Advanced Energy Storage Systems

MECH8290-60-R-2024F

Group 41

"Project - 01"

Submitted by

Vijayarangan Pandurengadurai Raju	110128279
Jaiharini Mani	110131837
Bharath Suresh	110128856
Mageshwari Palanisamy	110142170

Submitted to:

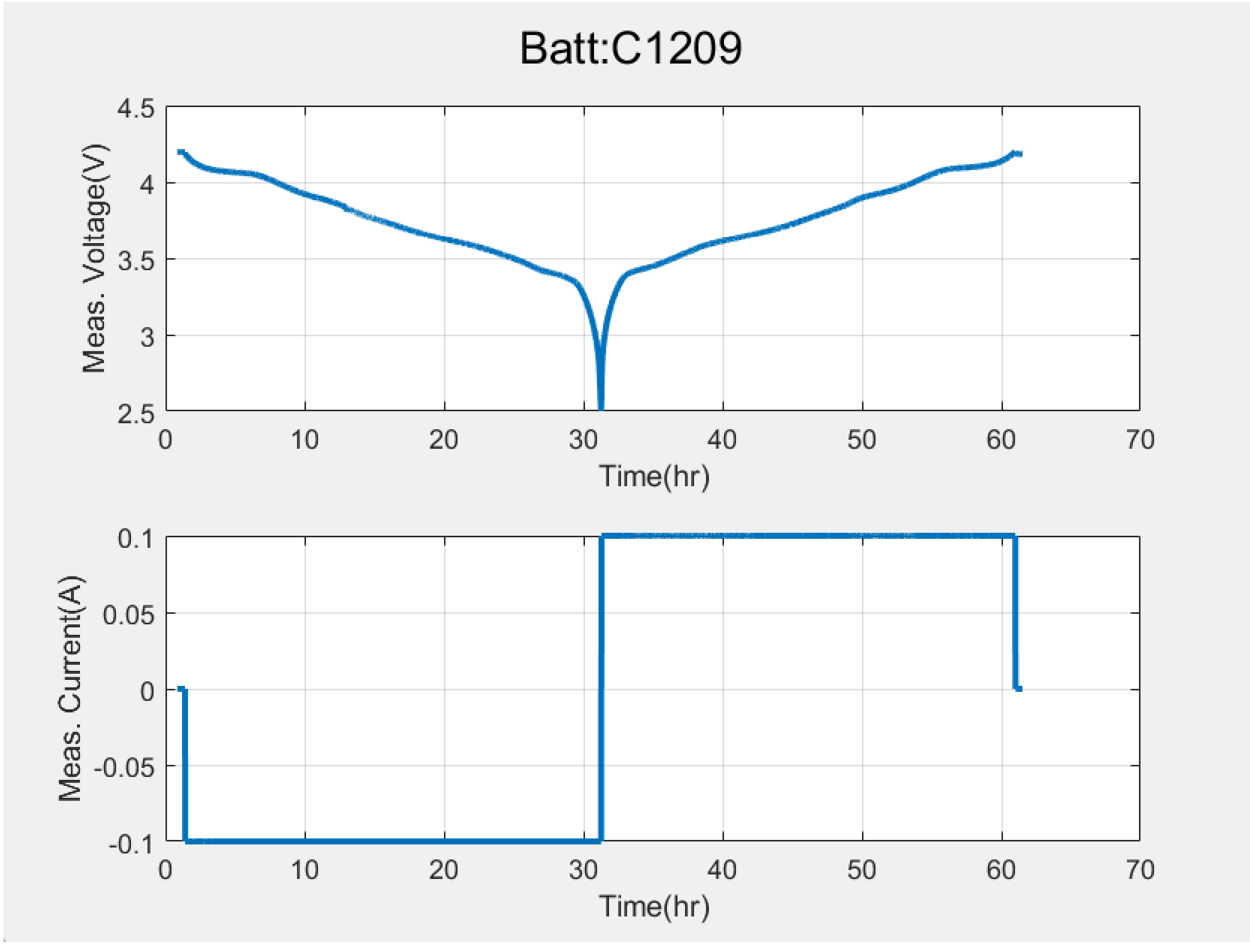
Dr. Balakumar Balasingam

Submitted on:

