SINGAPORE POLYTECHNIC
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

Digital Twin Application

for battery energy storage / server motor & its driver system

SUBMITTED BY: Abirami Baskaran (P2241801)

Project Supervisor: Dr Fred Wei, Dr Cai Zhi Qiang

February 2024

Acknowledgements

I would like to convey my appreciation to Dr Fred Wei and Dr Cai Zhi Qiang for assigning me into this internship project and overseeing and facilitating my internship experience. Dr Fred's unwavering support and mentorship have been vital to our group, and I am truly grateful for his expertise.

Furthermore, I would like to express my gratitude to Singapore Polytechnic for their instrumental role in arranging this internship. The commitment to foster practical learning environments is evident, and I appreciate the effort and coordination involved in providing me with this valuable experience.

This acknowledgement is a testament to the collaborative efforts of Dr Fred, Dr Cai and Singapore Polytechnic in ensuring the success of my internship. I am thankful for the support and the conducive learning environment that have consistently provided.

Abstract

This report delves into my immersive experience with the Digital Twinning project, funded by mVizn Pte Ltd within the academic setting of Singapore Polytechnic. My primary objective of this internship was to contribute to the development of a self-learning model for Lithium-ion batteries, funded by TIEFA, and servo motors, funded by Alpha. Tasks included the study of degradation and the implementation of fault prediction algorithms for both battery and servo motor systems.

Throughout the internship, I achieved significant milestones, notably gaining exposure through competition, participating in an exhibit showcase, and successfully completing intricate tasks. The report concludes with reflections on lessons learned and the profound impact of this internship-equivalent project on my professional growth within an academic environment.

Table of Contents

1. Introduction

Digital Twinning technology is a new growing technology gaining widespread adoption across various industries. A digital twin simply refers to a virtual replicate of a physical object. It spans the object's lifecycle and uses real-time data sent from embedded sensors within the physical object to accurately simulate the behavior. This approach enables technicians and industries to effortlessly monitor the object's performance without a need for extensive hands-on involvement.

In this project, the group aims to develop a digital twin for a lithium-ion battery and a servo motor. The application and reason for applying this technology for both objects is described below.

1.1. Lithium-ion (Li-ion) Battery

Over the years, there have been reports of electric vehicles (EV) catching on fire, raising concerns about Li-ion batteries and the fire risks they could present. Causes of fire include improper charging, physical damaging, high temperature and any faults within the battery. For most EVs, the fire starts in battery powered system. Though the moment of fire happens suddenly, the fault in the battery could have been detected earlier and if caught at the right time, could prevent any potential damage.

To prevent the risk of fire, a digital twin of the battery needs to be created to have a monitoring of the battery. With monitoring all the attached sensors within the battery, a battery management system (BMS) can be created, and the data readings could be used to develop a neural network to simulate a twin.

1.2. Servo Motor

In a motor, the values within fluctuates especially the resistors and inductors within it. It is crucial to have a digital model to predict the fluctuating values as it assists in gaining insights into the motor's behavior and optimizes its purpose. With a digital twin, there is real time monitoring of the inductor motor's electrical parameters, such as resistors, inductors, and capacitors. This allows predictive analysis of potential variations and engineers could identify trends and deviations, maintaining predictive maintenance strategies. In addition, a digital twin of a motor could also serve as a valuable tool for refining the motor's design, ensuring optimal operation and efficiency. This also contributes to improved performance, reduced downtime, and enhances operational efficiency.

2. Project Overview

To construct a digital twin for both a battery and a motor, a full set of processes is required. This comprehensive approach includes user testing, thorough data collecting, the creation of a user-friendly dashboard, and the deployment of predictive error modeling. The next portion of this section goes further into the complex details of these critical components, offering a detailed knowledge of the techniques and methodologies used.

2.1. My Technical Experience & Contributions to the project

The following details are stated in chronological order of my work.

2.1.1. Working on schematic model

On the first day of internship, 11 September, I was given my role for the group. My tasks were to develop a self-learning model and degradation study of the projects the team will be working on. i.e. Lithium-ion battery and Servo Motor.

