Programmable CCCV EV Battery Cycler Prototype.

ME 295B Final presentation
May 23, 2024
by
Beniam Ayele
B.S.M.E, San Jose State University , 2024
Building Automation System Technician III, EMCOR Service Mesa Energy

Presentation Topics

- Introduction
  - Objective
  - Motivation
  - Spesification

- Literature Review
  - LiB degradation mechanisms
  - LiB modeling

- MATLAB and Simulink Model
  - Simulink Model and Simulation Result
    - For 3A ,15A and 30A Charging and Discharging

Presentation Topics

- ❖ Electronic Circuit Design, Modeling and Testing
  - ➢ Basic working principle of CCCV Circuit
  - ➢ EasyEDA Circuit Design for Charging
    - ■ Circuit Diagram
    - ■ PCB Design
    - ■ PCB 3D Model
  - ➢ Working Prototype of CCCV Circuit with Discharge
    - ■ Hardware
      - ● Actual Circuit
      - ● Sensor & Components used in the Circuit

Presentation Topics

- Software
    - Arduino IDE code
    - Python Code for Data preparation
- ➢ Charging and Discharging Test results
    - 1 Amp
    - 2 Amp
    - 3 Amp
- ➢ Challenges, Future Improvements and Conclusion.

Introduction

Objective:-

- ❖ Design and build a programmable Lithium battery (LiB) cycler for studying battery State of Charge (SoC), State of Health (SoH) and Remaining Useful Life (RUL).
  - ➢ User specified current setpoint.
  - ➢ End Product will include both hardware and software part.
  - ➢ This project will mainly focus on the hardware part and include some basic function of software
  - ➢ Outcome of this project is to build and test functional battery cycler prototype

Introduction

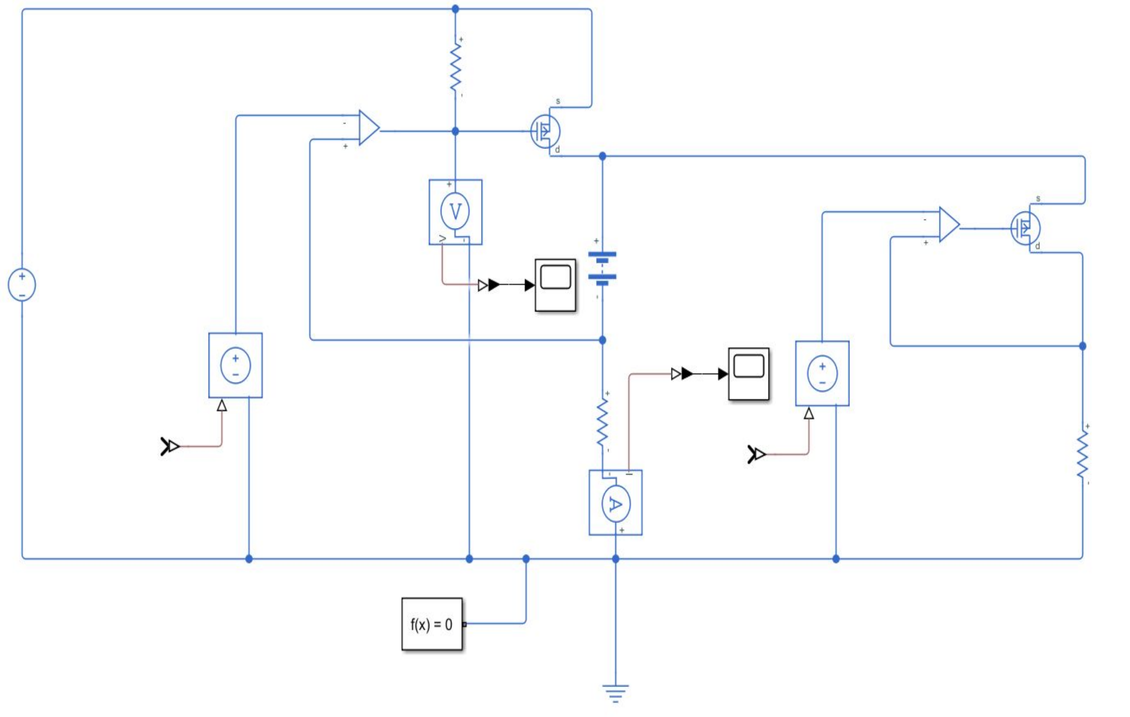
Motivation

- Cost
  - Current product available in the market is very expensive
  - Proprietary software
    - Difficult to making changes
    - Modify software
- Fight against Global warming
  - The transition from gas powered to electric vehicles
    - LiB is primary choice for energy storage
    - Not very effective and its susceptible degradation

Introduction

- Continuous study for next gen batteries
  - Final product will increase the number of universities and individual participate in this study
  - Train new generation of automotive technicians
- Finally I believe transition to EV is not enough we need to use energy efficiently and effectively.

Introduction



- Control current flow through battery and load resistor
  - P-MOSFET
  - Op-Amp
- Prototype target max current
  - 3 Amps
- Final product target Max current target
  - Programmable up to Max 30 Amps
  - Programing include SoC, SoH, and RUL.

Introduction

What make our product unique?

- Cost
- Circuit design simplicity
  - Compact design
- No expensive software subscription needed
  - Only need MATLAB
- Modifiable and easy to make change
- Programmable
  - User spacide charging and discharging current setpoint
  - User can define battery parameters
    - Amp hour, min and max battery voltage and SoC
    - Work with any rechargeable battery model

Introduction

Specification

	Project Specification	Product Specification
Max Charging and Discharging current Sp	3 Amps	30 Amps
Programing	❖ Battery Voltage & Current ❖ User defined min and max charging and discharging current setpoint and battery voltage ❖ Battery temperature and software safety	❖ Battery Voltage & Current ❖ User defined min and max charging and discharging current setpoint and battery voltage ❖ Battery parameters like Amp hours
SoC, SoH, and RUL	No	Yes, use equivalent LiB circuit model
Power supply	5 Volts	5 Volts

Degradation Mechanisms

- Primary degradation mechanisms
  - Solid Electrolyte Interface (SEI) layer growth
    - Unstable electrolyte :- Loss of Lithium Inventory (LLI)
  - Lithium platin
    - Fast charging, low temperature, and overcharging :- LLI
  - Particle fracture
    - Expanding and contracting of the cathode during charging :- LLI and loss of Active Material (LAM)

[9],[11],[12]

LiB Modeling

- LiB modeling for SoC and SoH estimation
  - Data-driven model
  - Electrochemical model
  - Equivalent circuit model

[9],[11],[12]

Chemical degradation model

- Model utilizes
  - Energy balance
  - Mass,and heat transfer principles
- Based on the side chemical reactions that occur
  - Investigate Capacity lose
  - Evaluate resistance increase
    - Solid electrolyte interface (SIE)
- Model suitable for storage condition
  - Irreversible capacity loss to diffusion-induced stresses (DISs)
  - SEI growth and stress factors such as DoD and SoC
- Model fails to account for
  - Effect of previous crack growth on physical stress
    - constant degradation rate

[9],[10],[11],[16],[20], [21],[23]

Data-driven Model

- ❖ SoC and SoH prediction
  - ➢ Data from experimentation
  - ➢ Data collected from EV
- ❖ Model suitable for operating condition
- ❖ Model fails to account for
  - ➢ Storage Conditions

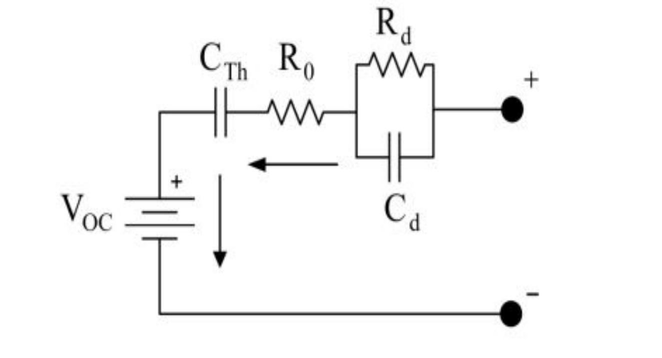
[17],[18],[19]

Equivalent Circuit Model

- Model utilizes
  - Circuit elements like resistors
  - Capacitors
  - Constant voltage sources
  - Charging and discharging current control
- Model represent dynamic behavior LiB
- Commonly used LiB model
  - Computational simplicity
  - LiB representation

[22] and [23]

Equivalent Circuit Model



$$V_L = V_{oc} - I_L R_0 - U_{Th} - U_d$$

$$\frac{d}{dt}U_d + \frac{U_d}{R_d C_d} = \frac{I_L}{C_d}$$

$$\frac{d}{dt}U_{Th} = I_L \frac{dV_{OC}}{dQ_{Th}}$$

$$V_L = V_{oc} - I_L R_0 - \frac{1}{C_{Th}}\int I_L dt - \frac{e^{-\frac{t}{R_d C_d}}}{C_d}\int e^{\frac{t}{R_d C_d}} I_L dt$$

$$SOC = SOC(t_0) - \frac{100}{Q_{rated}}\int_{t_0}^{t_f} I_L \cdot dt$$

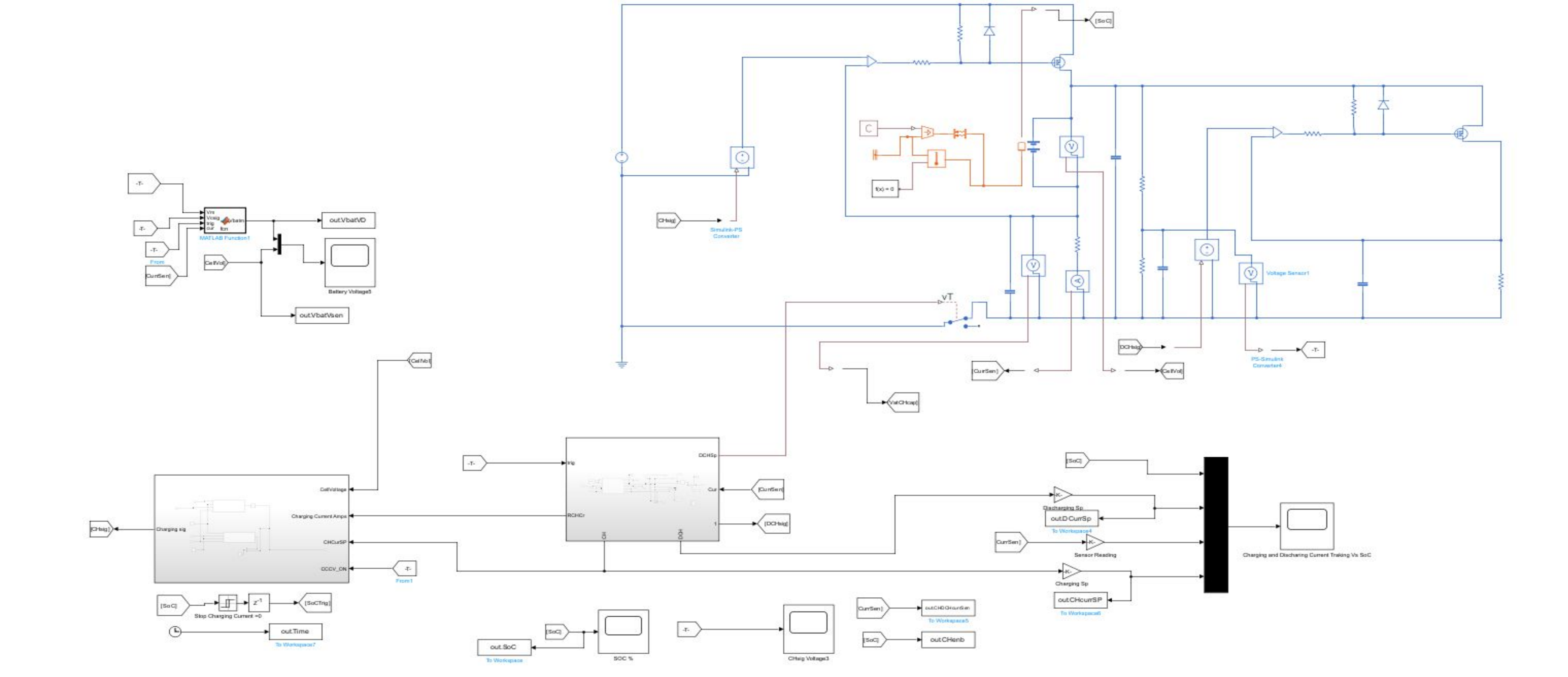
- R0 all electronic resistance
- Rd polarized resistance
- Cd accounts for ione diffusion
- Cth equivalent capacitance
- Voc open-circuit voltage

[31]

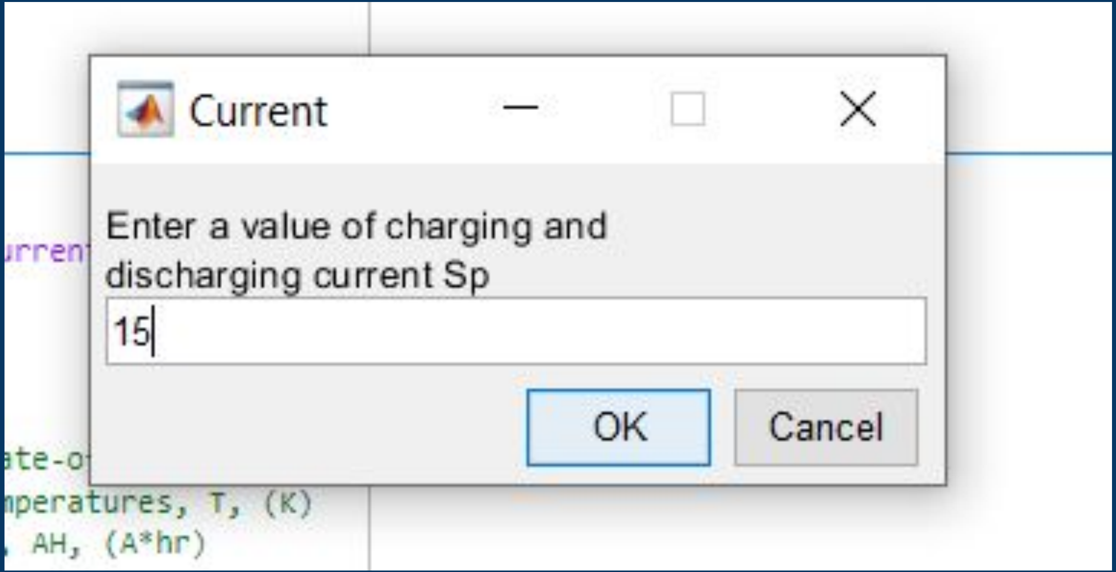
MATLAB and Simulink Model

- Changes made to Simulink example
  - Used the CCCV example from MATLAB as a base
    - Change model to fit project (Op-Amp and P-MOSFET)
    - Integrate user defined charging and discharging current Set-point
    - Incorporate discharging circuit

Current Simulink Model



MATLAB

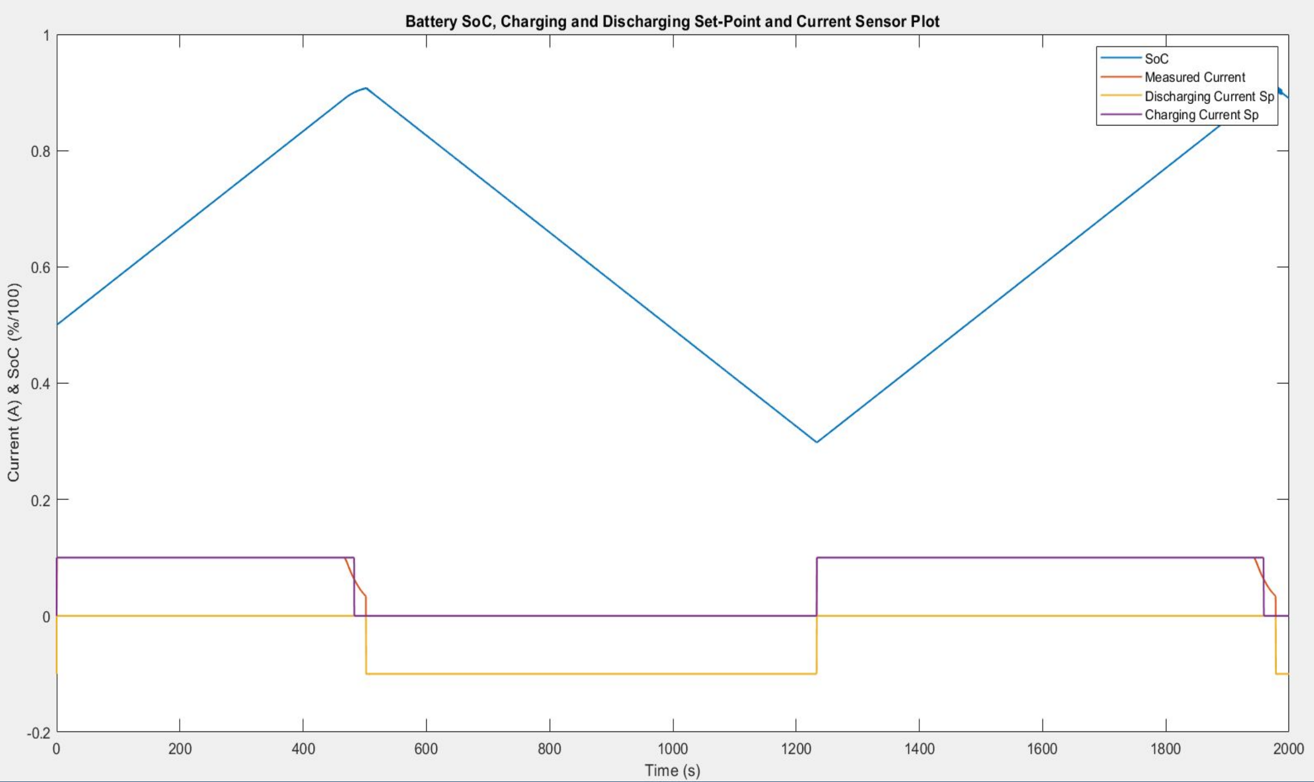


- User defined Charging and Discharging Current set-point.
- Future
  - Battery type and parameters
  - Max and Min battery voltage setpoint