13th November 2024

Describing the data

```
load('C1209.mat');
h=figure;
subplot(211);plot(Time,V, 'LineWidth',2); grid on
xlabel('Time(hr)');
ylabel('Meas. Voltage(V)');
subplot(212);plot(Time,I, 'LineWidth',2); grid on
xlabel('Time(hr)');
ylabel('Meas. Current(A)');
sgtitle('Batt:C1209');
```



1. Write a computer program to obtain the charge capacity Q_c and discharge capacity Q_d of the battery.

Matlab Code

```
% *********************Question 1*********************
% Identify the indices where the current I is positive (charging) and negative
(discharging)
charging_indices = find(I > 0);
discharging_indices = find(I < 0);

% Calculate the total charging and discharging time
charging_time = Time(charging_indices(end)) - Time(charging_indices(1));
discharging_time = Time(discharging_indices(end)) - Time(discharging_indices(1));

% Compute the average absolute current during charging and discharging
average_charging_current = mean(abs(I(charging_indices(1):charging_indices(end))));
average_discharging_current =
mean(abs(I(discharging_indices(1):discharging_indices(end))));

% Calculate the total charge for charging and discharging phases
Qc = charging_time * average_charging_current;
Qd = discharging_time * average_discharging_current;

% Display results
disp('*********************Question 1*********************');
fprintf('Charge Capacity (Qc): %.4f Ah\n', Qc);
fprintf('Discharge Capacity (Qd): %.4f Ah\n', Qd);
```

Output

```
**********************Question 1**********************
Charge Capacity (Qc): 2.9687 Ah
Discharge Capacity (Qd): 2.9843 Ah
```

2 Questions

Use the provided V-I data to answer the following questions.

1. Write a computer program to obtain the charge capacity Q_c and discharge capacity Q_d of the battery.

2. The SOC at the time instant t_k can be computed as

$$s(t_k) = s(t_{k-1}) + \frac{\Delta_k i(t_k)}{3600Q(t_k)} \tag{18}$$

where $\Delta_k = t_k - t_{k-1}$ and $Q(t_k)$ is the battery capacity that is defined as

$$Q(t_k) = \begin{cases} Q_c & i(t_k) > 0 \\ Q_d & i(t_k) < 0 \end{cases} \tag{19}$$

Assuming that the battery was full at the start, i.e, $s(t_k) = 1$ for the first sample, compute SOC and plot it against time.