I was introduced to 'PLECS' which is a software tool for system-level simulations of electrical circuits. The software is developed by Plexim and is designed for power electronics and electrical networks, which comes as a highly useful tool for the project.

plecs

Figure 1: Plecs Software

At the earlier stage, I faced difficulties understanding and adapting to the software as the theoretical knowledge of electronic networks were new to me since I was a computing student. However, with the seamless user interface of plecs, I started practicing and learnt the basics of schematic software tools and circuit theory.

After familiarizing myself with the tool, I studied the given circuit of lithium-ion battery to understand its behavior and possible faults to occur. This stage required a lot of testing to thoroughly understand the circuit and how the attached sensors are affected when there is a fault. I added more sensors to understand how the voltage drops and current flow changes and acts across the circuit.

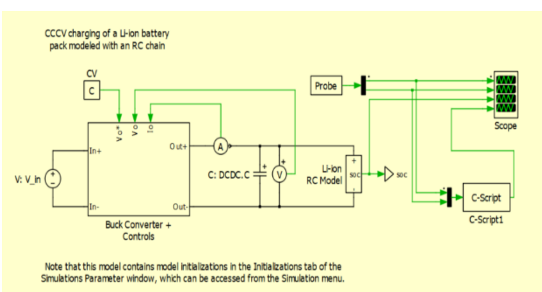


Figure 2: Original Circuit of battery (version 1)

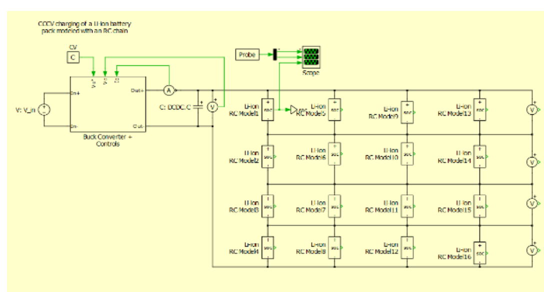


Figure 3: Final version of model

I modified the final version of schematic model to further improve the battery features. Originally, it was a 4x4 series, parallel circuit where each cells were directly connected to each other. There were 4 voltmeters that were connected parallel to the battery cells *(Refer to Figure 3)*.

Though this schematic was designed according to the physical purchased battery, I thought of improving on its features for a better functionality.

<u>Changes Done</u>

a) I inserted resistors serially between each battery cell. After several trials, I finalized the resistor value to be 0.5 for a more noticeable difference to detect during effect.
b) 4 Ammeters are connected serially to each column of cells. *(Refer to figure 4)*

<u>Reason for Change</u>

a) When the battery cells are connected directly with each other, if one individual cell gets affected for example 'short circuit', it possesses a risk of the entire row to be short circuit requiring replacement of entire row. However, with the addition of resistors in between, there is a voltage drop which protects other cells.
b) With only 4 Voltmeters connected parallel, location of fault couldn't be precise. The row of faulty battery cells can be detected; however, the specific cell would be unknown. Hence with an addition of ammeters, the column of faulty battery cell could be detected. Knowing the row and the column, it would be easier to identify the specific location of the battery at fault.

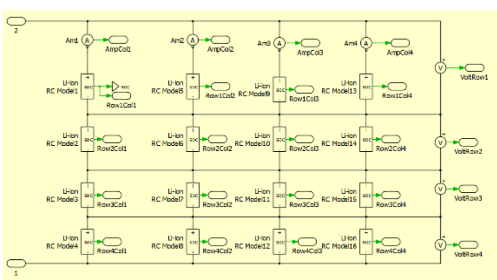


Figure 4: Modified model

```
For file Resist_0.5R/Row1Col2.csv error is at Row1 and Col2
----------------- Amp difference:9.712915451 Volt Difference:3.422659963
For file Resist_0.5R/Row1Col3.csv error is at Row1 and Col3
----------------- Amp difference:9.575032682 Volt Difference:0.593856427
For file Resist_0.5R/Row1Col4.csv error is at Row1 and Col4
----------------- Amp difference:5.1704470030000005 Volt Difference:0.0
For file Resist_0.5R/Row2Col1.csv error is at Row2 and Col1
----------------- Amp difference:5.5865091730000005 Volt Difference:3.594870136
For file Resist_0.5R/Row2Col2.csv error is at Row2 and Col2
----------------- Amp difference:5.739504775 Volt Difference:3.283878563
For file Resist_0.5R/Row2Col3.csv error is at Row2 and Col3
----------------- Amp difference:5.569879895 Volt Difference:0.85618756
```

Figure 5: Output of predicted errors

In Figure 5, the green underlined boxes are the name of data files, and the red underlined boxes are the detected output based on a built algorithm.