Simulation

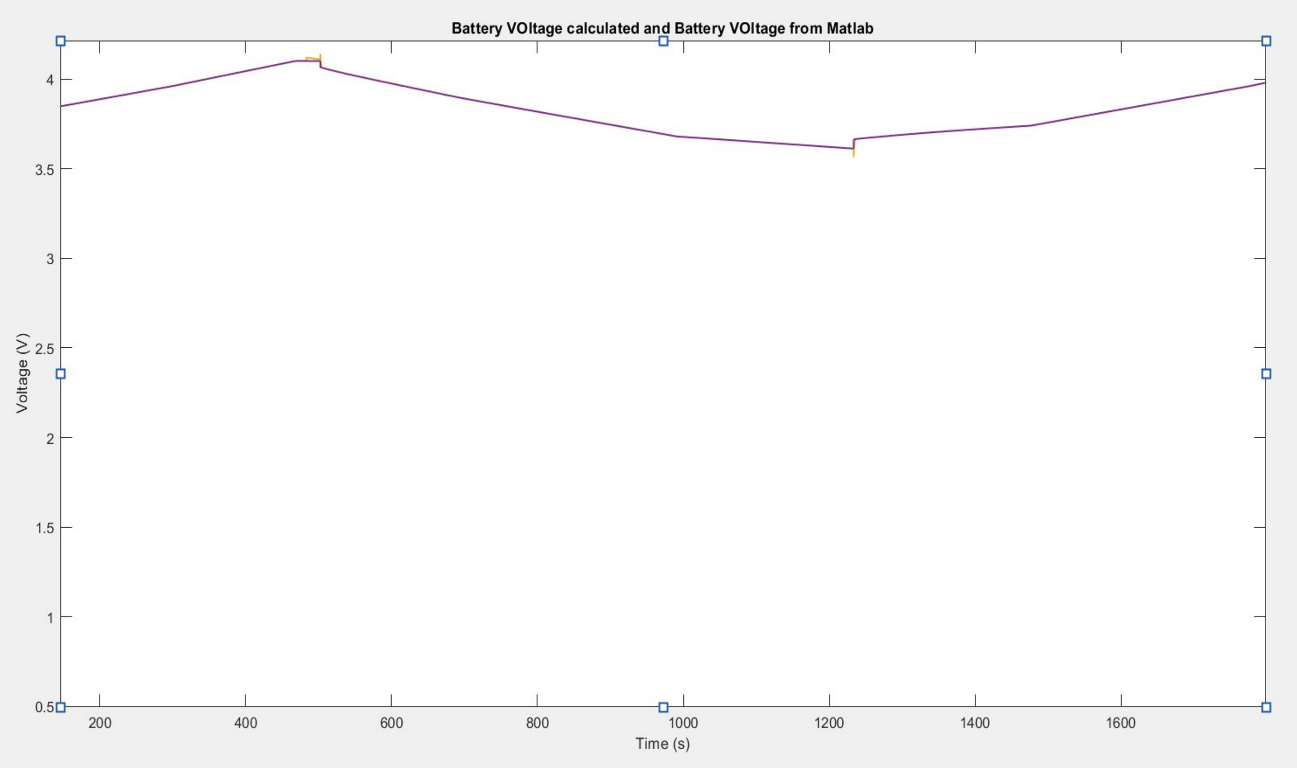
- Simulink Model and Simulation Result
  - 3A Charging and Discharging
  - 15A Charging and discharging
  - 30A Charging and discharging

3A Charging and Discharging



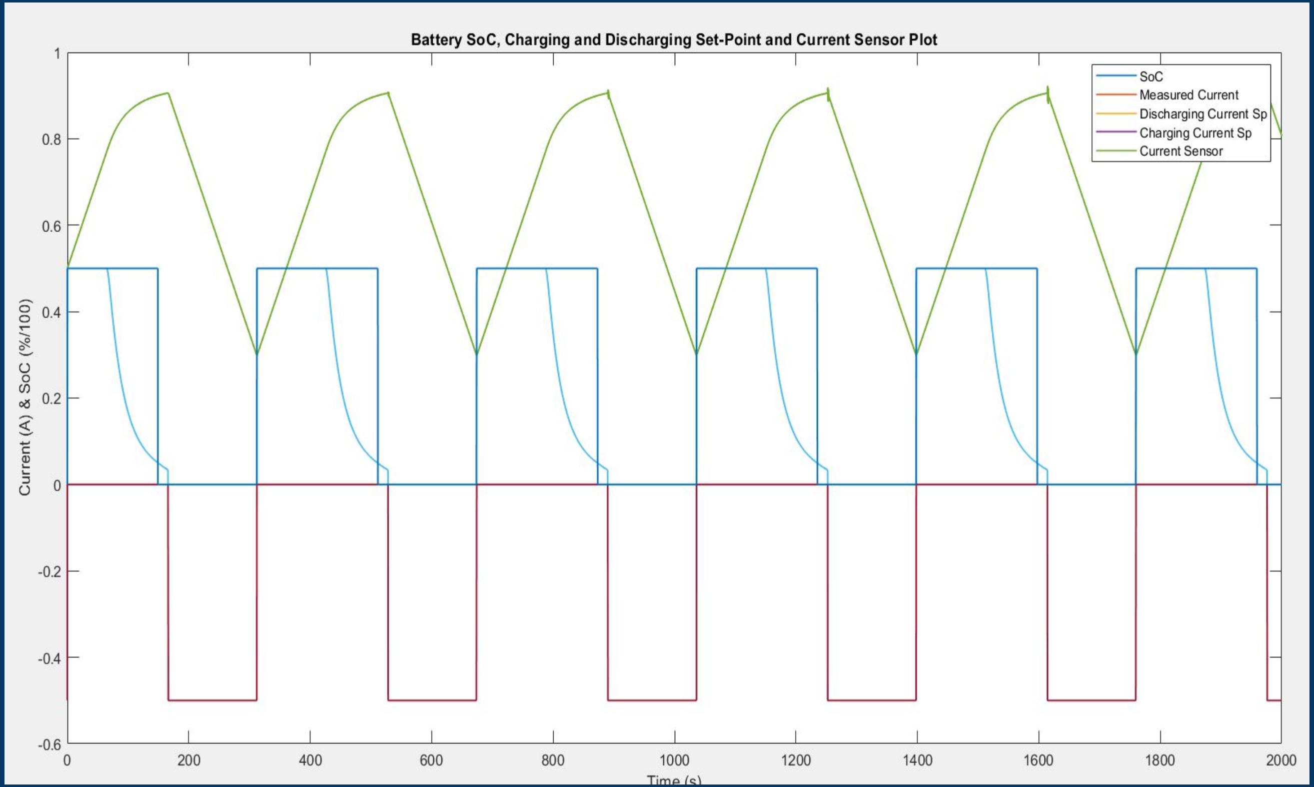
- SoC
  - 50-90%
- CHcurrSP
  - Max 30A
- DCHcurrSP
  - Max 30A
- ACcurrSEN

3A Charging and Discharging



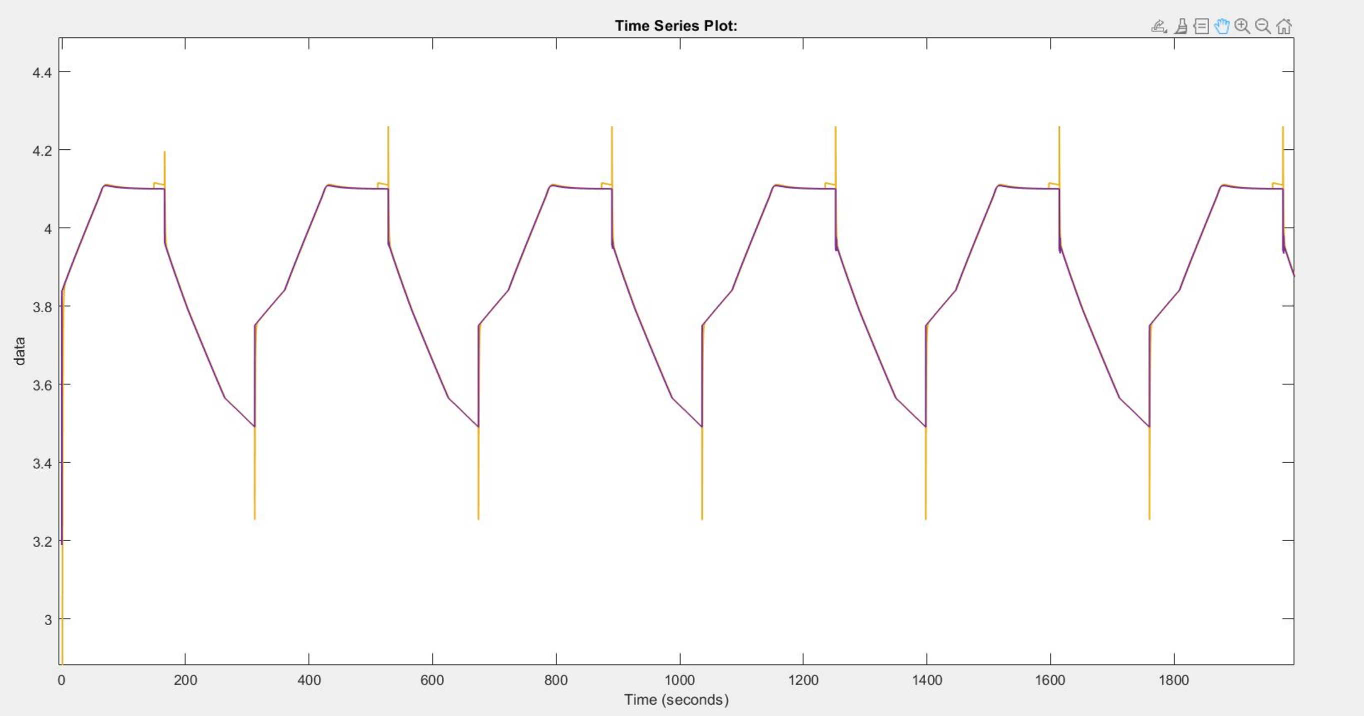
- BattVolMATLAB
- BattVolCal

15A Charging and Discharging



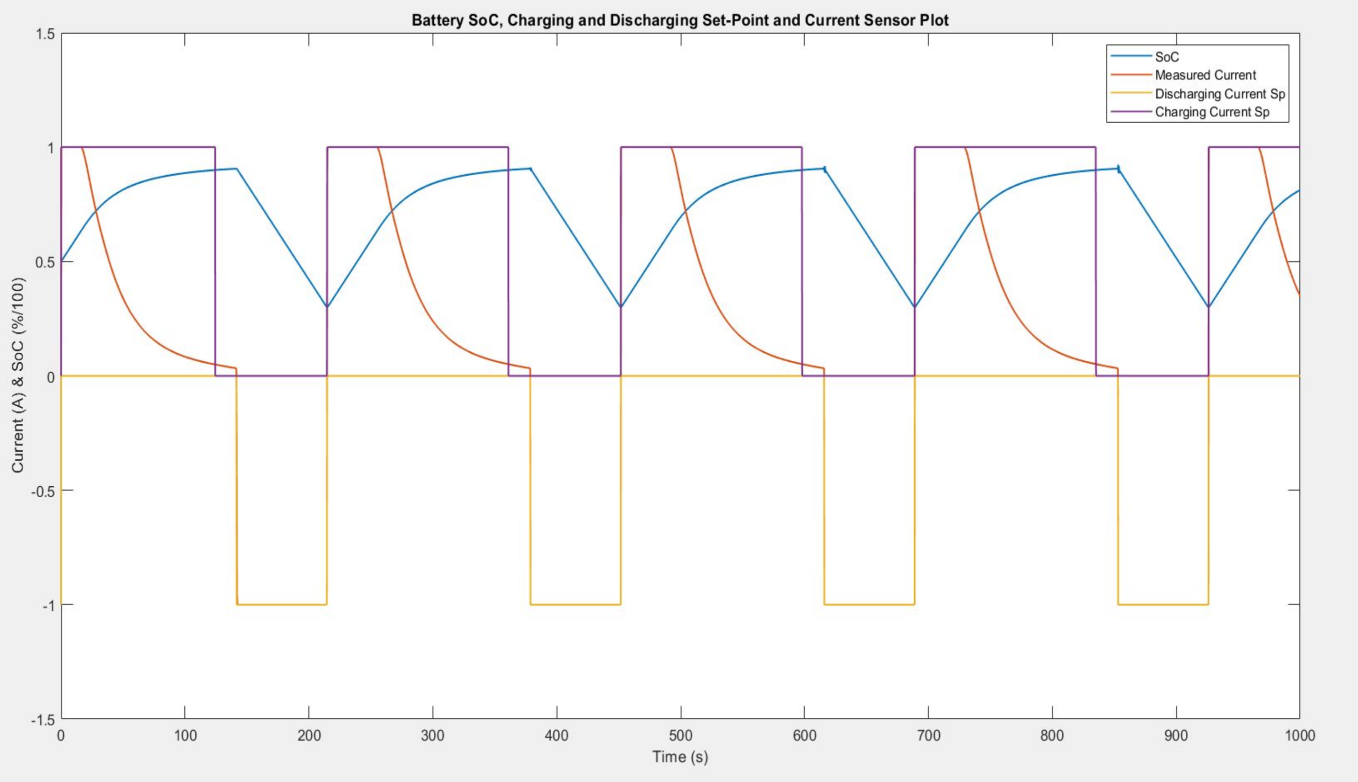
- SoC
  - 50-90%
- CHcurrSP
  - Max 30A
- DCHcurrSP
  - Max 30A
- ACcurrSEN

15A Charging and discharging



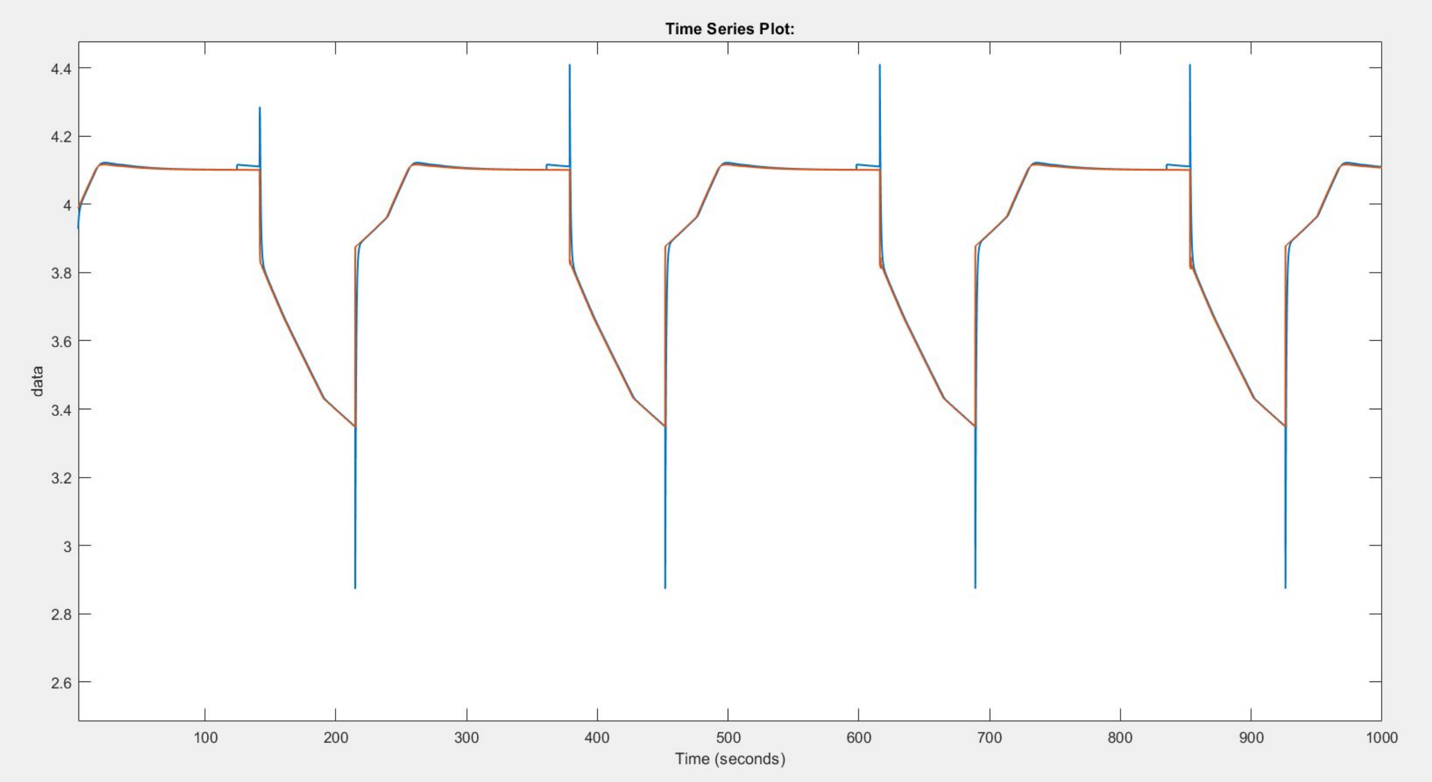
- BattVolMATLAB
- BattVolCal

30A Charging and Discharging



- SoC
  - 50-90%
- CHcurrSP
  - Max 30A
- DCHcurrSP
  - Max 30A
- ACcurrSEN

30A Charging and Discharging



BattVolMATLAB
BattVolCal

Electronic Circuit Design, Modeling and Testing

Basic working principle of CCCV Circuit

- An op-amp is signal amplifier
    - DC gain 10^5 to 10^7

VO=A*vd

vd=VI-vn

vn=ß*VO

vd=VI-ß*VO

VO/A=VI-ß*VO

VO/VI=A/(1+A*ß)

VO/VI=(1/ß)*(1/(1+(1/A*ß)))

1/ß :- Ideal circuit gain

1/(1+(1/A*ß)) :- gain accuracy or real circuit gain

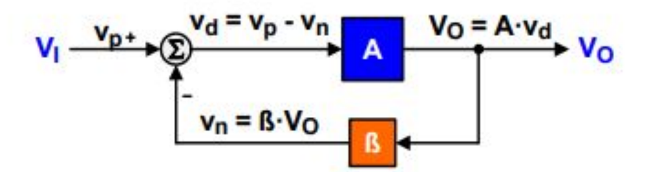


Figure 3. Model of Non-Inverting Amplifier Circuit

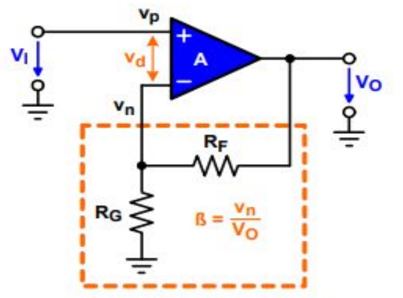


Figure 2. Amplifier Circuit

https://www.renesas.com/us/en/document/oth/tb514-operational-amplifier-basics

P-MOSFET

❖ Current output depends on VDS and VGS

$$I_g = \frac{V_{GS} - V_{gs}}{R_g} \qquad (1)$$

Ig = 0

VGS=Vgs when gate is open

VDS is Power supply voltage

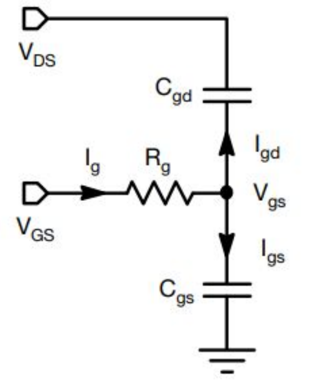


Fig. 1 - An Equivalent MOSFET Gate Circuit Showing Just C_{gs}, C_{gd}, and R_g

untitled (vishay.com)