Matlab Code

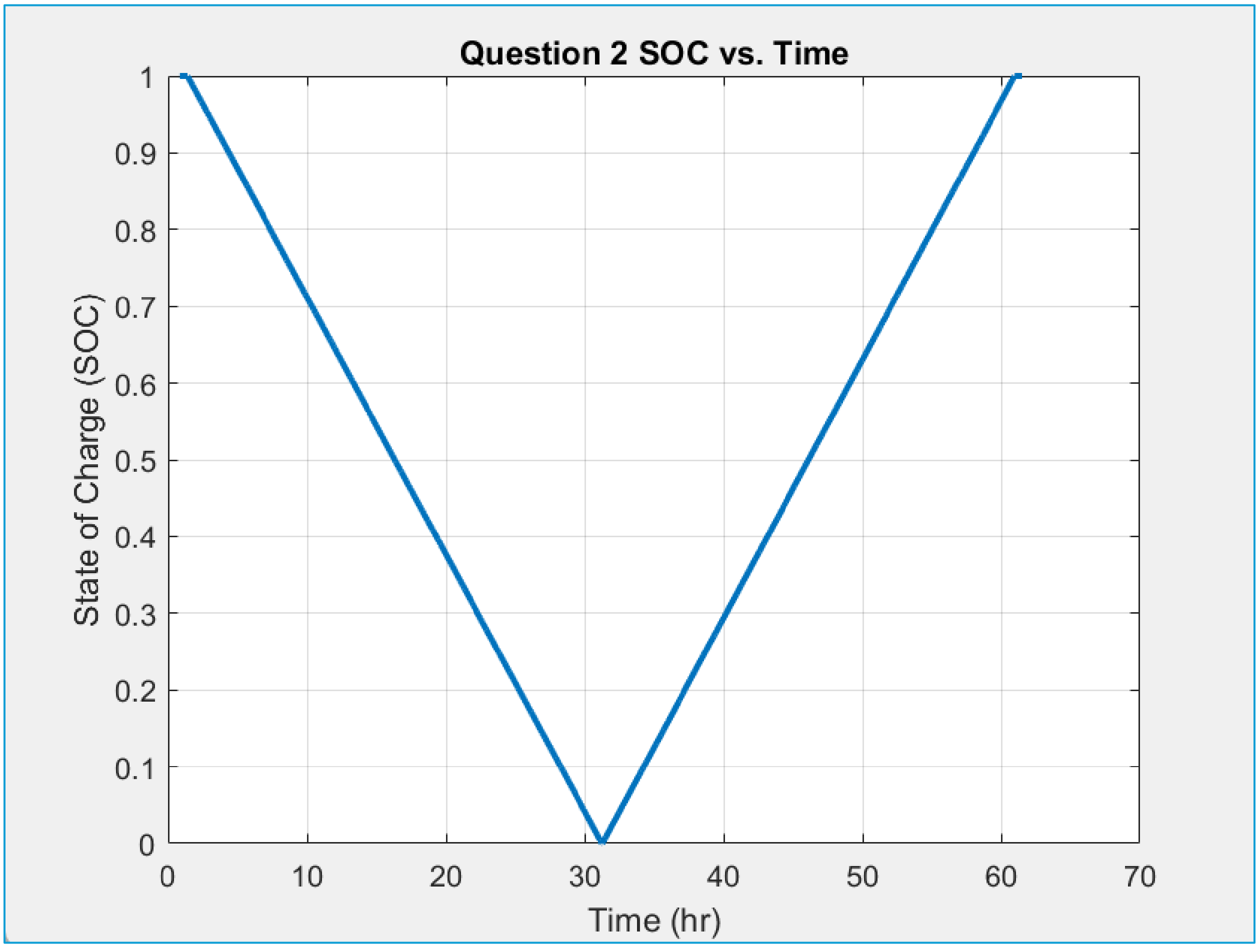
```
% ********************Question 2*********************
SOC = zeros(length(I), 1);
SOC(1) = 1;

for i = 2:length(I)
    delta = Time(i) - Time(i-1);
    if(I(i))<0
        SOC(i) = SOC(i-1) + (delta * I(i))/Qd;
    else
        SOC(i) = SOC(i-1) + (delta * I(i))/Qc;
    end

    if (SOC(i) < 0)
        SOC(i) = 0;
    elseif(SOC(i) > 1)
        SOC(i) = 1;
    end
end

% Plot Question 2 SOC vs. Time
figure;
plot(Time, SOC, 'LineWidth', 2);
xlabel('Time (hr)');
ylabel('State of Charge (SOC)');
title('Question 2 SOC vs. Time');
grid on;
```

Output



Question 2 SOC vs. Time
State of Charge (SOC)
Time (hr)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70

3. Compute the OCV parameters for the two models listed in Section 1.1 (use linear scaling with $\epsilon = 0.175$). Round the parameters for 4 decimal points and display the computed parameters for both models.

Matlab Code

```
% *********************Question 3*********************
% Define epsilon for scaling
epsilon = 0.175;
s_scaled = epsilon + (1 - 2 * epsilon) * SOC; % Apply linear scaling

% Define the matrix P for both models
% Combined model
P_combined = [ones(size(s_scaled)), 1 ./ s_scaled, s_scaled, log(s_scaled), log(1 -
s_scaled), I];

% Combined+3 model
P_combined3 = [ones(size(s_scaled)), 1 ./ s_scaled, 1 ./ (s_scaled.^2), 1 ./
(s_scaled.^3), ...
                    1 ./ (s_scaled.^4), s_scaled, log(s_scaled), log(1 - s_scaled),
I];

% Calculate the least squares solution for both models
% Combined model
c_combined = (P_combined' * P_combined) \ (P_combined' * V);
c_combined = round(c_combined, 4); % Round to 4 decimal places

% Combined+3 model
c_combined3 = (P_combined3' * P_combined3) \ (P_combined3' * V);
c_combined3 = round(c_combined3, 4); % Round to 4 decimal places

disp('*********************Question 3*********************')
% Display the parameters
disp('Combined model parameters:');
disp(c_combined);

disp('Combined+3 model parameters:');
disp(c_combined3);
```

Output

```
**********************Question 3**********************
Combined model parameters:
   -7.9632
   -1.8722
   18.3512
  -10.6537
    1.6283
    0.1951

Combined+3 model parameters:
  -12.6338
  215.6074
  -37.1312
    4.1399
   -0.2017
 -172.6805
  305.2210
   -2.4440
    0.1935
```

4. Compute and display the error metrics listed in Section 1.3. Based on these metrics, select an OCV-SOC model that is most suitable for SOC estimation.

