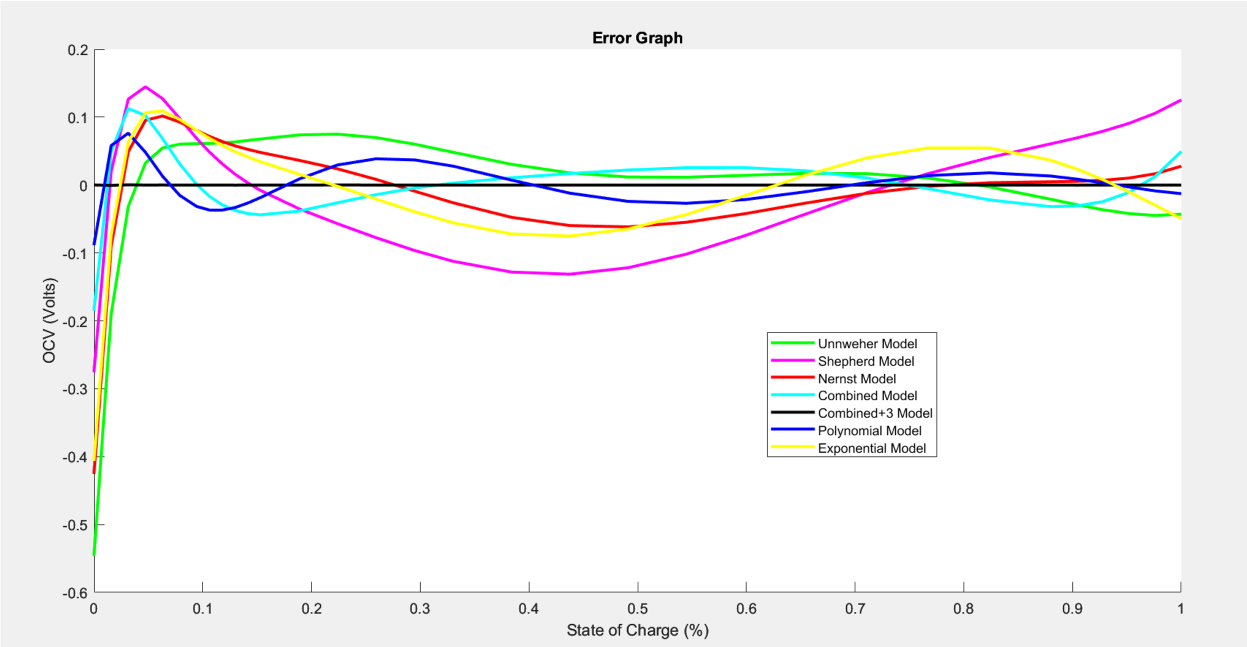


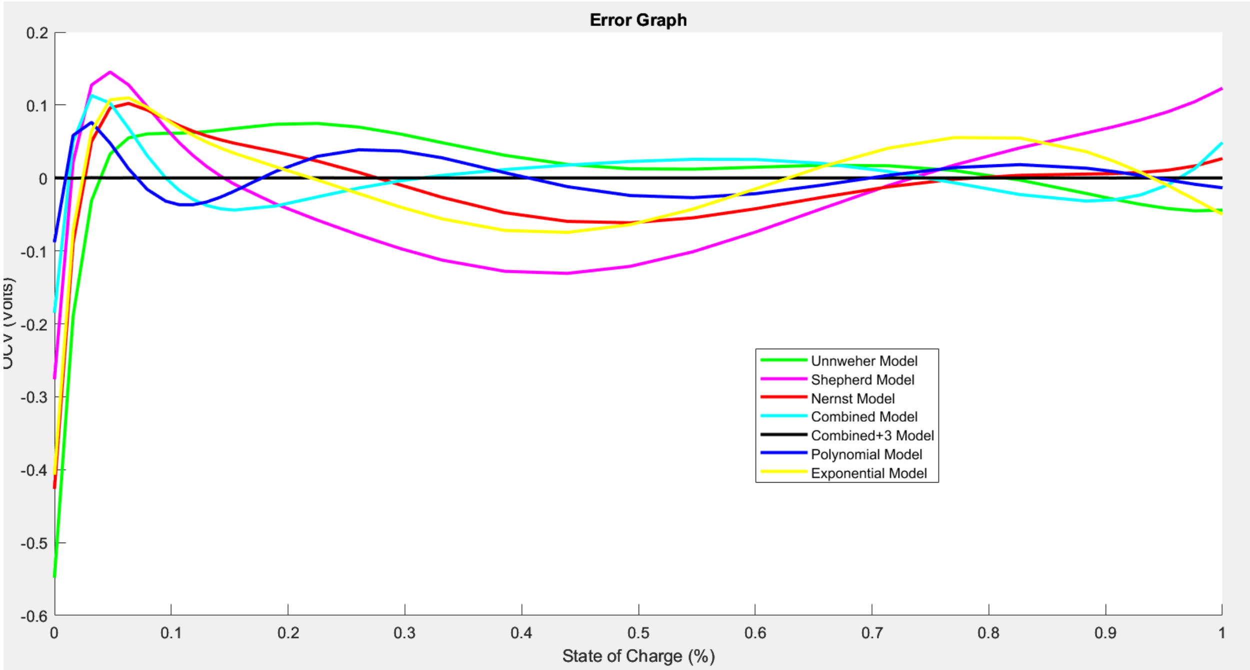


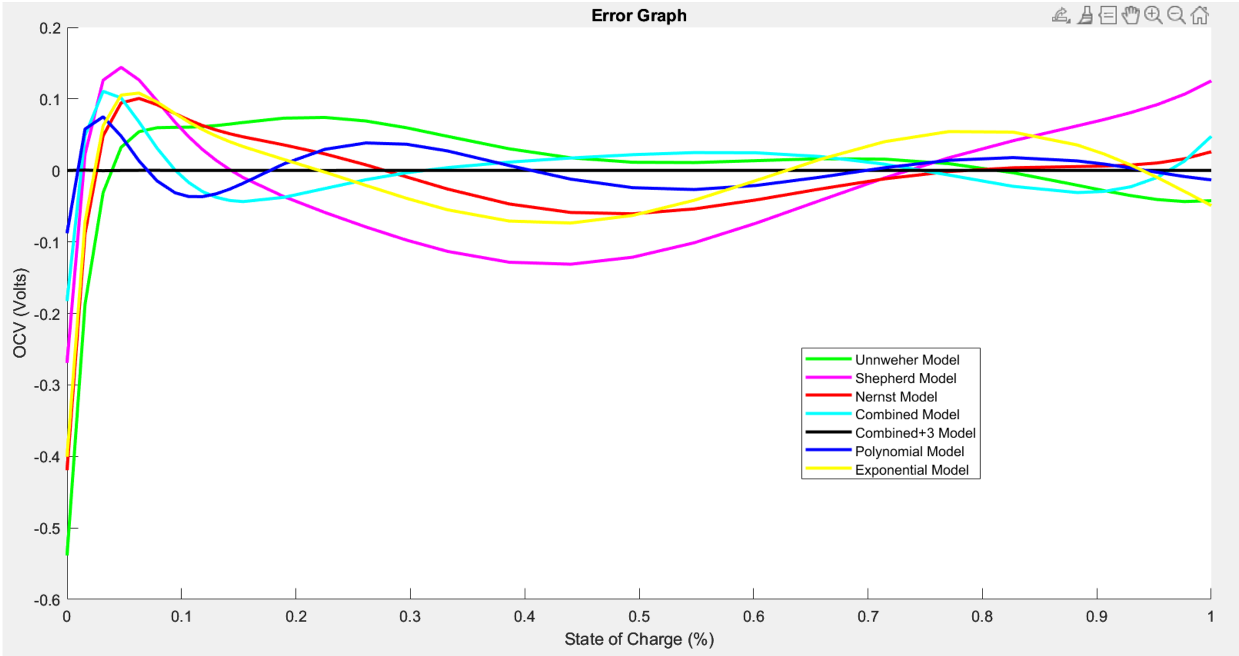

7 models of C1204
SOC-OCV data
Unnewehr model
Shepherd Model
Nernst Model
Combined Model
Combined+3 Model
Polynomial Model
Exponential model
OCV (Volts)
SOC (%)
4.4
4.2
4
3.8
3.6
3.4
3.2
3
2.8
2.6
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