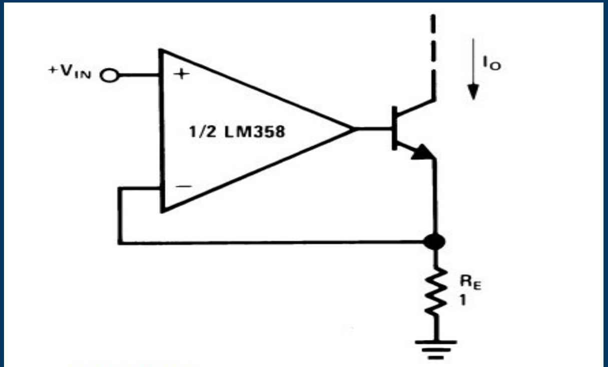
Op-Amp

LM358N

- ❖ Vp = Vn
- ❖ Ig = Io = VRE/RE

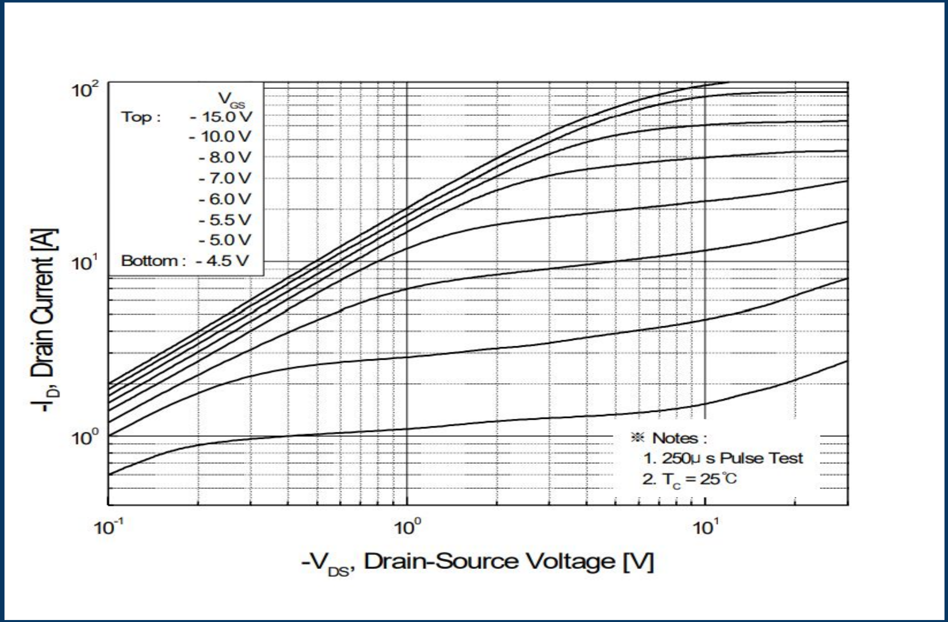
On Characteristics

$V_{GS(th)}$	Gate Threshold Voltage	$V_{DS} = V_{GS}$, $I_D = -250\ \mu A$	-2.0	--	-4.0	V
$R_{DS(on)}$	Static Drain-Source On-Resistance	$V_{GS} = -10\ V$, $I_D = -13.5\ A$	--	0.055	0.07	Ω
g_{FS}	Forward Transconductance	$V_{DS} = -30\ V$, $I_D = -13.5\ A$ (Note 4)	--	12.4	--	S



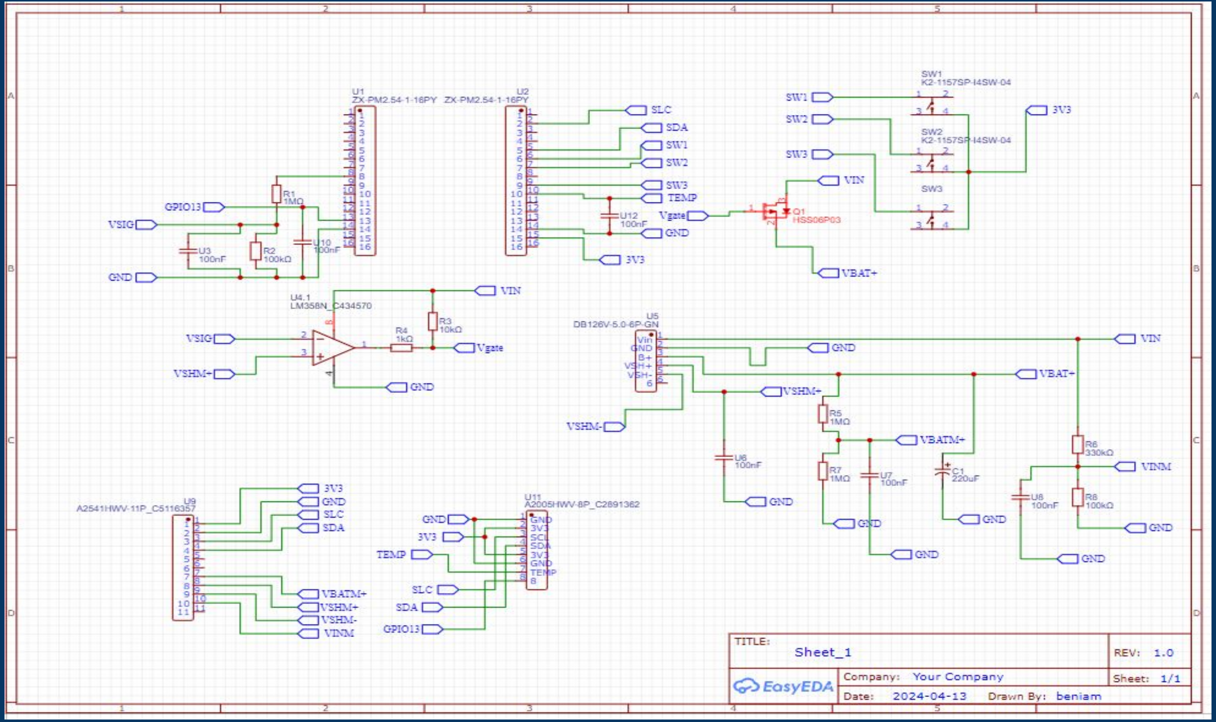
I_O = 1 amp/volt V_{IN}
(Increase R_E for I_O small)

Figure 8-14. High Compliance Current Sink



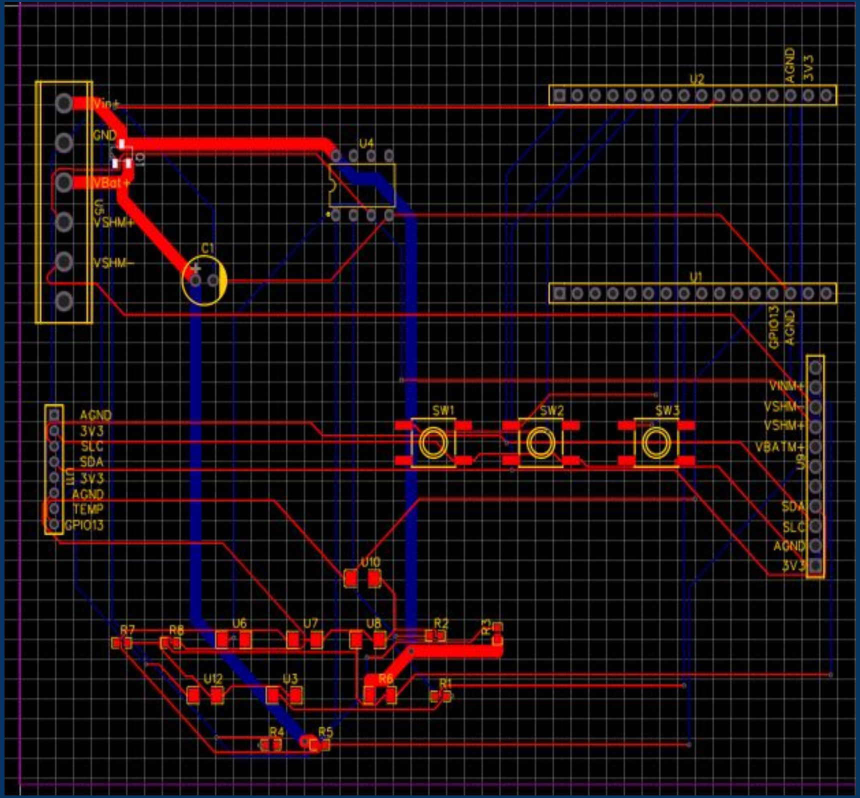
https://www.ti.com/lit/ds/symlink/lm358-n.pdf?ts=1716395150280&ref_url=https%253A%252F%252Fwww.mouser.cl%252F

Electronic Circuit Design

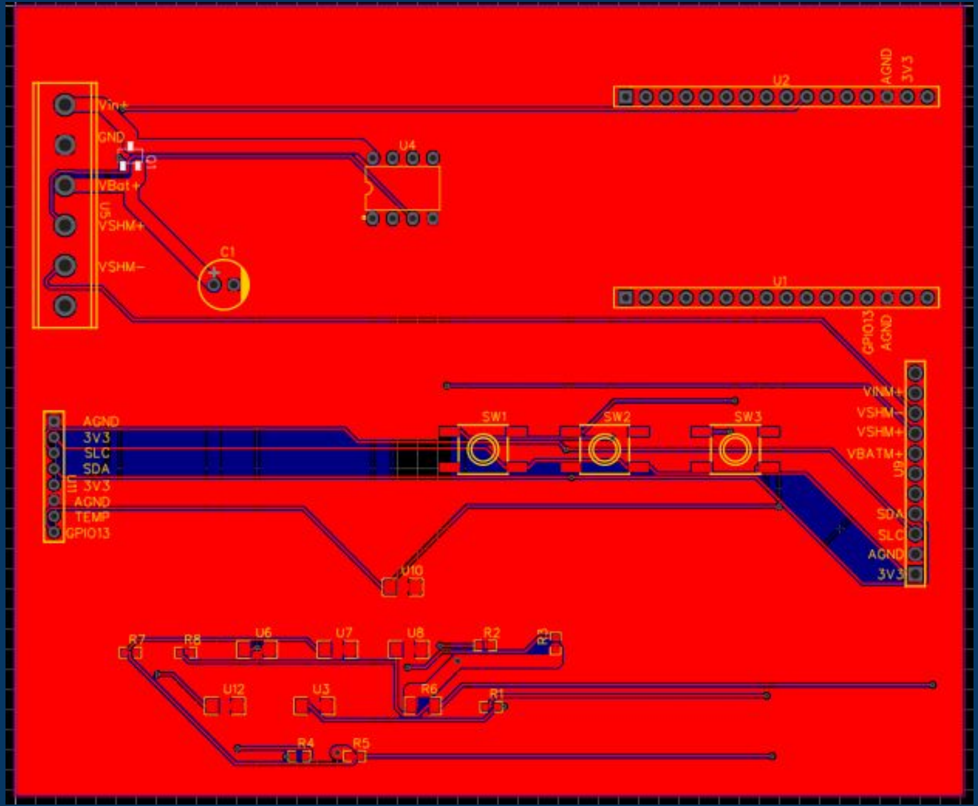


- Op-Amp
- P-Mosfet

PCB Design

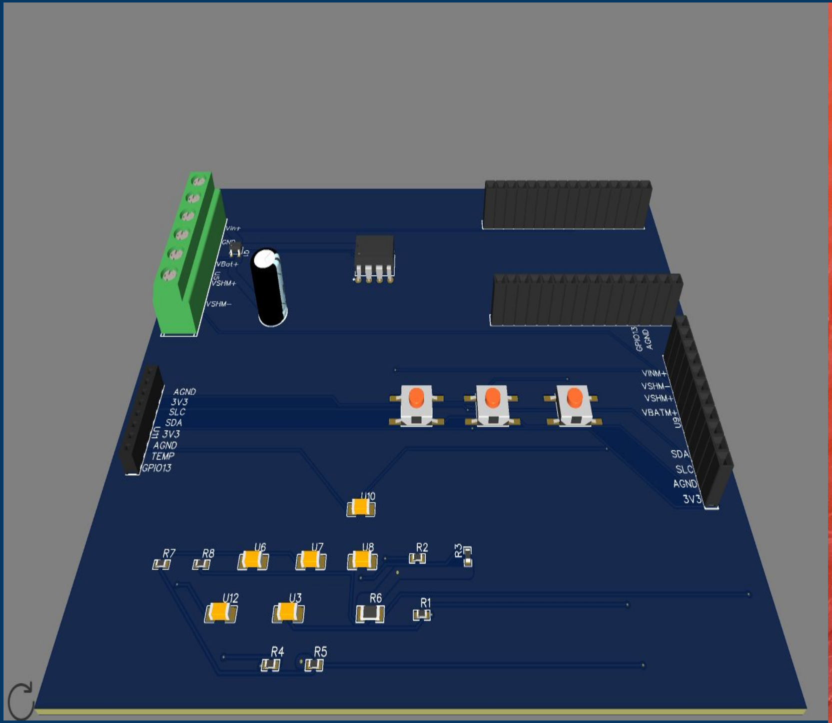


U2
AGND
3V3
Vin+
GND
VBat+
U5
VSHM+
VSHM–
U4
C1
U1
GPIO13
AGND
VINM+
VSHM–
VSHM+
VBATM+
U9
SDA
SLC
AGND
3V3
AGND
3V3
SLC
SDA
U11
3V3
AGND
TEMP
GPIO13
SW1
SW2
SW3
U10
R7
R8
U6
U7
U8
R2
R3
U12
U3
R6
R1
R4
R5

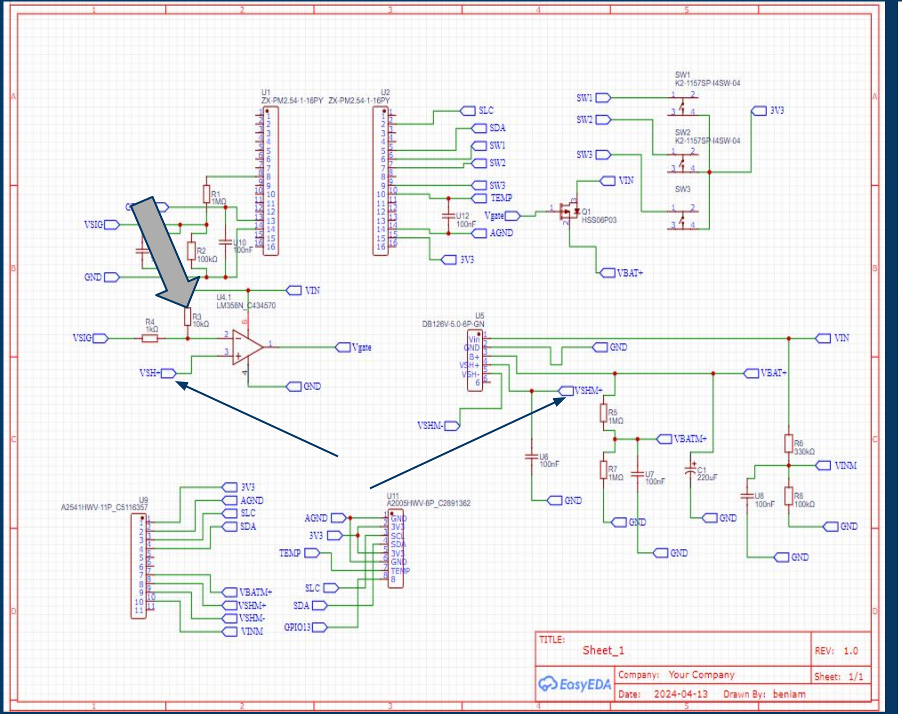


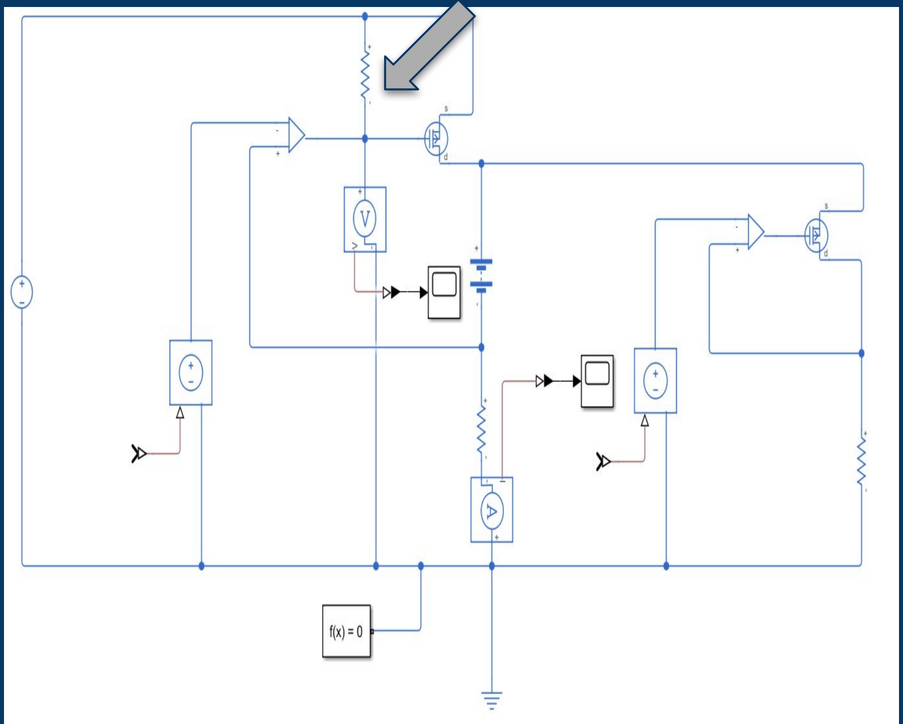
U2
AGND
3V3
Vin+
GND
VBat+
U5
VSHM+
VSHM–
U4
C1
U1
GPIO13
AGND
VINM+
VSHM–
VSHM+
VBATM+
U9
SDA
SLC
AGND
3V3
AGND
3V3
SLC
SDA
U11
3V3
AGND
TEMP
GPIO13
SW1
SW2
SW3
U10
R7
R8
U6
U7
U8
R2
R3
U12
U3
R6
R1
R4
R5