Matlab Code

```
% *********************Question 4**********************
% Number of data points
N = length(V);

% Model parameters count
M_combined = length(c_combined);
M_combined3 = length(c_combined3);

% Predicted voltage for Combined model
V_combined_hat = P_combined * c_combined;

% Calculate Error Metrics for Combined model
V_mean = mean(V);
R2_combined = (1 - norm(V - V_combined_hat)^2 / norm(V - V_mean)^2) * 100;
MaxError_combined = max(abs(V - V_combined_hat));
RMS_combined = norm(V - V_combined_hat) / sqrt(N - M_combined);

% Display Results for Combined model
disp("*********************Question 4**********************")
fprintf('--- Combined Model ---\n');
fprintf('R² Fit: %.2f%%\n', R2_combined);
fprintf('Max Error: %.4f V\n', MaxError_combined);
fprintf('Root-Mean Square Error (RMS): %.4f V\n', RMS_combined);

% Predicted voltage for Combined+3 model
V_combined3_hat = P_combined3 * c_combined3;

% Calculate Error Metrics for Combined+3 model
R2_combined3 = (1 - norm(V - V_combined3_hat)^2 / norm(V - V_mean)^2) * 100;
MaxError_combined3 = max(abs(V - V_combined3_hat));
RMS_combined3 = norm(V - V_combined3_hat) / sqrt(N - M_combined3);

% Display Results for Combined+3 model
fprintf('\n--- Combined+3 Model ---\n');
fprintf('R² Fit: %.2f%%\n', R2_combined3);
fprintf('Max Error: %.4f V\n', MaxError_combined3);
fprintf('Root-Mean Square Error (RMS): %.4f V\n', RMS_combined3);
```

Output

```
**********************Question 4**********************
--- Combined Model ---
R² Fit: 97.12%
Max Error: 0.2917 V
Root-Mean Square Error (RMS): 0.0534 V

--- Combined+3 Model ---
R² Fit: 99.68%
Max Error: 0.1097 V
Root-Mean Square Error (RMS): 0.0178 V
```

5. Plot the selected OCV-SOC model

(hint: create s = 0:0.01:1 and compute $V_\circ(s)$; then plot (s, $V_\circ(s)$))