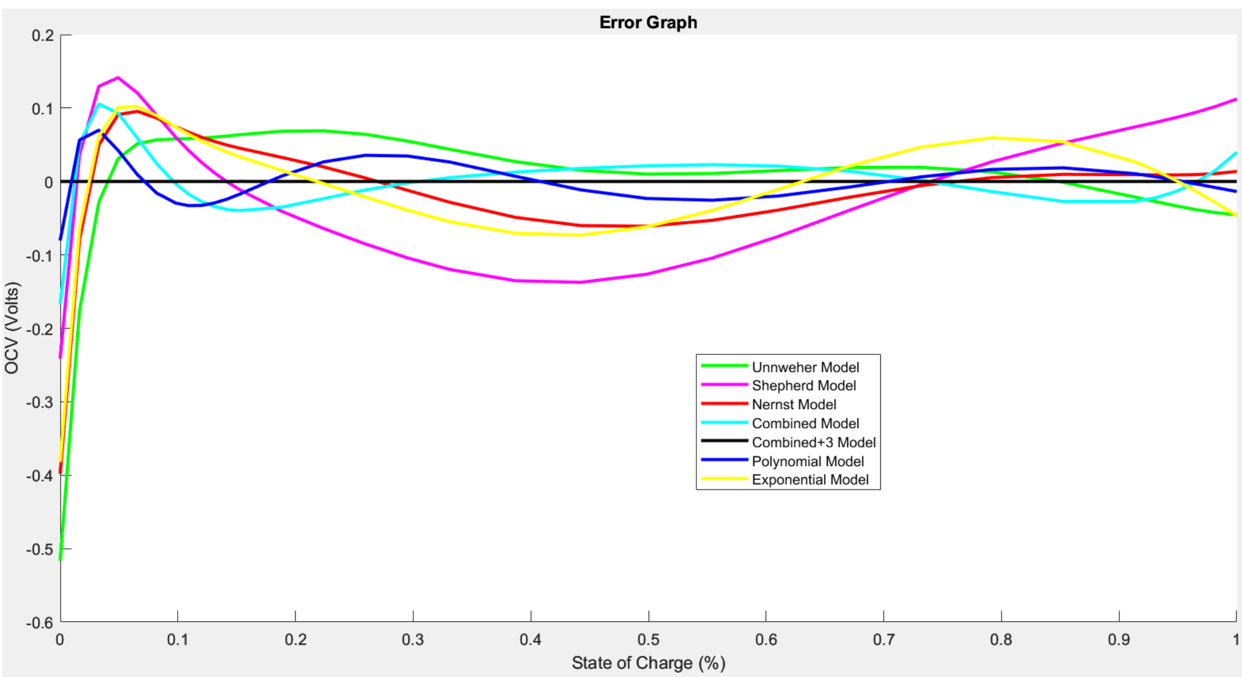

7 models of C1205
SOC-OCV data
Unnewehr model
Shepherd Model
Nernst Model
Combined Model
Combined+3 Model
Polynomial Model
Exponential model
OCV (Volts)
SOC (%)
4.4
4.2
4
3.8
3.6
3.4
3.2
3
2.8
2.6
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

4. Error Graph of 7 models for all batteries:






Error Graph
OCV (Volts)
State of Charge (%)
Unnweher Model
Shepherd Model
Nernst Model
Combined Model
Combined+3 Model
Polynomial Model
Exponential Model


Error Graph
OCV (Volts)
State of Charge (%)
Unnweher Model
Shepherd Model
Nernst Model
Combined Model
Combined+3 Model
Polynomial Model
Exponential Model

5. Ranking Table:

The ranking table is very essential to measure the rank of the above 7 OCV models of the battery. All models are ranked based on error. Thus, the addition of all the rankings associated with each error of the models is known as BORDA ranking.

As per the result, all the batteries have the same ranking table. Additionally, the combined+3 model is the best model according to the data.

The below figure depicts the BORDA ranking table for each of the 4 batteries.

OCV Models	BF	R^2	RMSE	Max Error	AIC	BORDA Ranking
Unnewehr Model	7	7	7	7	7	7
Shepherd Model	6	6	6	6	6	6
Nernst Model	4	4	4	4	4	4
Combined Model	3	3	3	3	3	3
Combined+3 Model	1	1	1	1	1	1
Polynomial Model	2	2	2	2	2	2
Exponential Model	5	5	5	5	5	5

6. Conclusion:

Open Circuit Voltage is very crucial for analyzing the SOC and the electrode's electronic energy. It is very critical to generate correct and accurate OCV modeling. It is proved to be vital for Lithium-ion battery management.

Through modeling the cells, we obtain model ranks. This allows us to find the most optimal model for the cells. The BORDA rank table is here reflecting the 1sr model as the most optimal one and the 7th one as the least optimal. After examination, the Combined+3 model is recommended as the most optimal model.

7. References:

[1] B. B. e. a. Mostafa Shaban Ahmed, "Experimental data on open circuit voltage characterization for Li-ion batteries," *ELSEVIER,* p. 4, 2021.

[2] M. P. a. M. A. Danzer, "Advancements in OCV Measurement and Analysis," *IEEE TRANSACTIONS ON ENERGY CONVERSION,,* vol. 28, no. 3, p. 675 TO 681, 2022.