PCB 3D Design Vs Printed Circuit

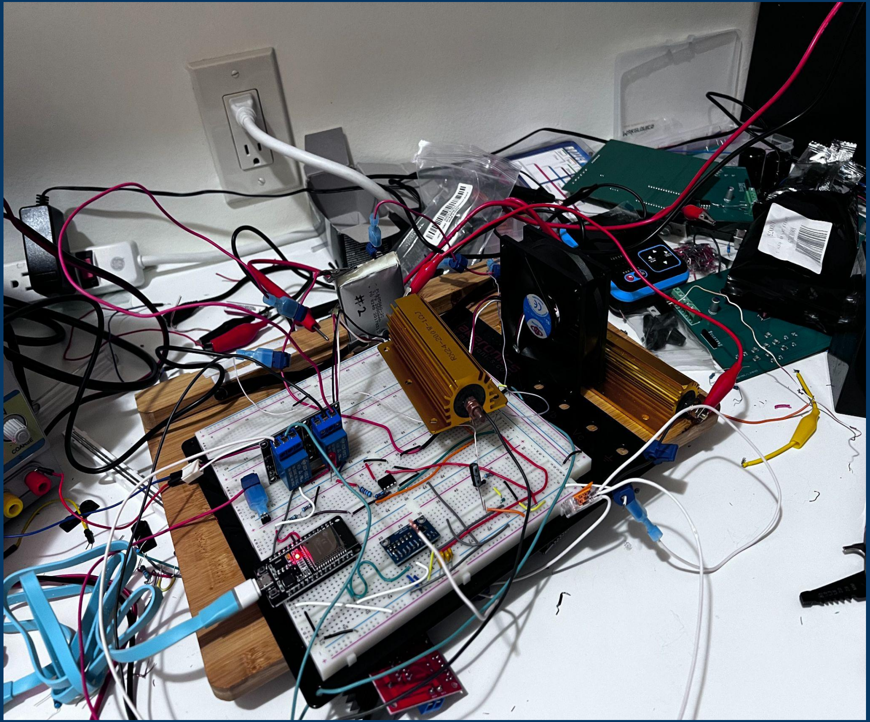


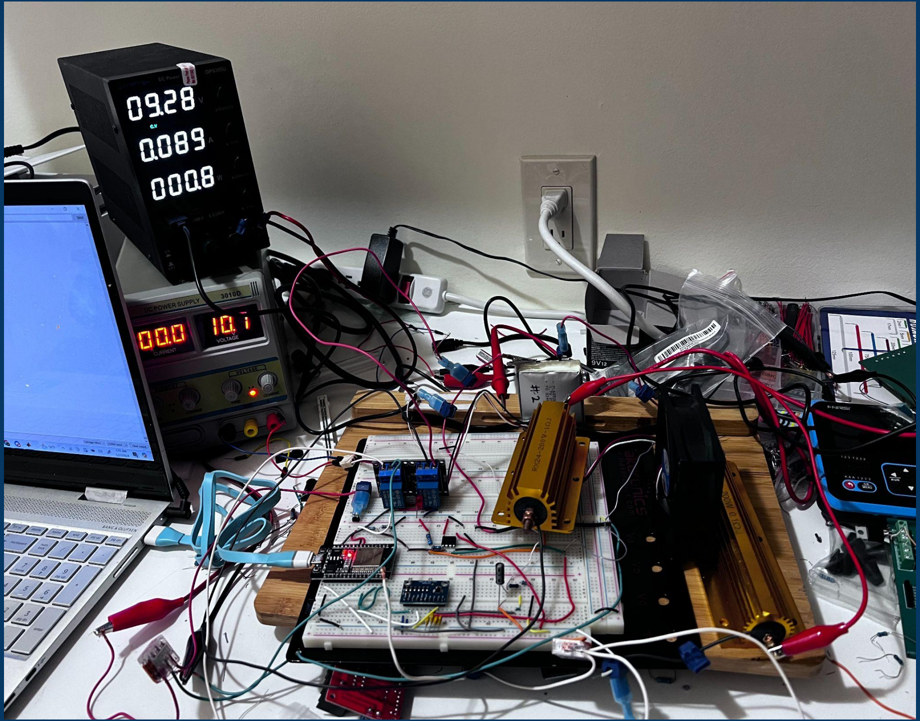
PCB Design Error





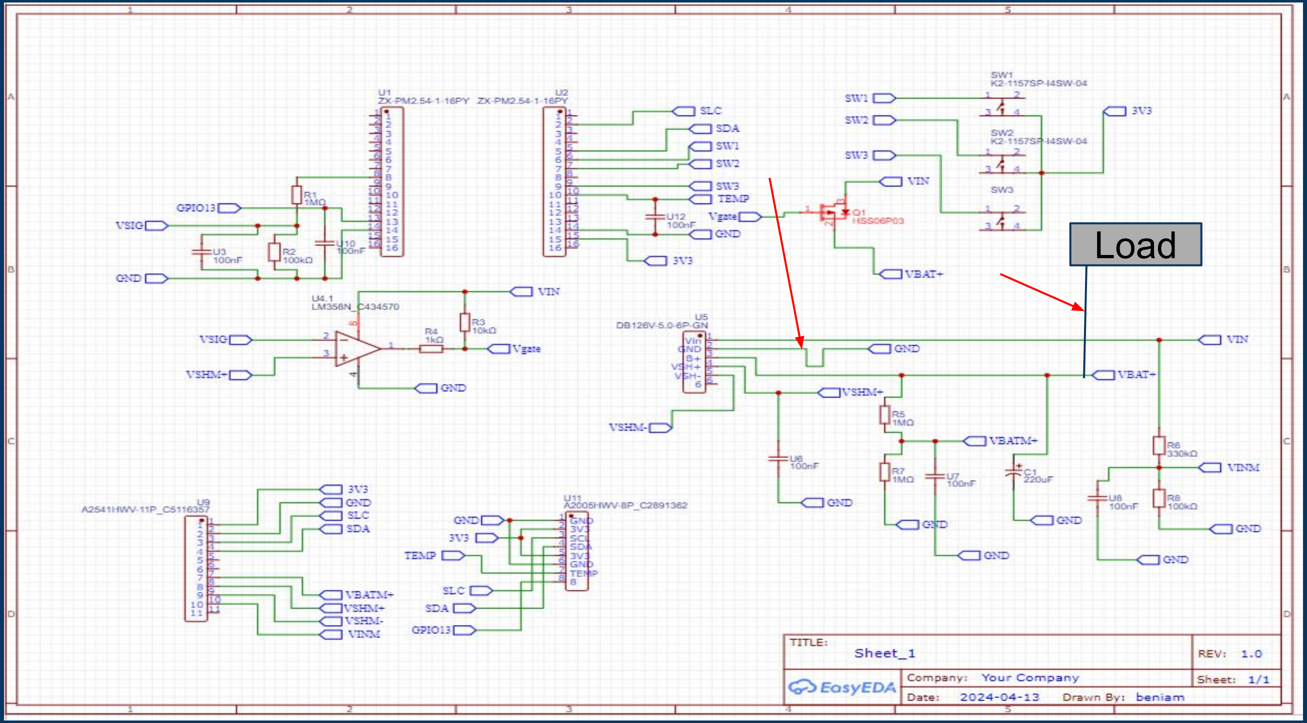
Working Prototype of CCCV Circuit with Discharge





Change Made to Incorporate Discharging

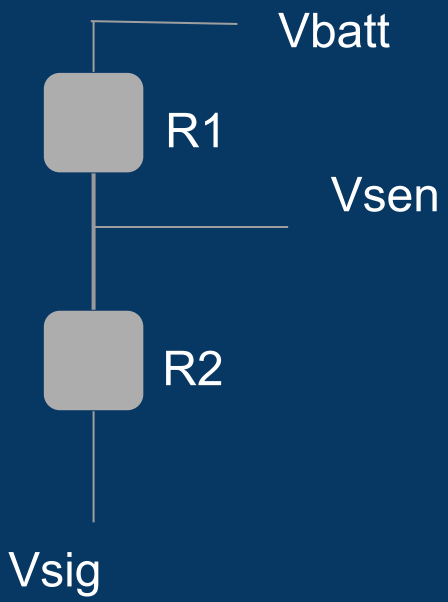
- Added 2 Relays



Voltage Sensor using voltage divider

R1=219K ohm

R2=103k Ohm



Points OF reading

- Battery + terminal

Batt Vol = Vbatt+*((R2+R1)/R2)-Vshun+

- Voltage signal
  R3=4.7 M ohm
  R4=4.7 K ohm

Vsig=(R4/(R3+R4))*VsigEsp

CCCV Control

- Constant Current Control
  - Controlled by Op-Amp gain
    - A= Vo/(Vn-Vp)
- Constant Voltage
  - Controlled by Protinal Controller in Arduino IDE code
    - Charging current above 2 Amps K=10
    - Charging current below 2 Amps K=2
  - Code
    - Deadband 0.04 V

```
if(BatVol>4.19 && BatVol<4.23){
 trigVal= trigVal+K*( 4.2-BatVol);
 dacWrite(dac2, trigVal );}
 else if ( BatVol<4.19){
 trigVal= trigVal+K*( 4.2-BatVol);
 dacWrite(dac2, trigVal );
```

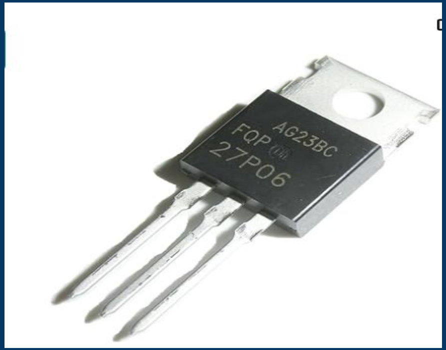
  -

Components Used in the Prototype

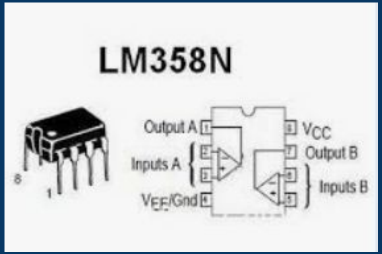
Load Resistor 1 ohm 200 W

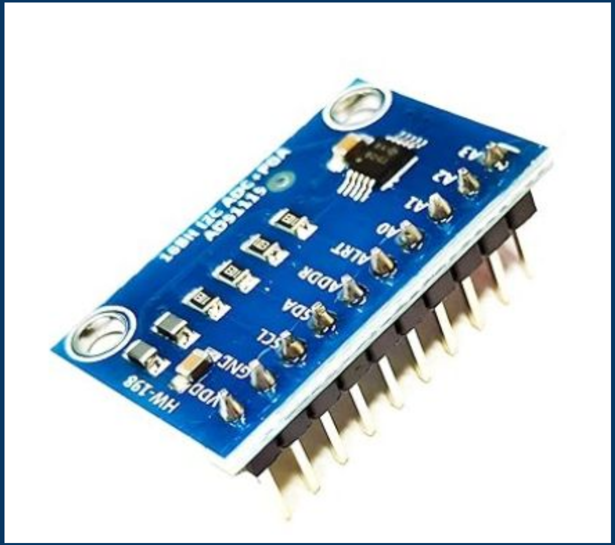
Shunt Resistor 0.1 ohm 200W

Components used in the Prototype

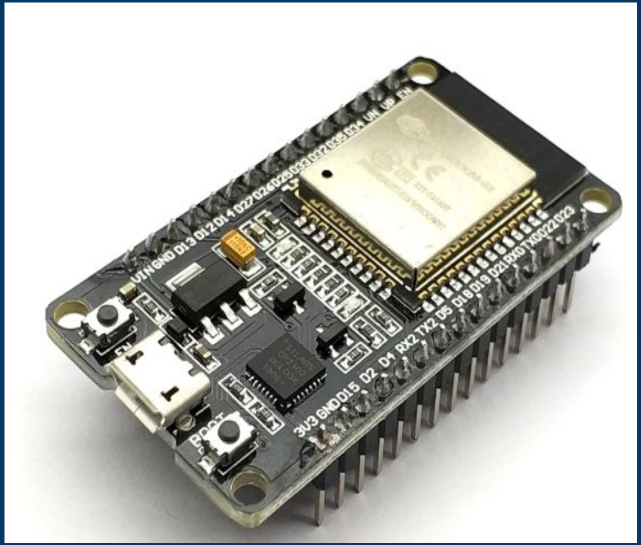


FQP27P06





ADS1115



ESP32 DevKit 1

Software

- Charging Circuit
  - Ask user Desired Charging current setpoint 0 to 3 Amps

```
chan2.trigger();
dacWrite(dac1,  0 );
dacWrite(dac2,  0 );
    Serial.println("INPUT DESIRED CHARGING CURRENT MAX SP 3 Amps");
        while (Serial.available() == 0) { }
         CHCU = Serial.parseFloat();
        CHSigSP=CHCU*RSH;
    if (CHCU <= 3.3 && CHCU > 0) {
              Serial.println(" ");Serial.print("Time (s)");Serial.print(" , ");Serial.print("Shunt V");Serial.print(" , ");Serial.print("BatVol");Serial.print(" , ");Serial.print("Chr

               adc0 = ads.readADC_SingleEnded(0);
               adc1 = ads.readADC_SingleEnded(1);
               adc2 = ads.readADC_SingleEnded(2);
               adc3 = ads.readADC_SingleEnded(3);

               volts0 = ads.computeVolts(adc0);
               volts1 = ads.computeVolts(adc1);
               volts2 = ads.computeVolts(adc2);
               volts3 = ads.computeVolts(adc3);
               BatVol = (volts0*VBG) - (volts1*VSRG);
               VolSH = (volts1*VSRG) - (volts2);
               CHCurrmes = VolSH / RSH;
```

Software

- CC Charging

```
while(BatVol < 4.2) {
                CHSigSP=CHCU*RSH;
                v=CHSigSP*(255/0.3);
                trigVal=v;
                dacWrite(dac2, trigVal );
                dacWrite(dac1, 0 );

              adc0 = ads.readADC_SingleEnded(0);
              adc1 = ads.readADC_SingleEnded(1);
              adc2 = ads.readADC_SingleEnded(2);
              adc3 = ads.readADC_SingleEnded(3);

             volts0 = ads.computeVolts(adc0);
             volts1 = ads.computeVolts(adc1);
             volts2 = ads.computeVolts(adc2);
             volts3 = ads.computeVolts(adc3);

              BatVol = (volts0*VBG) - (volts1*VSRG);
              VolSH = (volts1*VSRG) - (volts2);
              CHCurrmes = VolSH / RSH;
              VsigR= VolSH*(1/VGsig);
Serial.println(" ");Serial.print( millis());Serial.print("   ,   ");Serial.print(VolSH,4);Serial.print("   ,   ");Serial.print(BatVol,4);Serial.print("    ,    ");Serial.print(CHCurrmes,4);Seria
   delay(100);}

   if(BatVol > 4.18 && BatVol < 4.25 && CHCurrmes>0.001 )
```

Software

- CV Charging

```
    if(BatVol > 4.18 && BatVol < 4.25 && CHCurrmes>0.001 )
                                        {
      Serial.println(" ");Serial.print("constant vlotage mode ");
                         v=CHSigSP*255/0.315;
                         trigVal=v;
                           if (CHCU >= 2)
             K=10;
             else
             K=2;

while (BatVol> 4.16 && BatVol < 4.25 && CHCurrmes>0.5){

adc0 = ads.readADC_SingleEnded(0);
adc1 = ads.readADC_SingleEnded(1);
adc2 = ads.readADC_SingleEnded(2);
adc3 = ads.readADC_SingleEnded(3);

volts0 = ads.computeVolts(adc0);
volts1 = ads.computeVolts(adc1);
volts2 = ads.computeVolts(adc2);
volts3 = ads.computeVolts(adc3);
BatVol = (volts0*VBG) - (volts1*VSRG);
VolSH = (volts1*VSRG) - (volts2);
CHCurrmes = VolSH / RSH;

VsigR= VolSH*(1/VGsig);
if(BatVol>4.19 && BatVol<4.23){
 trigVal= trigVal+K*( 4.2-BatVol);
dacWrite(dac2, trigVal );}

else if ( BatVol<4.19){
     trigVal= trigVal+K*( 4.2-BatVol);
dacWrite(dac2, trigVal );
                        }
Serial.println(" ");Serial.print( millis());Serial.print("  ,  ");Serial.print(VolSH,4);Serial.print("  ,  ");Serial.print(BatVol,4);Serial.print("   ,   ");Serial.print(CHCurrmes,4);Serial.print("   ,   ");Serial.print(trigVal,4);Ser
    delay(100);}
```

Software

- Discharging

```
while ( BatVol>3 ) {
                    dacWrite(dac2,  zero );
                    digitalWrite(Rs, HIGH);
                    v=DCHcurrsp*(255/0.3);
                    trigVal=v;
                    dacWrite(dac1, 255 );
                    dacWrite(dac2, 0 );
                    adc0 = ads.readADC_SingleEnded(0);
                    adc1 = ads.readADC_SingleEnded(1);
                    adc2 = ads.readADC_SingleEnded(2);
                    adc3 = ads.readADC_SingleEnded(3);

                   volts0 = ads.computeVolts(adc0);
                   volts1 = ads.computeVolts(adc1);
                   volts2 = ads.computeVolts(adc2);
                   volts3 = ads.computeVolts(adc3);


                  BatVol = (volts0*VBG) - (volts1*VSRG);
                  VolSH = (volts1*VSRG) - (volts2);
                  CHCurrmes = VolSH / RSH;
                  VsigR= VolSH*(1/VGsig);
Serial.println(" ");Serial.print( millis());Serial.print("  ,  ");Serial.print(VolSH,4);Serial.print("  ,  ");Serial.print(BatVol,4);Serial.print("   ,   ");Serial.print(CHCurrmes,4);Serial.print("
     delay(100);}
```

Data Capture and Preparation

- Arduino Serial Monitor

```
INPUT DESIRED CHARGING CURRENT MAX SP 3 Amps

Time (s) , Shunt V , BatVol , ChrgCurr A , sig 0-255 , sig V
10012  ,  0.1783  ,  3.7226  ,  1.7831  ,  170.0000  ,  1.9614
10149  ,  0.1785  ,  3.9226  ,  1.7850  ,  170.0000  ,  1.9635
10286  ,  0.1781  ,  3.9253  ,  1.7813  ,  170.0000  ,  1.9594
10423  ,  0.1779  ,  3.9273  ,  1.7794  ,  170.0000  ,  1.9573
10560  ,  0.1781  ,  3.9282  ,  1.7813  ,  170.0000  ,  1.9594
10697  ,  0.1783  ,  3.9286  ,  1.7831  ,  170.0000  ,  1.9614
10834  ,  0.1781  ,  3.9294  ,  1.7813  ,  170.0000  ,  1.9594
10971  ,  0.1779  ,  3.9308  ,  1.7794  ,  170.0000  ,  1.9573
11108  ,  0.1781  ,  3.9312  ,  1.7813  ,  170.0000  ,  1.9594
11245  ,  0.1783  ,  3.9322  ,  1.7831  ,  170.0000  ,  1.9614
```