After simulating errors in the circuit, the data is extracted and saved into a Comma Separated Values (csv) file. The csv data file is annotated accordingly and is tested. Looking at figure 5, almost all the output values are correctly predicted.

Methodology

During the moment of short circuit, there will be a sudden rise in the affected column of ammeter while the remaining ammeters drops. Meanwhile, the affected row of voltmeter has a drop while the other voltmeter rises slightly. Hence the program searches for the change and outputs the affected row and column.

2936.984	0.504094	0.332804	0.098045	0.044084	0.029161	16.4	3.808676	3.812998	3.81592	3.817369
2956.984	0.504072	0.332763	0.09805	0.044091	0.029168	16.4	3.808679	3.813004	3.815928	3.817378
2976.984	0.504049	0.332721	0.098055	0.044098	0.029175	16.4	3.808683	3.81301	3.815935	3.817386
2996.984	0.504027	0.33268	0.098061	0.044105	0.029181	16.4	3.808686	3.813016	3.815942	3.817394
3000	0.504024	0.332674	0.098061	0.044106	0.029182	16.4	3.808686	3.813017	3.815943	3.817395
3000	0.504024	-0.1072	-0.74096	-0.85964	2.383774	16.4	3.986599	4.206048	4.509862	0
3000.637	0.706003	-0.10654	-0.73506	-0.84064	2.36073	16.39965	3.987302	4.207997	4.514446	0
3001.275	0.743078	-0.10569	-0.72924	-0.82217	2.338286	16.39979	3.987984	4.209895	4.518926	0
3001.912	0.697632	-0.10478	-0.72353	-0.80424	2.316425	16.40008	3.988647	4.211745	4.523302	0

Figure 6: Spreadsheet data of affected Row 4 Column 4 Cell

Problems Faced

Though the modified circuit allowed me to detect the specific location of the affected cell, the new schematic arose some faults.

With the new addition of resistors, fault, or changes in the further end of the sensors were hard to notice.

For example, faults at column 1 were harder to notice by the voltmeters compared to faults at column 3 and 4. This occurred because the voltmeters were connected further end towards right, and columns 3 and 4 were closer. Hence during effect, if the fault is at column 3 or 4, there is a visible drop at the voltmeter. However, if there is a fault in column 1, there is not much visible drop, nor a difference detected at the affected row of voltmeter compared to other voltmeters. This is because the impedance of a battery cell is low and due to the addition of resistors the current flowing through the resistors prevents it from having high change of voltage at the right side of circuit.

In addition, I later understood that the changes would not be practical to be applied as the physical battery which was manufactured, it is to make most with minimum sensors and is also not advised to add in additional components. Furthermore, with regards to the resistors, it is advised to exclude not just the affected, but all the battery cells that are nearer and wielded to it. Hence, I revert back to the original circuit which was accurate to the battery and worked with on it further.

2.1.2. Creation of Algorithm

Creating a neural network for the battery with only being limited to 4 voltmeters was a struggle at first. With more sensors, more information about the battery could be retrieved. However, since it is not practical modify the physical circuit, I adapted with the available sensors.