Matlab Code

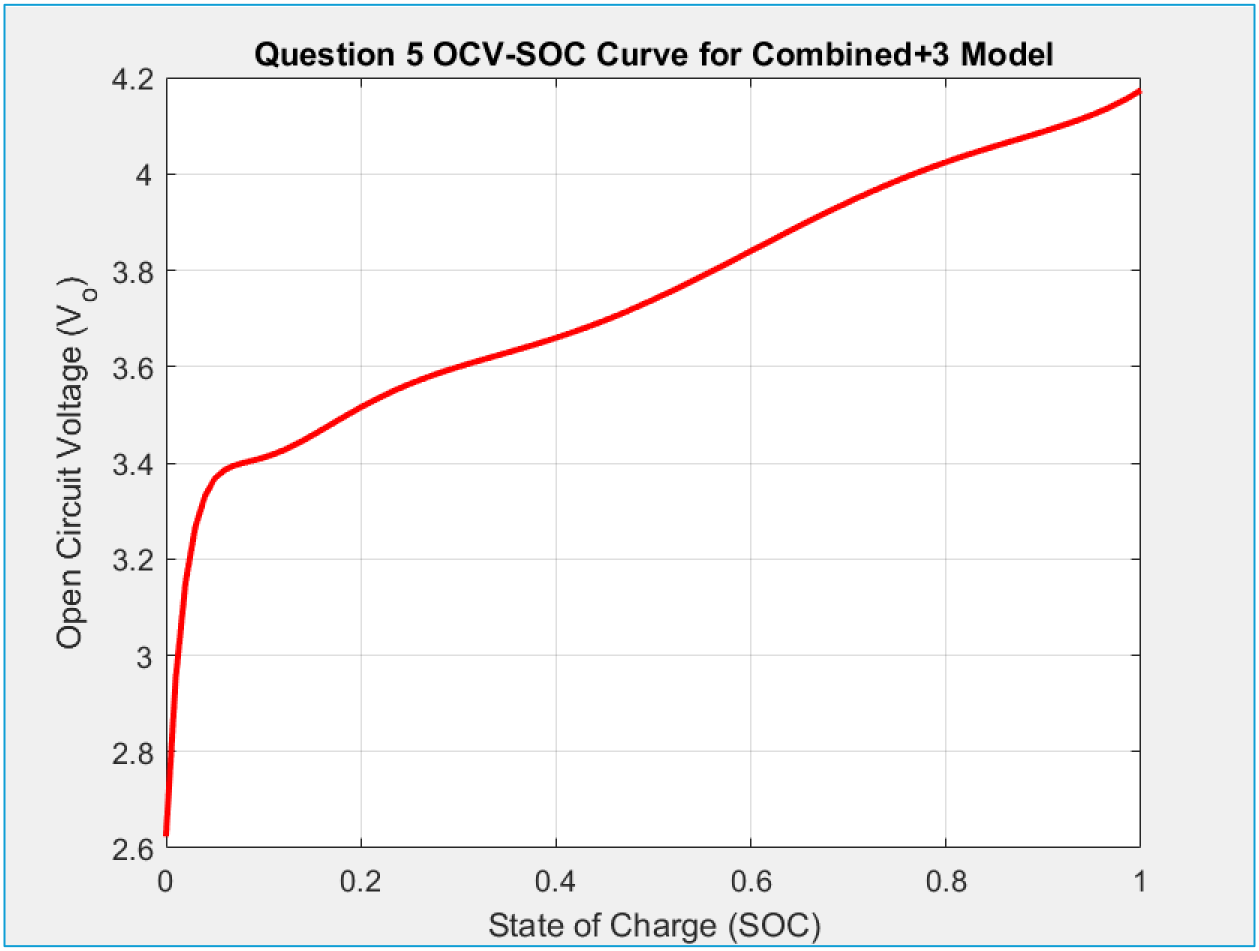
```
% *********************Question 5*********************
% Define SOC range from 0 to 1 with a step of 0.01
s_raw = (0:0.01:1);
s = epsilon + (1 - 2 * epsilon) * s_raw;
k = c_combined3;

% Combined+3 Model Calculation using the array values
V_combined3 = k(1) + k(2)./s + k(3)./s.^2 + k(4)./s.^3 + k(5)./s.^4 + ...
       k(6) * s + k(7) * log(s) + k(8) * log(1 - s);

% Plot the Combined+3 Model
figure; hold on; box on;
plot(s_raw, V_combined3, 'r-', 'LineWidth', 2);


% Add labels and title
xlabel('State of Charge (SOC)');
ylabel('Open Circuit Voltage (V_{o})');
title('Question 5 OCV-SOC Curve for Combined+3 Model');
grid on;
```

Output


Question 5 OCV-SOC Curve for Combined+3 Model
Open Circuit Voltage (V_o)
State of Charge (SOC)
4.2
4
3.8
3.6
3.4
3.2
3
2.8
2.6
0
0.2
0.4
0.6
0.8
1

6. Using the OCV-SOC curve in question 5 above, answer the following.

(a) What is the OCV when the battery SOC is 25%, 50% and 75%.