Data Capture and Preparation

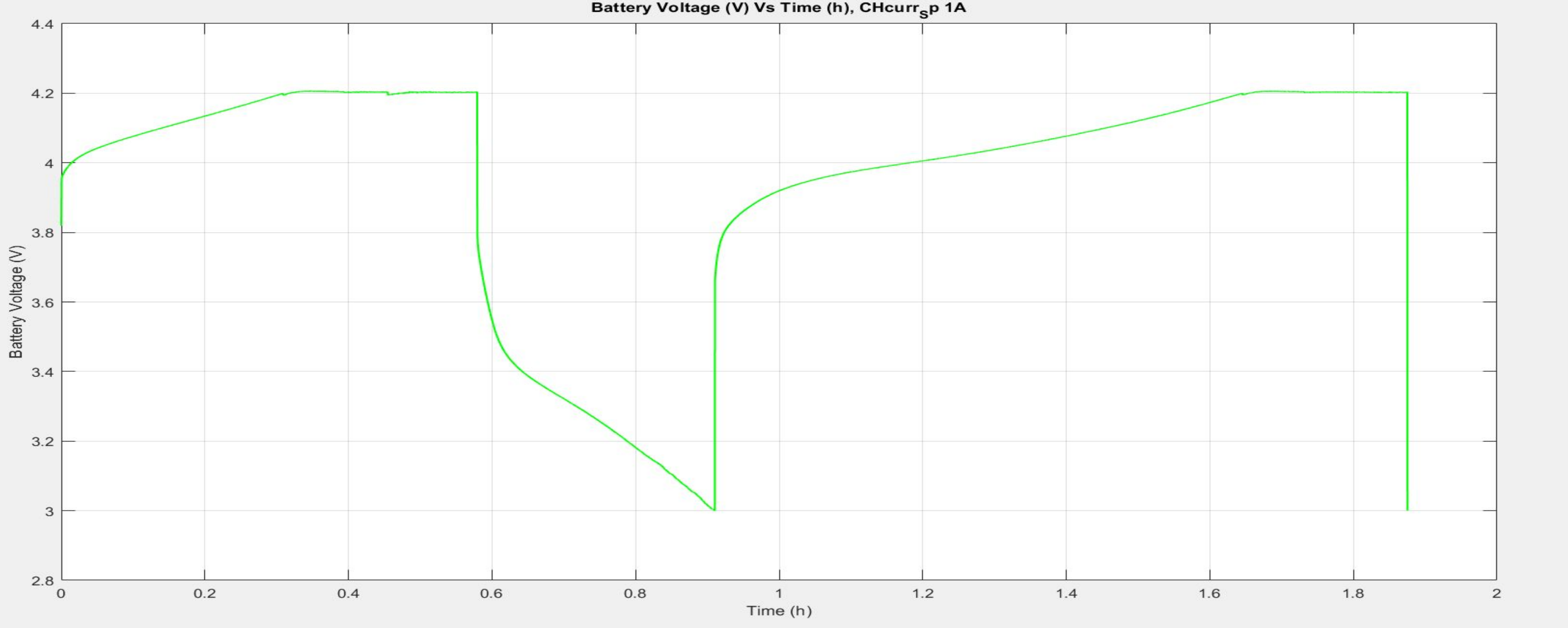
- Converting string to CSV file (Mayolo Valencia)

```python
import csv

def read_csv_data(filename):

  """
  This function reads data from a CSV file, removes quotation marks,
  converts elements to floats (up to 6 columns), and returns a list of lists.
  """
  data = []
  with open(filename, 'r') as csvfile:
    reader = csv.reader(csvfile)
    next(reader)  # Discard the first row (assuming it's the header)

    for row in reader:
      # Remove quotes from each element and limit to 6 elements

      row = row[0].replace(" ", "")
      clean_row = [element.strip(',') for element in row]

      clean = []
      combined_string= ''

      for el in clean_row:
        if el != '':
          combined_string = combined_string + "" + el
        else:
          clean.append(combined_string)
          combined_string= ''

      # Convert elements to floats (handling potential errors)
      try:
        float_row = [float(element) for element in clean]
      except ValueError:
        # Handle potential conversion errors (e.g., non-numeric elements)
        print(f"Error converting row: {clean_row}. Skipping...")
        continue

      data.append(float_row)
```

```python
def write_csv_data(filename, data):
  """
  This function writes a list of lists (data) to a new CSV file.
  """
  with open(filename, 'w', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerows(data)


# Specify the file that you want to fix, upload the file to the notebook then copy its path here.
original_csv = "/content/2A001.csv"  # Replace with your actual filename

# Read data from the CSV file
csv_data = read_csv_data(original_csv)

# add the csv path here that will take in the new fixed data
new_csv = "/content/fixed_file2.csv"

try:
    with open(new_csv, 'x') as file:
        pass
except FileExistsError:
    pass

write_csv_data(new_csv,csv_data)
# Print the data (optional)
if csv_data:
  print("Cleaned and converted CSV data (limited to 6 columns):")
  for row in csv_data:
    print(row)
else:
  print("No data found in the CSV file.")
```

https://colab.research.google.com/drive/17U5Wobr3vXstTqV0AfxATHZdIaKNtinz#scrollTo=jZYQS4iMrgpT

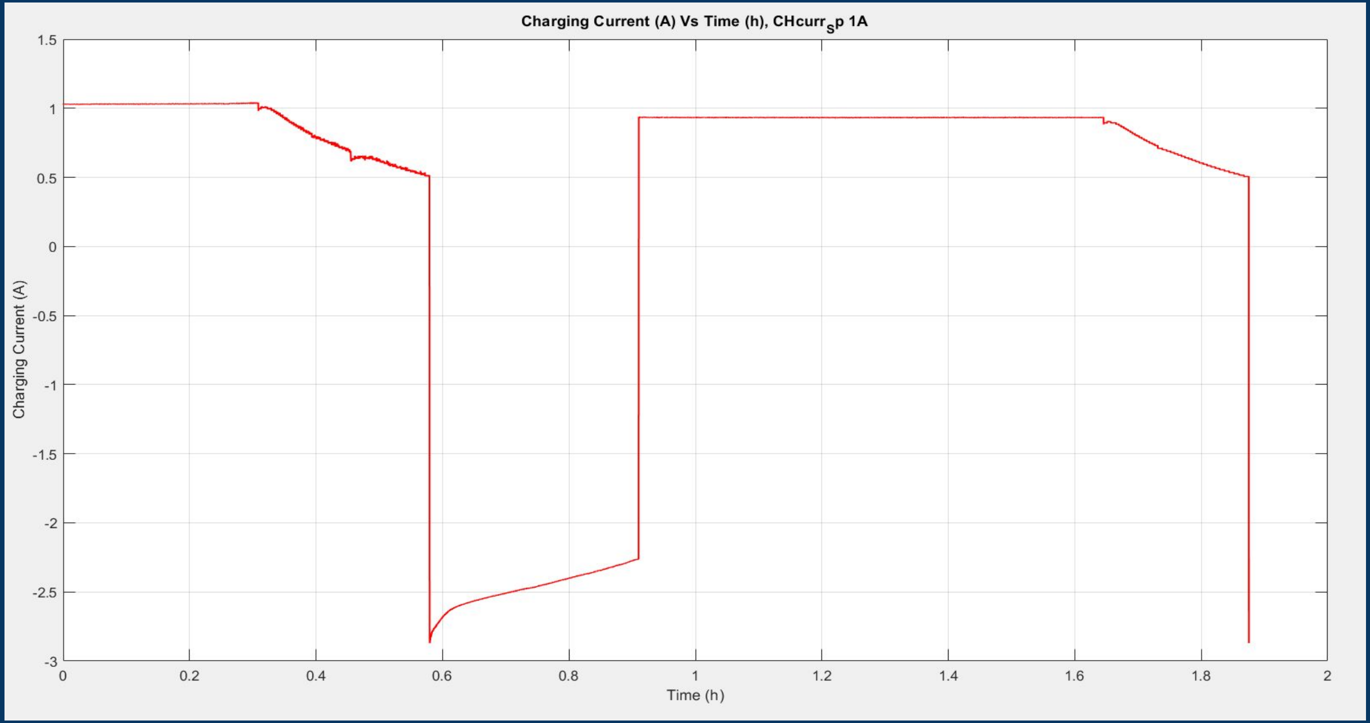
Charging and Discharging Test Results 1A



Battery Voltage (V) Vs Time (h)

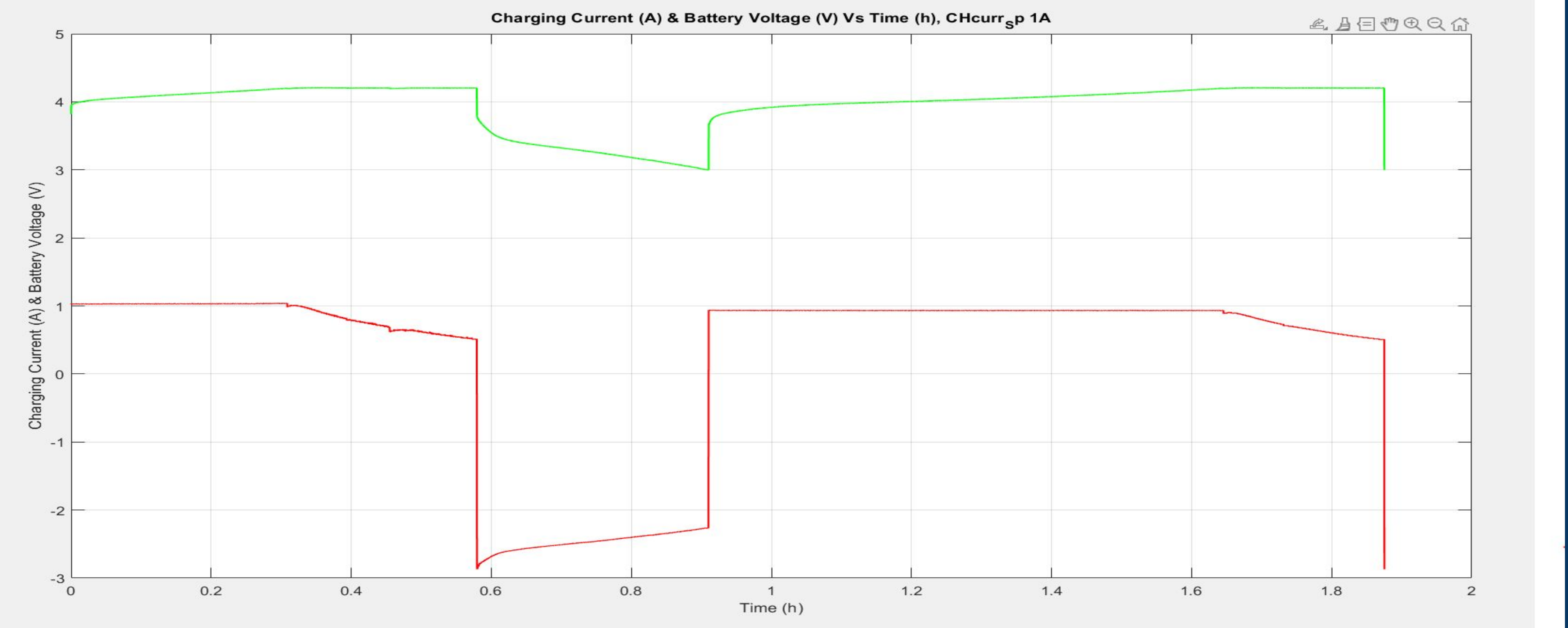
Charging and Discharging Test Results 1A

- Error 6.5 %



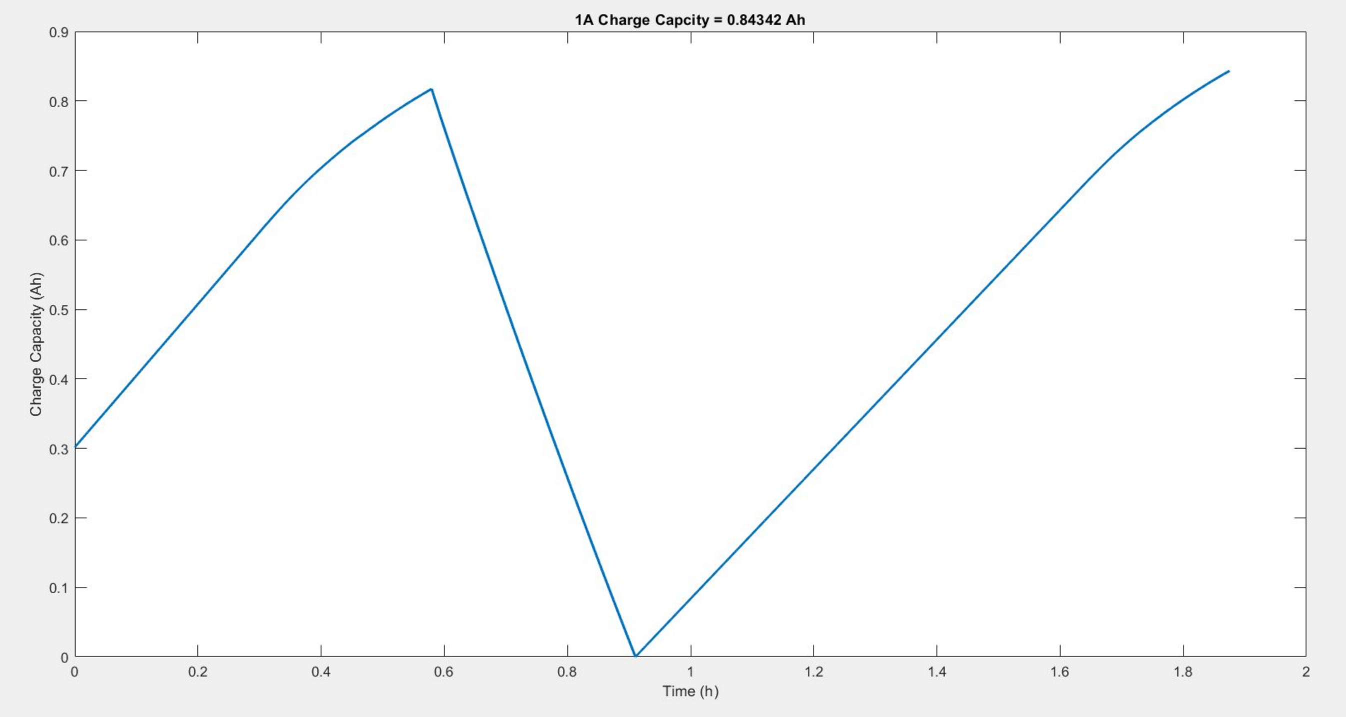
Charging Current (A) Vs Time (h)

Charging and Discharging Test Results 1A

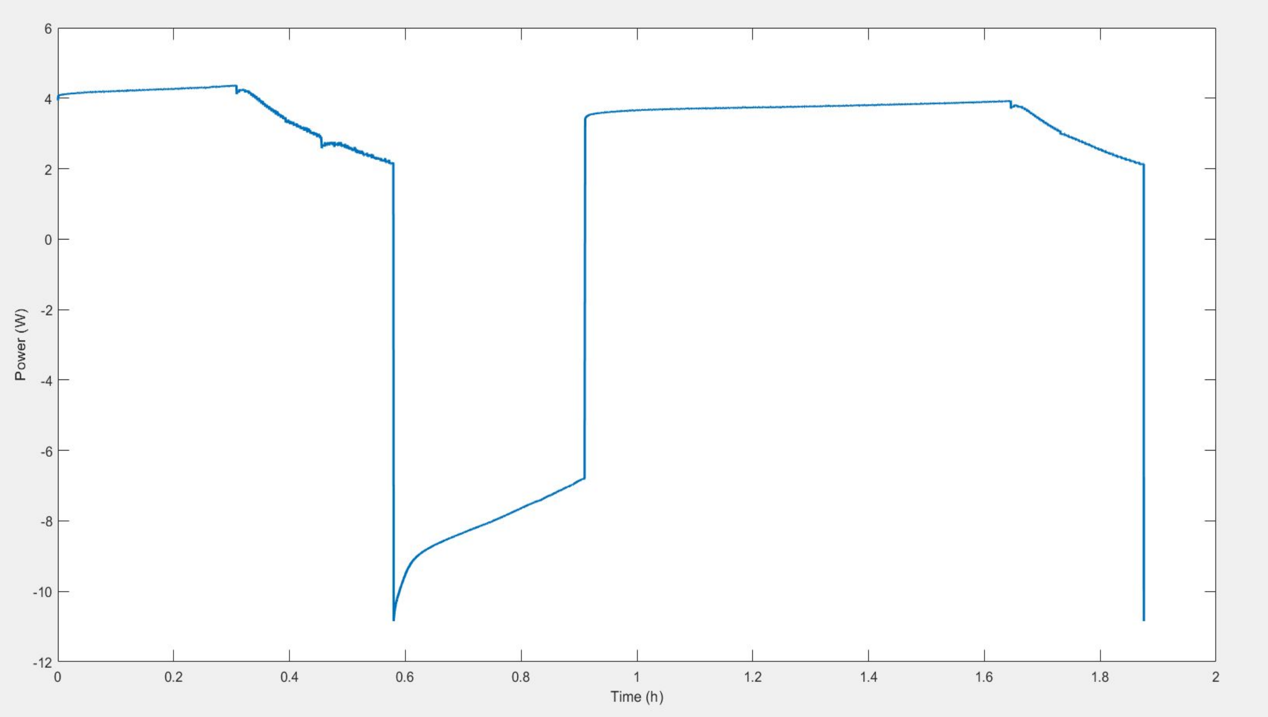


Charging Current (A) &Battery Voltage (V) Vs Time (h)

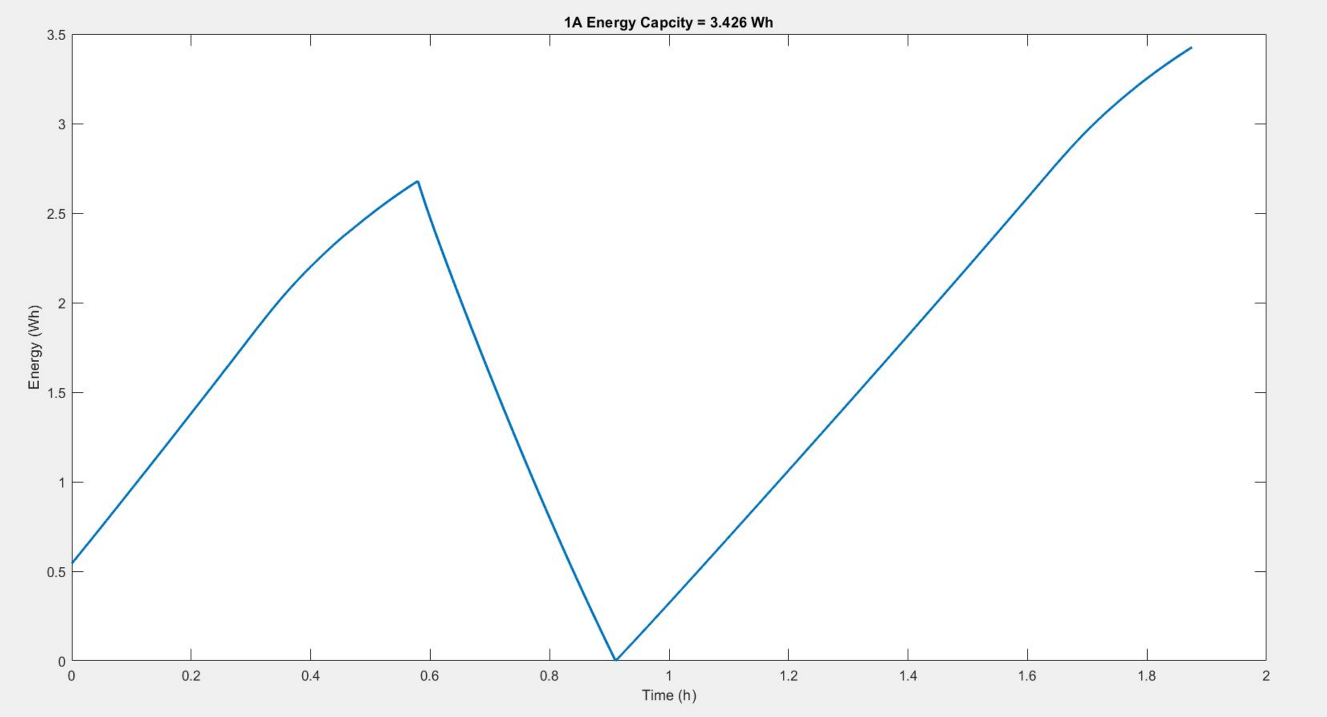
SoC at 1A Charging



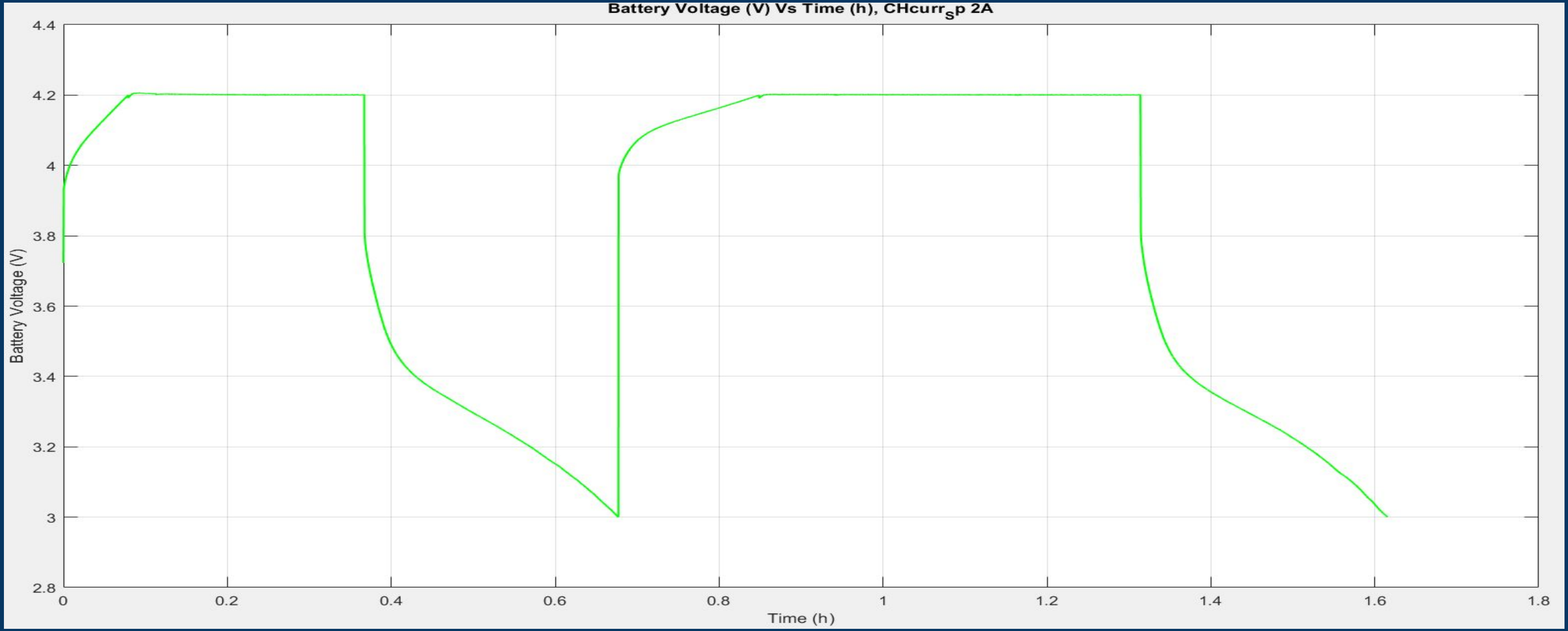
Power (W) Vs Time (h) at 1A



Energy Capacity at 1A Charging



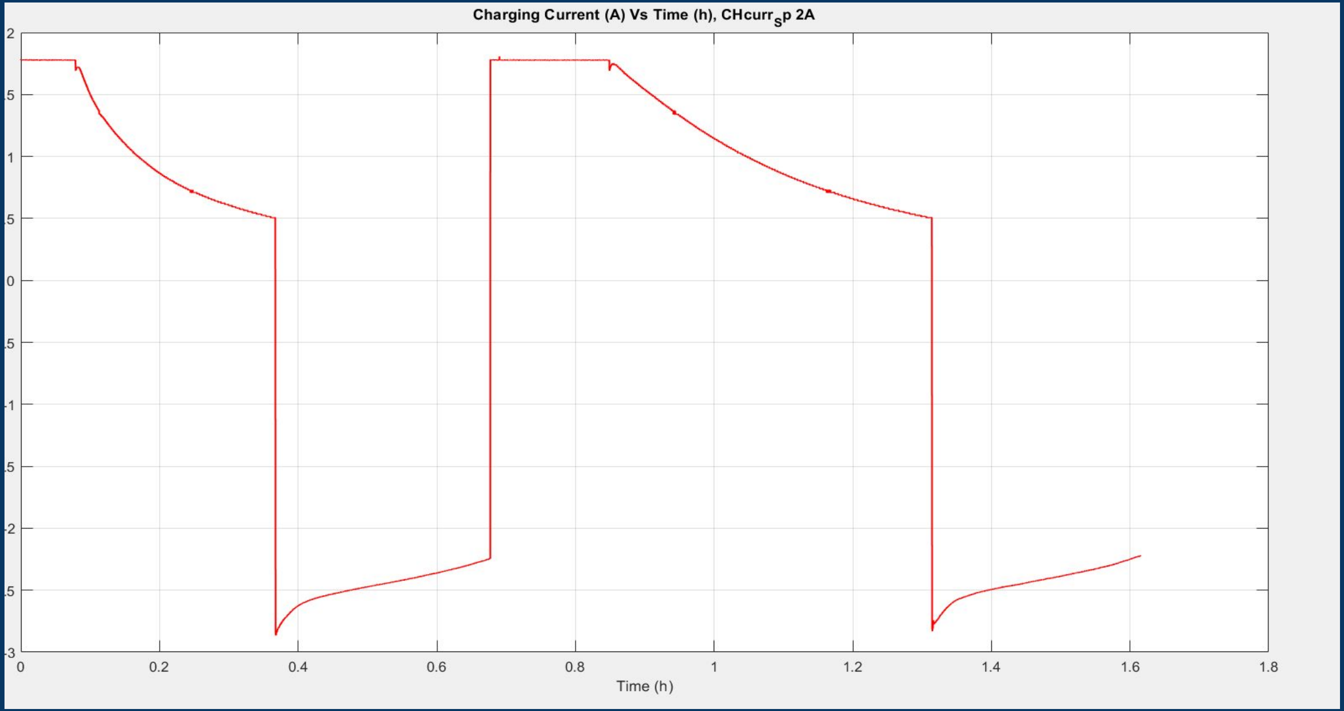
Charging and Discharging Test Results 2A



Battery Voltage (V) Vs Time (h)

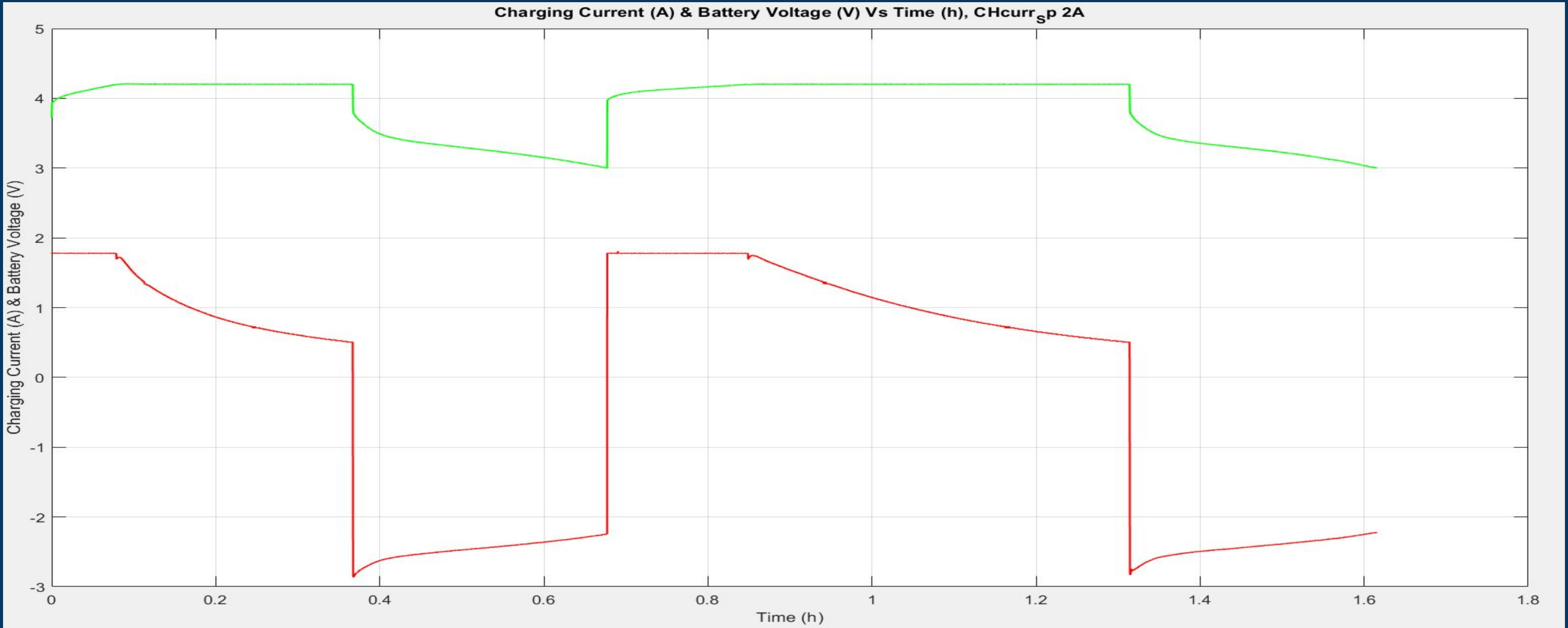
Charging and Discharging Test Results 2A

- Error: 11.5%



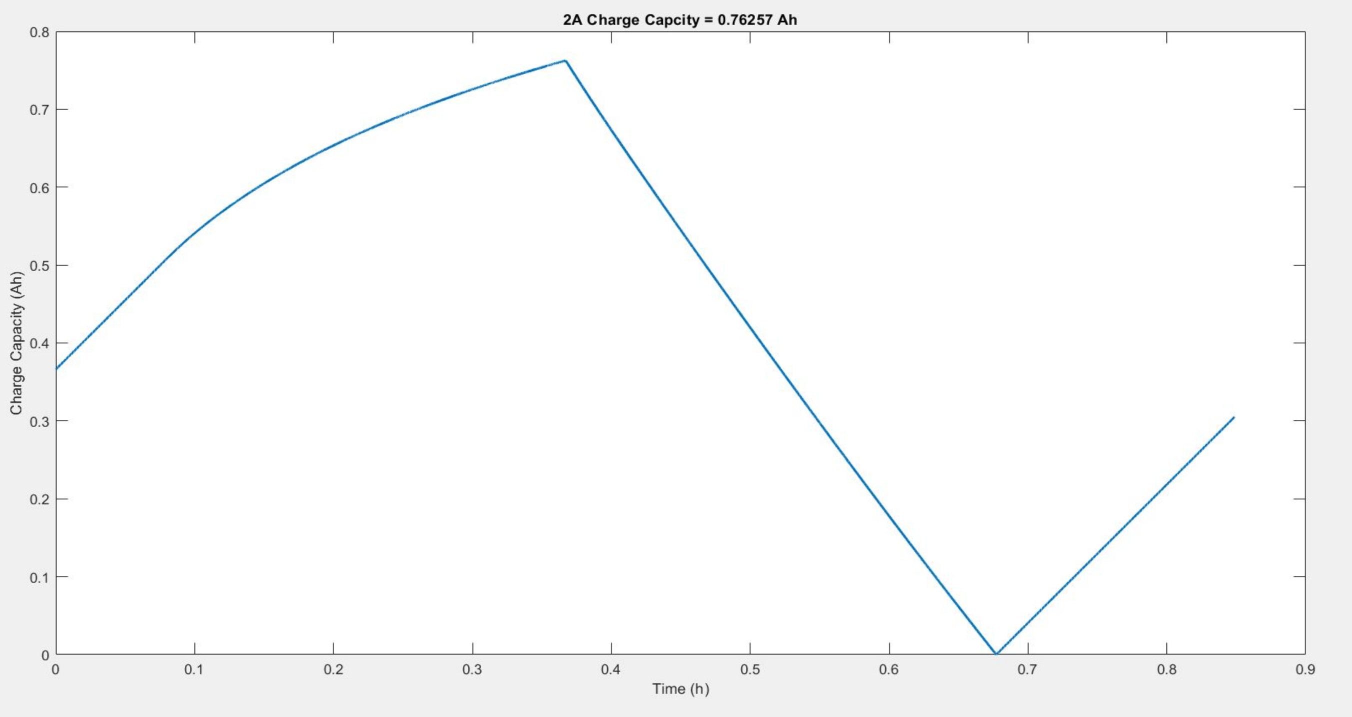
Charging Current (A) Vs Time (h)

Charging and Discharging Test Results 2A

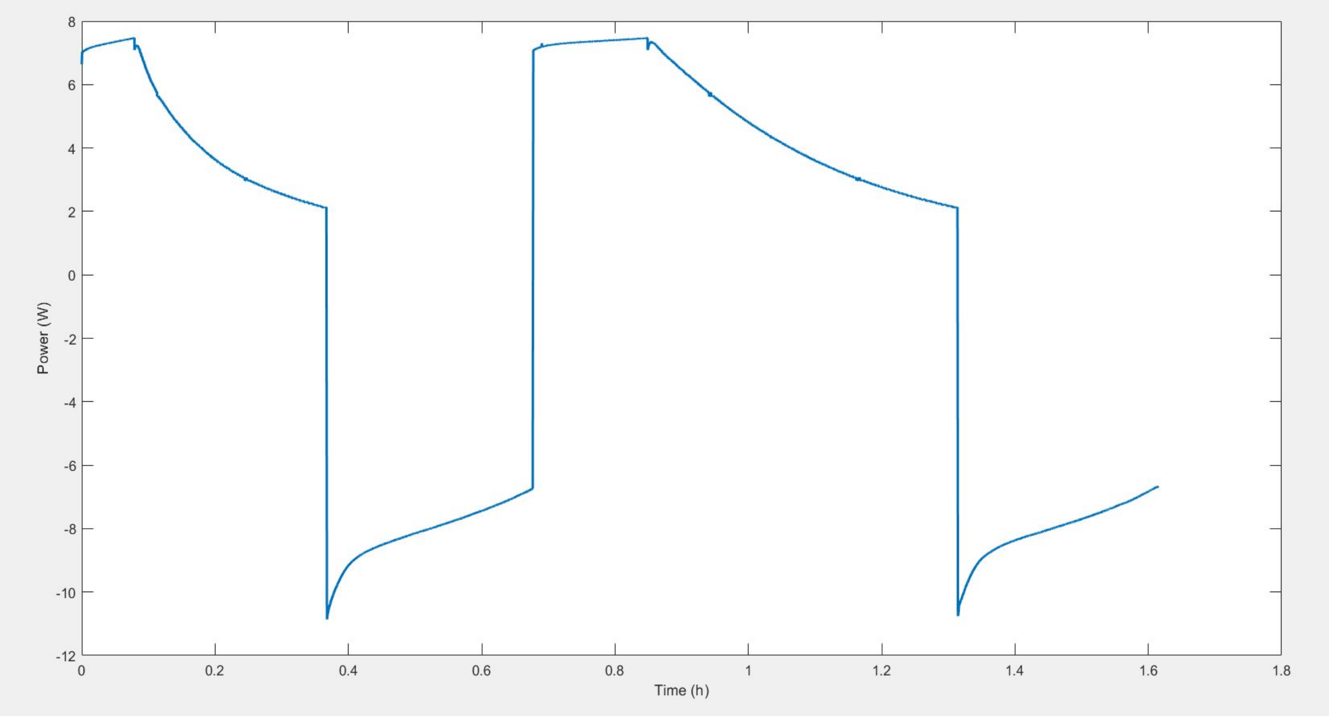


Charging Current (A) &Battery Voltage (V) Vs Time (h)