Matlab Code

```
fprintf('\n*********Question 6a*********\n')
SOC_levels = [0.25, 0.50, 0.75];
OCV_values = interp1(s_raw, V_combined3, SOC_levels, 'linear');
fprintf('OCV when the battery SOC is 25%%: %.4f V\n', OCV_values(1));
fprintf('OCV when the battery SOC is 50%%: %.4f V\n', OCV_values(2));
fprintf('OCV when the battery SOC is 75%%: %.4f V\n', OCV_values(3));
```

Output

```
*********Question 6a*********
OCV when the battery SOC is 25%: 3.5638 V
OCV when the battery SOC is 50%: 3.7389 V
OCV when the battery SOC is 75%: 3.9861 V
```

(b) What is the SOC, when the battery measured 4V? (assume zero current)

Matlab Code

```
fprintf('*********Question 6b*********\n')
% Target OCV value
target_OCV = 4.0;
% Interpolating SOC at 4V
SOC_at_4V = interp1(V_combined3, s_raw, target_OCV, 'linear');

fprintf('SOC at 4V (zero current): %.4f\n', SOC_at_4V);
```

Output

```
*********Question 6b*********
SOC at 4V (zero current): 0.7672
```

(c) What is the SOC, when the battery measured 3.8V across the terminals, while being discharged using a 4A load current?

Matlab Code

```
fprintf('*********Question 6c*********\n')
% Target OCV value with a discharge current of 1A
Vo_OCV_3_8V = 3.8;

Reff = c_combined3(end);
discharge_current6c = -1;
target_OCV_3_8V = Vo_OCV_3_8V - discharge_current6c * Reff;
```

```
% Interpolating SOC at 3.8V
SOC_at_3_8V = interp1(V_combined3, s_raw, target_OCV_3_8V, 'linear');

fprintf('SOC at %.4fV with 1A discharge: %.4f\n', target_OCV_3_8V, SOC_at_3_8V);
```

Output

```
*********Question 6c*********
SOC at 3.9935V with 1A discharge: 0.7591
```

7. Answer the questions in Question 6 using the OCV parameters that you selected in Question 4. Explain the differences if any.

Matlab Code

```
% *********************Question 7*********************
% from question 4
fprintf('\n*********Question 7a*********\n')

% Check for duplicates in the SOC array
[SOC_unique, idx_unique] = unique(SOC, 'stable'); % Get unique SOC values and their
indices
V_combined3_hat_unique = V_combined3_hat(idx_unique); % Corresponding OCV values
without duplicates

OCV_values_7a = interp1(SOC_unique, V_combined3_hat_unique, SOC_levels, 'linear');
fprintf('OCV when the battery SOC is 25%%: %.4f V\n', OCV_values_7a(1));
fprintf('OCV when the battery SOC is 50%%: %.4f V\n', OCV_values_7a(2));
fprintf('OCV when the battery SOC is 75%%: %.4f V\n', OCV_values_7a(3));

fprintf('*********Question 7b*********\n')
% Target OCV value
target_OCV = 4.0;
% Interpolating SOC at 4V
SOC_at_4V = interp1(V_combined3_hat_unique, SOC_unique, target_OCV, 'linear');

fprintf('SOC at 4V (zero current): %.4f\n', SOC_at_4V);

fprintf('*********Question 7c*********\n')
% Target OCV value with a discharge current of 1A
Vo_OCV_3_8V = 3.8;

Reff = c_combined3(end);
discharge_current6c = -1;
target_OCV_3_8V = Vo_OCV_3_8V - discharge_current6c * Reff;
% Interpolating SOC at 3.8V
SOC_at_3_8V = interp1(V_combined3_hat_unique, SOC_unique, target_OCV_3_8V, 'linear');

fprintf('SOC at %.4fV with 1A discharge: %.4f\n', target_OCV_3_8V, SOC_at_3_8V);
```

Output

```
*********Question 7a*********
OCV when the battery SOC is 25%: 3.5587 V
OCV when the battery SOC is 50%: 3.7456 V
OCV when the battery SOC is 75%: 3.9998 V
*********Question 7b*********
SOC at 4V (zero current): 0.7852
*********Question 7c*********
SOC at 3.9935V with 1A discharge: 0.7839
```

Differences from Question 4 and Question 5 (using range 0:0.01:1)

Differences:

	Question 7a		
SOC	OCV (V) from Question 5	OCV (V) from Question 4	Differences
25%	3.5638	3.5587	0.0051
50%	3.7389	3.7456	0.0067
75%	3.9861	3.9998	0.0137
	Question 7b		
V	SOC from Question 5	SOC from Question 4	Differences
4V	0.7672	0.7852	0.018
	Question 7c		
V	SOC from question 5	Soc from Question 4	Differences
3.9935	0.7591	0.7839	0.0248