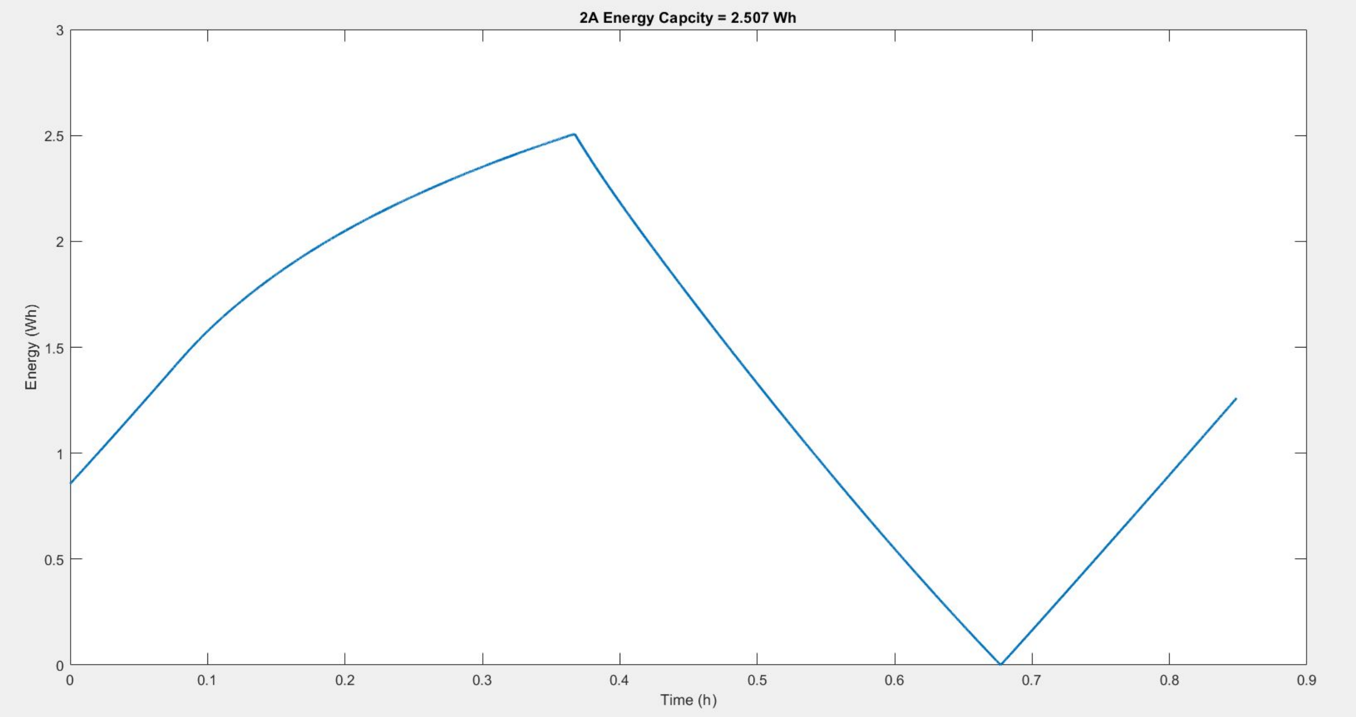
SoC at 2A Charging



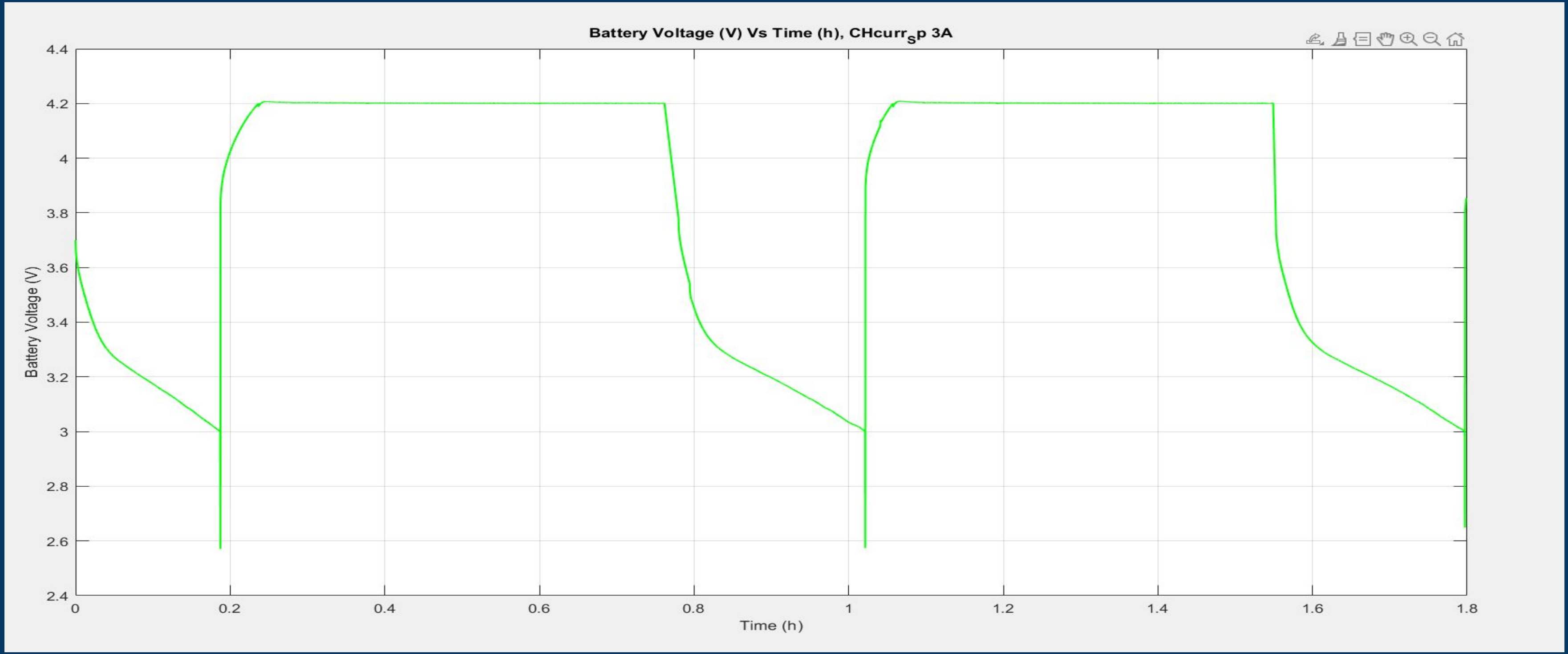
Power (W) Vs Time (h) at 2A



Energy Capacity at 2A Charging



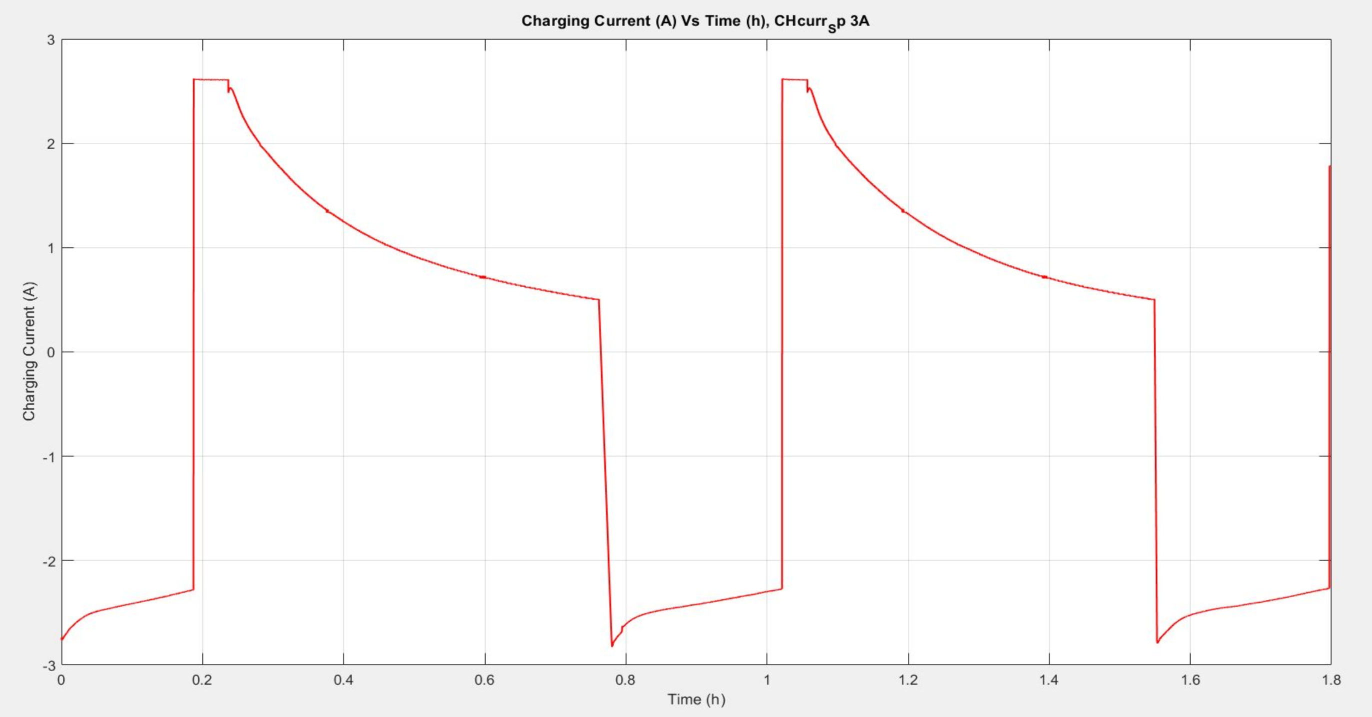
Charging and Discharging Test Results 3A



Battery Voltage Vs Time h

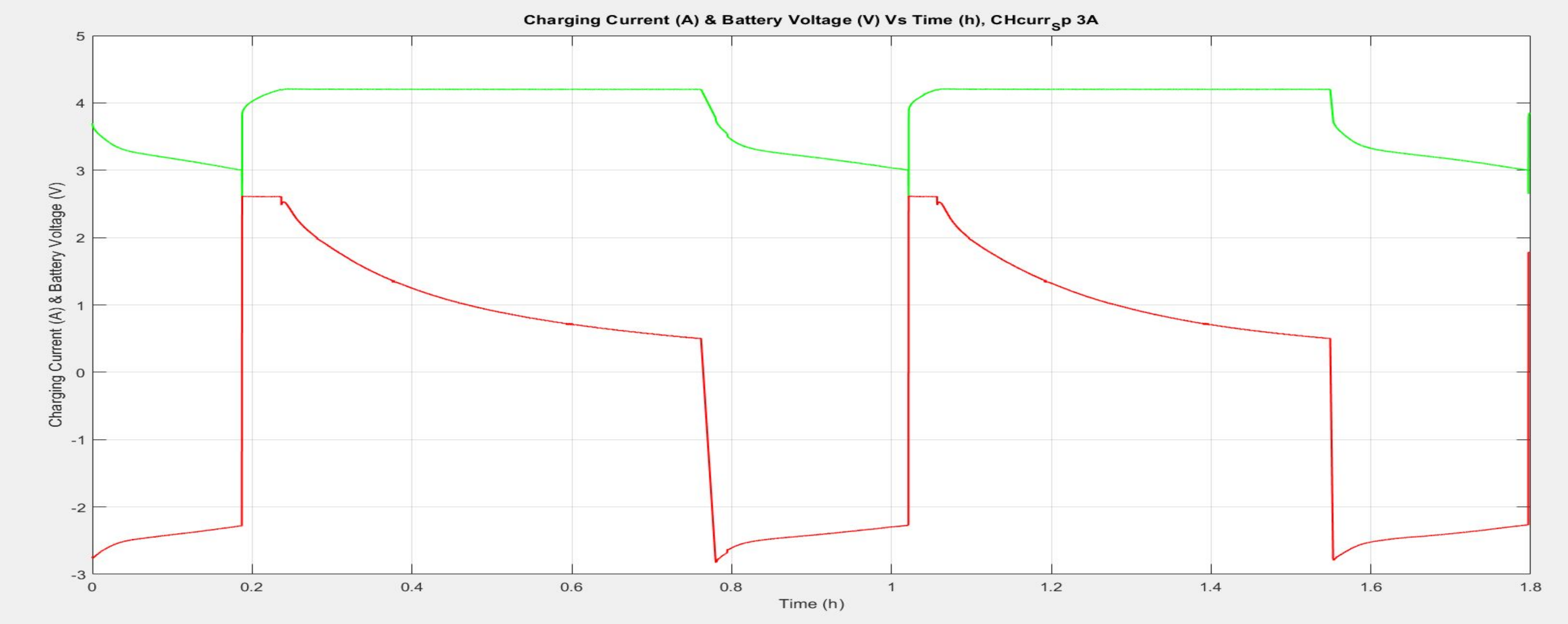
Charging and Discharging Test Results 3A

- Error: 10.66%



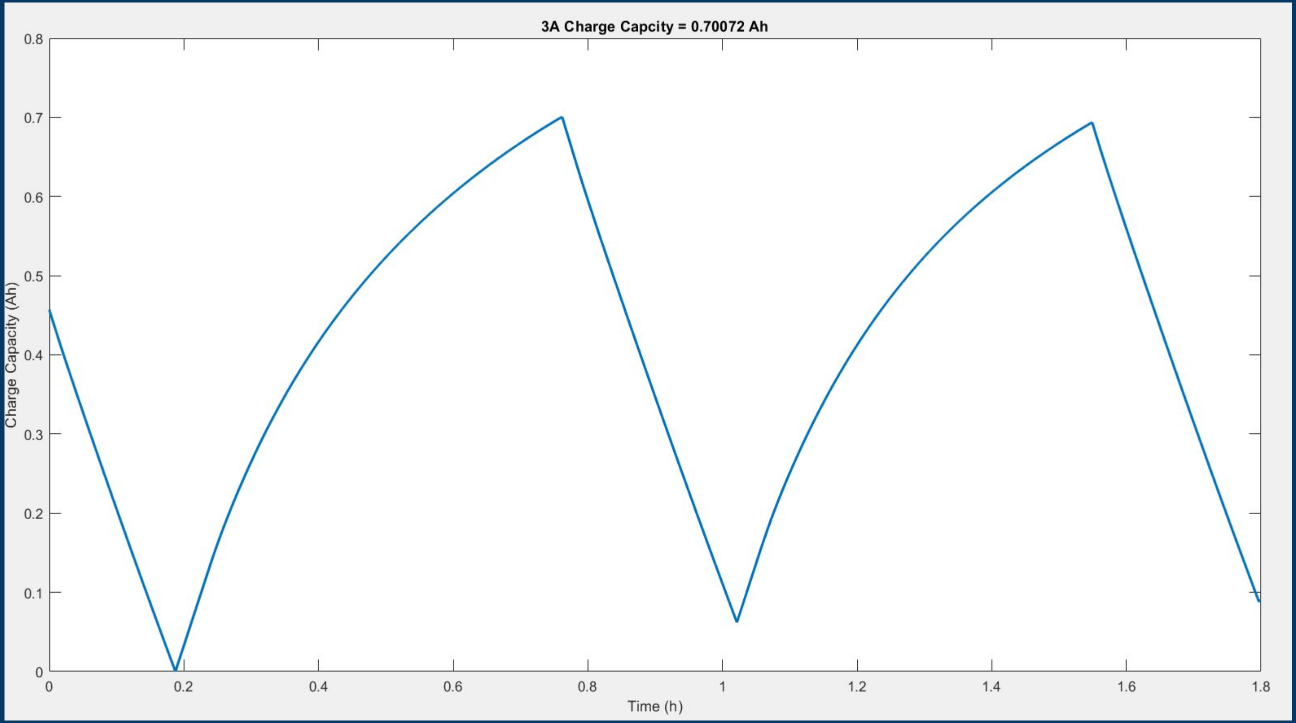
Charging Current (A) Vs Time (h)

Charging and Discharging Test Results 3A

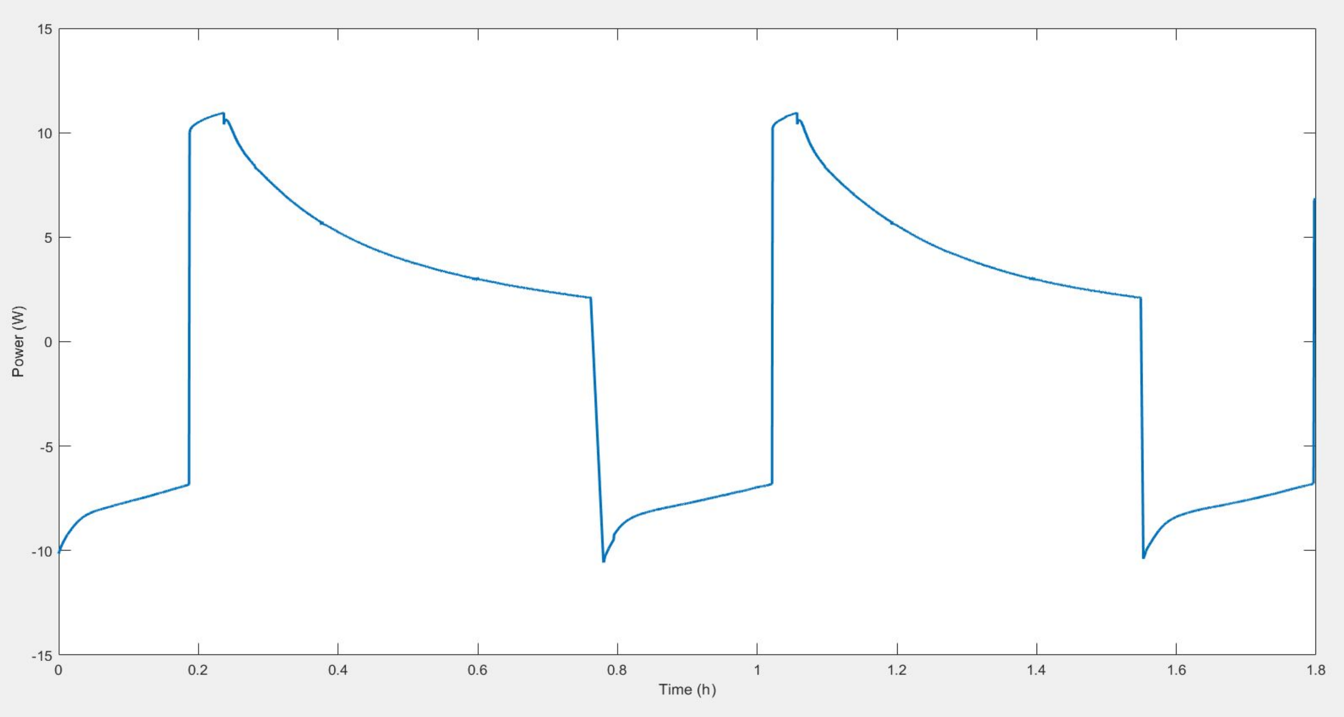


Charging Current (A) &Battery Voltage (V) Vs Time (h)

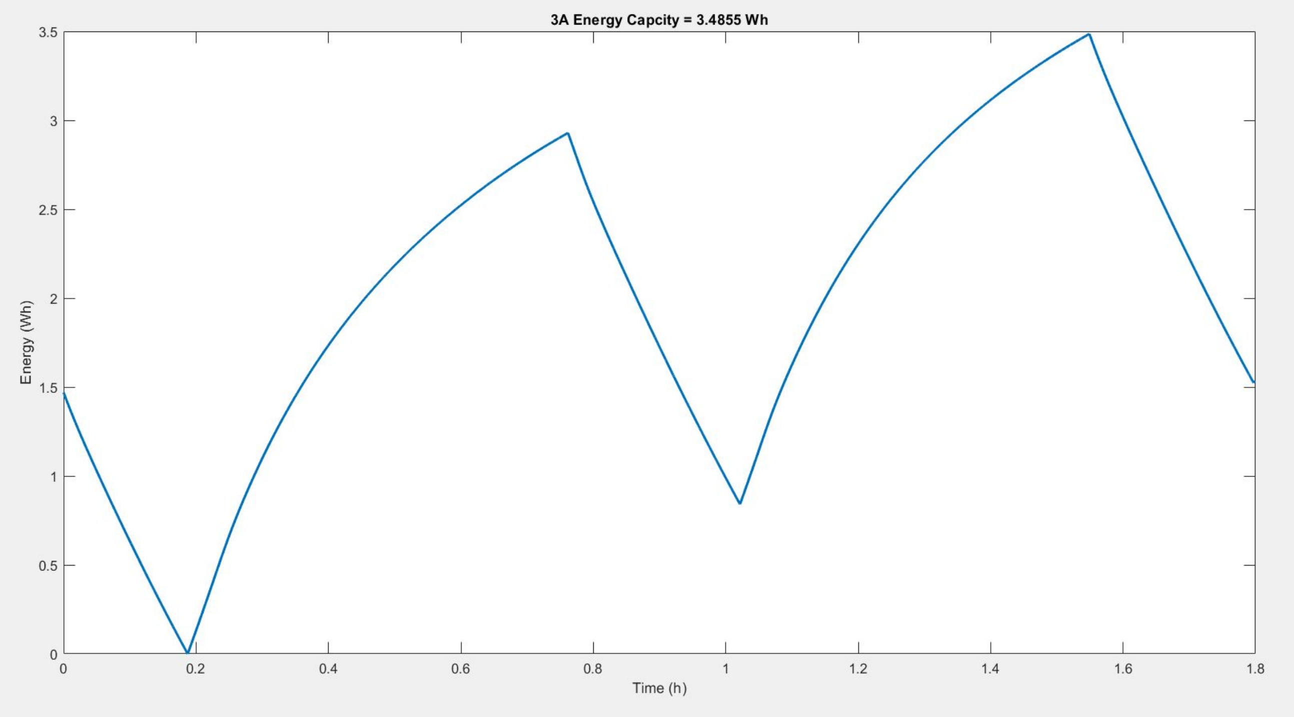
SoC at 3A Charging



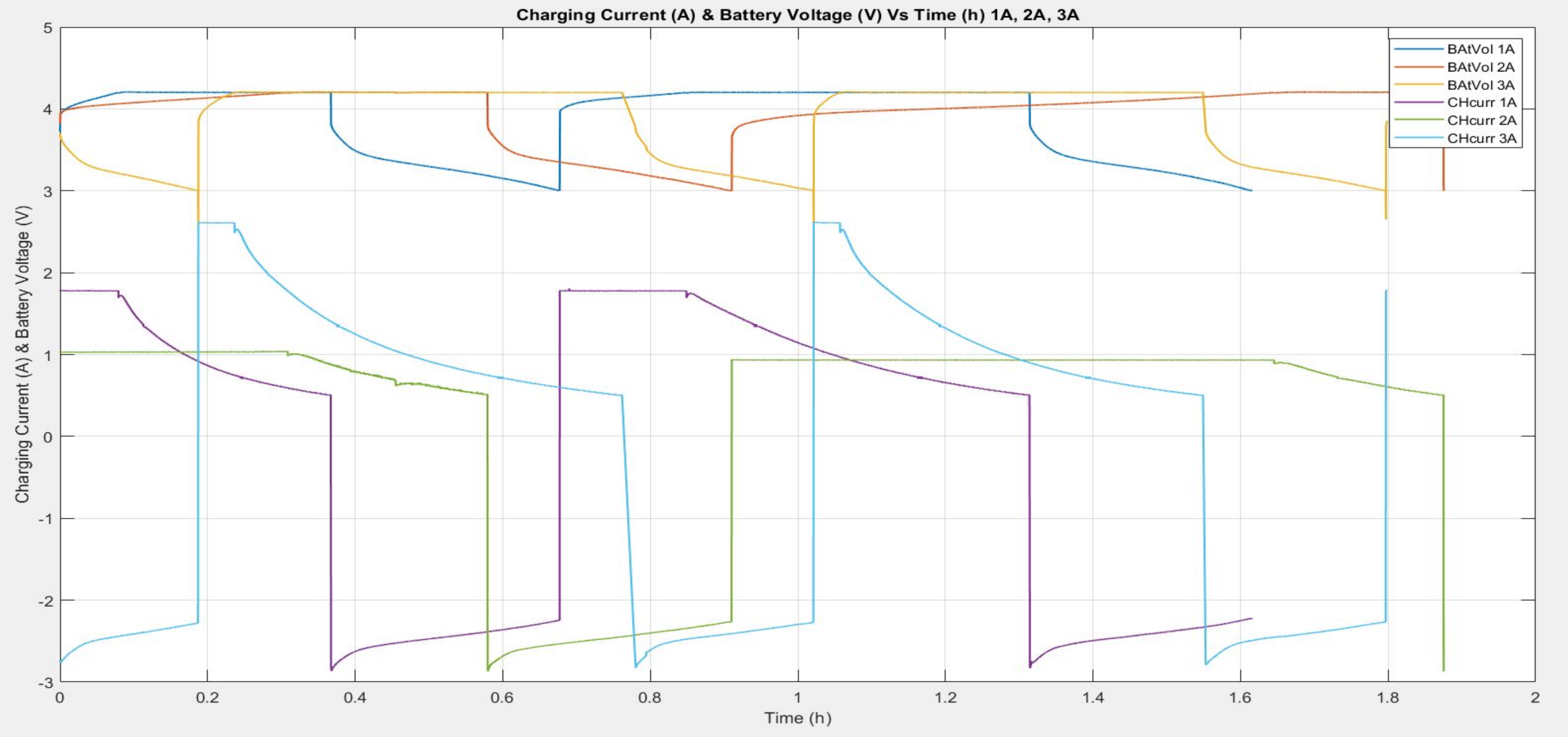
Power (W) Vs Time (h) at 3A



Energy Capacity at 3A Charging



Charging and Discharging Test Results 1A, 2A, & 3A



MATLAB Code

- SoC, Power, and Energy
  , Capacity
    - Dr. Bashash

```
clear ll; close all;
importdata('C:\Users\Owner\Downloads\3A - 3A.csv')
Time = ans.data(1:end,7);
BttVol = ans.data(1:end,3);
Chcurr = ans.data(1:end,4);
save(" t_I_V_Data3A.mat","Time","Chcurr","BttVol")

figure;
plot(Time, BttVol,LineStyle='-',Color="g",LineWidth=1);
title('Battery Voltage (V) Vs Time (h), CHcurr_Sp 3A');
xlabel('Time (h)');
ylabel('Battery Voltage (V)');
grid on;
figure;
plot(Time, Chcurr,LineStyle='-',Color="r",LineWidth=1);
title('Charging Current (A) Vs Time (h), CHcurr_Sp 3A');
xlabel('Time (h)');
ylabel('Charging Current (A)');
grid on;
figure;
plot(Time, BttVol,LineStyle='-',Color="g",LineWidth=1);
hold on;
plot(Time, Chcurr,LineStyle='-',Color="r",LineWidth=1);
title('Charging Current (A) & Battery Voltage (V) Vs Time (h), CHcurr_Sp 3A');
xlabel('Time (h)');
ylabel('Charging Current (A) & Battery Voltage (V)');
grid on;
load ' t_I_V_Data3A.mat'
%Data: t (time, hours),  V (voltage, V), I (current, A)
P = BttVol.*Chcurr;
dt = diff(Time);
dt = [dt(1); dt];  % If t is a row vector: dt = [dt(1) dt];
E = cumsum(P.*dt); % Integral of power (W-hour)
C = cumsum(Chcurr.*dt); % Charge capacity (A-hr)

figure; plot(Time,C-min(C),'linewidth',2);
xlabel('Time (h)'), ylabel('Charge Capacity (Ah)')
title(['3A Charge Capcity = ' num2str(max(C-min(C))) ' Ah'])

figure; plot(Time,P,'linewidth',2)
xlabel('Time (h)'), ylabel('Power (W)')

figure; plot(Time,E-min(E),'linewidth',2);
xlabel('Time (h)'), ylabel('Energy (Wh)')
title(['3A Energy Capcity = ' num2str(max(E-min(E))) ' Wh'])
```

Challenges and Future Improvements

- Further investigation
  - Battery used was 2 Amp hour Lib
  - The highest achieved SoC 42.5 % (0.85 Amp hour)
    - CV charging until 0.5 Amps
    - Human error
    - Battery
- Challenges
  - Data collection

Conclusion

- Primary objective was accomplished
  - proved that the Programmable CCCV LiB charger circuit work
    - Op-Amp
    - MOSFET
- A working prototype was built and tested
- Observation as charging current decrease SoC increase

Reference cited

[1] S. Meredith, “Europe is struggling with a precarious water situation ahead of another drought-driven summer,” *CNBC*, Jun. 05, 2023. [Online]. Available: https://www.cnbc.com/2023/06/05/expected-european-drought-likely-to-worsen-the-regions-water-crisis-.html. [Accessed: Jul. 01, 2023]
[3] “CVRP.” [Online]. Available: https://ww2.arb.ca.gov/sites/default/files/movingca/cvrp.html. [Accessed: Jul. 01, 2023]
[4] “Demand for electric cars is booming, with sales expected to leap 35% this year after a record-breaking 2022,” *IEA*. [Online]. Available: https://www.iea.org/news/demand-for-electric-cars-is-booming-with-sales-expected-to-leap-35-this-year-after-a-record-breaking-2022. [Accessed: Jul. 01, 2023]
[5] W. Xing, “Development of a High Energy Density, Long Cycle Life and Safety Enhanced Li-S Battery,” *Meet. Abstr.*, vol. MA2020–01, no. 2, p. 165, May 2020, doi: 10.1149/MA2020-012165mtgabs. [Online]. Available: https://iopscience.iop.org/article/10.1149/MA2020-012165mtgabs/meta. [Accessed: Jul. 01, 2023]
[6] Z. Geng, S. Wang, M. J. Lacey, D. Brandell, and T. Thiringer, “Bridging physics-based and equivalent circuit models for lithium-ion batteries,” *Electrochim. Acta*, vol. 372, p. 137829, Mar. 2021, doi: 10.1016/j.electacta.2021.137829.
[7] A. M. Colclasure, X. Li, L. Cao, D. P. Finegan, C. Yang, and K. Smith, “Significant life extension of lithium-ion batteries using compact metallic lithium reservoir with passive control,” *Electrochim. Acta*, vol. 370, p. 137777, Feb. 2021, doi: 10.1016/j.electacta.2021.137777.

Reference cited

[8] O. Borodin, X. Ren, J. Vatamanu, A. von Wald Cresce, J. Knap, and K. Xu, "Modeling Insight into Battery Electrolyte Electrochemical Stability and Interfacial Structure," *Acc. Chem. Res.*, vol. 50, no. 12, pp. 2886–2894, Dec. 2017, doi: 10.1021/acs.accounts.7b00486.

*[9] S. E. J. O'Kane *et al.*, "Lithium-ion battery degradation: how to model it," *Phys. Chem. Chem. Phys.*, vol. 24, no. 13, pp. 7909–7922, Mar. 2022, doi: 10.1039/d2cp00417h.

*[10] D. Andre, C. Appel, T. Soczka-Guth, and D. U. Sauer, "Advanced mathematical methods of SOC and SOH estimation for lithium-ion batteries," *J. Power Sources*, vol. 224, pp. 20–27, Feb. 2013, doi: 10.1016/j.jpowsour.2012.10.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775312015303

*[11] G. Dong and J. Wei, "A physics-based aging model for lithium-ion battery with coupled chemical/mechanical degradation mechanisms," *Electrochim. Acta*, vol. 395, p. 139133, Nov. 2021, doi: 10.1016/j.electacta.2021.139133.

[12] S. J. An, J. Li, C. Daniel, D. Mohanty, S. Nagpure, and D. L. Wood, "The state of understanding of the lithium-ion-battery graphite solid electrolyte interphase (SEI) and its relationship to formation cycling," *Carbon N. Y.*, vol. 105, pp. 52–76, Aug. 2016, doi: 10.1016/j.carbon.2016.04.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0008622316302676

*[13] J. Wang *et al.*, "Degradation of lithium ion batteries employing graphite negatives and nickel–cobalt–manganese oxide + spinel manganese oxide positives: Part 1, aging mechanisms and life estimation," *J. Power Sources*, vol. 269, pp. 937–948, Dec. 2014, doi: 10.1016/j.jpowsour.2014.07.030. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037877531401074X

*[14] J. Wang *et al.*, "Cycle-life model for graphite-LiFePO4 cells," *J. Power Sources*, vol. 196, no. 8, pp. 3942–3948, Apr. 2011, doi: 10.1016/j.jpowsour.2010.11.134. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775310021269

Reference cited

*[15] J. Zhu *et al.*, “Investigation of lithium-ion battery degradation mechanisms by combining differential voltage analysis and alternating current impedance,” *J. Power Sources*, vol. 448, p. 227575, Feb. 2020, doi: 10.1016/j.jpowsour.2019.227575. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037877531931568X

*[16] X. Zhu, Y. Xie, H. Chen, and W. Luan, “Numerical analysis of the cyclic mechanical damage of Li-ion battery electrode and experimental validation,” *Int. J. Fatigue*, vol. 142, p. 105915, Jan. 2021, doi: 10.1016/j.ijfatigue.2020.105915. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142112320304461

[17] F. Chinesta, N. Hascoet, and E. Cueto, “Chapter 4 - Data-driven modelling and artificial intelligence,” in *Reduced Order Models for the Biomechanics of Living Organs*, F. Chinesta, E. Cueto, Y. Payan, and J. Ohayon, Eds. Academic Press, 2023, pp. 65–94 [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323899673000068

*[18] Y. Song, D. Liu, H. Liao, and Y. Peng, “A hybrid statistical data-driven method for on-line joint state estimation of lithium-ion batteries,” *Appl. Energy*, vol. 261, p. 114408, Mar. 2020, doi: 10.1016/j.apenergy.2019.114408. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261919320951

[19] N.-D. Hoang and D. Tien Bui, “Chapter 18 - Slope Stability Evaluation Using Radial Basis Function Neural Network, Least Squares Support Vector Machines, and Extreme Learning Machine,” in *Handbook of Neural Computation*, P. Samui, S. Sekhar, and V. E. Balas, Eds. Academic Press, 2017, pp. 333–344 [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128113189000181

[20] M. A. Rosen and A. Farsi, “Chapter 3 - Electrochemical modeling of batteries,” in *Battery Technology*, M. A. Rosen and A. Farsi, Eds. Academic Press, 2023, pp. 65–90 [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780443188626000070

Reference cited

*[21] X. Cui and X. Lu, “Order reduction electrochemical mechanism model of lithium-ion battery based on variable parameters,” *Electrochim. Acta*, vol. 446, p. 142107, Apr. 2023, doi: 10.1016/j.electacta.2023.142107. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S001346862300292X

[22] S. Wang *et al.*, “Chapter 2 - Electrical equivalent circuit modeling,” in *Battery System Modeling*, S. Wang, Y. Fan, D.-I. Stroe, C. Fernandez, C. Yu, W. Cao, and Z. Chen, Eds. Elsevier, 2021, pp. 47–94 [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323904728000081

*[23] C. Ge, Y. Zheng, and Y. Yu, “State of charge estimation of lithium-ion battery based on improved forgetting factor recursive least squares-Extended Kalman Filter joint algorithm,” *Journal of Energy Storage*, vol. 55, p. 105474, Nov. 2022, doi: 10.1016/j.est.2022.105474. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X22014669

[24] “Charging your Tesla,” Nov. 18, 2014. [Online]. Available: https://www.tesla.com/support/charging. [Accessed: Jul. 05, 2023]

*[25] T. R. Tanim, M. G. Shirk, R. L. Bewley, E. J. Dufek, and B. Y. Liaw, “Fast charge implications: Pack and cell analysis and comparison,” *J. Power Sources*, vol. 381, pp. 56–65, Mar. 2018, doi: 10.1016/j.jpowsour.2018.01.091. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775318301009

*[26] W. Mai, A. M. Colclasure, and K. Smith, “Model-Instructed Design of Novel Charging Protocols for the Extreme Fast Charging of Lithium-Ion Batteries Without Lithium Plating,” *J. Electrochem. Soc.*, vol. 167, no. 8, p. 080517, May 2020, doi: 10.1149/1945-7111/ab8c84. [Online]. Available: https://iopscience.iop.org/article/10.1149/1945-7111/ab8c84/meta. [Accessed: Jun. 17, 2023]

*[27] C. Zhang, J. Jiang, Y. Gao, W. Zhang, Q. Liu, and X. Hu, “Charging optimization in lithium-ion batteries based on temperature rise and charge time,” *Appl. Energy*, vol. 194, pp. 569–577, May 2017, doi: 10.1016/j.apenergy.2016.10.059. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261916315033

Reference cited

*[28] M. Abdel Monem *et al.*, "Lithium-ion batteries: Evaluation study of different charging methodologies based on aging process," *Appl. Energy*, vol. 152, pp. 143–155, Aug. 2015, doi: 10.1016/j.apenergy.2015.02.064. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261915002494

*[29] S. Bashash, "Recursive Least Squares Estimation of Battery Charge Capacity and State of Charge." *Proc. of the Interdisciplinary Conference on Mechanics, Computers and Electrics* (ICMECE 2022) 6-7 October 2022, Barcelona, Spain. Available: Not available very recent paper

[30] The author generated this text in part with CGPT, OpenAI's large-scale language-generation model for fixing grammar, spelling, and punctuation. Upon generating draft language, the author reviewed, edited, and revised the language to their own liking and takes ultimate responsibility for the content of this publication.

[31] Uddin, Kotub & Picarelli, Alessandro & Lyness, Christopher & Taylor, Nigel & Marco, James. (2014). An Acausal Li-Ion Battery Pack Model for Automotive Applications. Energies. 7. 5675-5700. 10.3390/en7095675.

Thank you for your Time.

Dr. Bashash

Dr. Du

Dr. Jiang