Data Collection and Analyzing

I meticulously collected extensive and diverse data, encompassing various scenarios such as "typical battery operation", "varied fault types", and obtained data from different spatial locations within the battery system. Later, they were labelled so could be analyzed respectively.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Vm2:Me	Vm3:Me	Vm4:Me	Vm5:Me	Results	Vm2:Me	Vm3:Me	Vm4:Me	Vm5:Me	Results	Vm2:Me	Vm3:Me	Vm4:Me	Vm5:Me	Results	Vm2:Me	Vm3:Me	Vm4:Me	Vm5:Me	Results	Vm2:Me	Vm3:Me	Vm4:Me	Vm5:Me	Results
2	0.0394	0.0394	0.0394	0.0394	No Fault	0	4.3112	4.3112	4.3112	Fault 1	4.3112	4.3112	0	4.3112	Fault3	4.0419	4.0419	4.0419	0	Fault 4	4.0545	0	4.0545	4.0545	Fault 2
3	0.0785	0.0785	0.0785	0.0785	No Fault	0	4.0419	4.0419	4.0419	Fault 2	4.0419	4.0419	0	4.0419	Fault4	4.0419	4.0419	4.0419	0	Fault 5	4.0546	0	4.0546	4.0546	Fault 3
4	0.1171	0.1171	0.1171	0.1171	No Fault	0	4.0418	4.0418	4.0418	Fault 3	4.0418	4.0418	0	4.0418	Fault5	4.0419	4.0419	4.0419	0	Fault 6	4.0547	0	4.0547	4.0547	Fault 4
5	0.1553	0.1553	0.1553	0.1553	No Fault	0	4.0418	4.0418	4.0418	Fault 4	4.0418	4.0418	0	4.0418	Fault6	4.0419	4.0419	4.0419	0	Fault 7	4.0548	0	4.0548	4.0548	Fault 5
6	0.1931	0.1931	0.1931	0.1931	No Fault	0	4.0418	4.0418	4.0418	Fault 5	4.0418	4.0418	0	4.0418	Fault7	4.0419	4.0419	4.0419	0	Fault 8	4.0549	0	4.0549	4.0549	Fault 6
7	0.2309	0.2309	0.2309	0.2309	No Fault	0	4.0418	4.0418	4.0418	Fault 6	4.0418	4.0418	0	4.0418	Fault8	4.0419	4.0419	4.0419	0	Fault 9	4.0549	0	4.0549	4.0549	Fault 7
8	0.2683	0.2683	0.2683	0.2683	No Fault	0	4.0418	4.0418	4.0418	Fault 7	4.0418	4.0418	0	4.0418	Fault9	4.0419	4.0419	4.0419	0	Fault 10	4.0549	0	4.0549	4.0549	Fault 8
9	0.3053	0.3053	0.3053	0.3053	No Fault	0	4.0418	4.0418	4.0418	Fault 8	4.0418	4.0418	0	4.0418	Fault10	4.0419	4.0419	4.0419	0	Fault 11	4.0549	0	4.0549	4.0549	Fault 9
10	0.3419	0.3419	0.3419	0.3419	No Fault	0	4.0418	4.0418	4.0418	Fault 9	4.0418	4.0418	0	4.0418	Fault11	4.0419	4.0419	4.0419	0	Fault 12	4.0549	0	4.0549	4.0549	Fault 10
11	0.3781	0.3781	0.3781	0.3781	No Fault	0	4.0418	4.0418	4.0418	Fault 10	4.0418	4.0418	0	4.0418	Fault12	4.0419	4.0419	4.0419	0	Fault 13	4.0549	0	4.0549	4.0549	Fault 11
12	0.6135	0.6135	0.6135	0.6135	No Fault	0	4.0418	4.0418	4.0418	Fault 11	4.0418	4.0418	0	4.0418	Fault13	4.0419	4.0419	4.0419	0	Fault 14	4.0549	0	4.0549	4.0549	Fault 12
13	0.6432	0.6432	0.6432	0.6432	No Fault	0	4.0418	4.0418	4.0418	Fault 12	4.0418	4.0418	0	4.0418	Fault14	4.0419	4.0419	4.0419	0	Fault 15	4.0549	0	4.0549	4.0549	Fault 13
14	0.6432	0.6432	0.6432	0.6432	No Fault	0	4.0418	4.0418	4.0418	Fault 13	4.0418	4.0418	0	4.0418	Fault15	4.042	4.042	4.042	0	Fault 16	4.0549	0	4.0549	4.0549	Fault 14
15	0.9182	0.9182	0.9182	0.9182	No Fault	0	4.0418	4.0418	4.0418	Fault 14	4.0418	4.0418	0	4.0418	Fault16	4.042	4.042	4.042	0	Fault 17	4.0549	0	4.0549	4.0549	Fault 15
16	1.1695	1.1695	1.1695	1.1695	No Fault	0	4.0418	4.0418	4.0418	Fault 15	4.0418	4.0418	0	4.0418	Fault17	4.042	4.042	4.042	0	Fault 18	4.0549	0	4.0549	4.0549	Fault 16
17	1.3989	1.3989	1.3989	1.3989	No Fault	0	4.0418	4.0418	4.0418	Fault 16	4.0418	4.0418	0	4.0418	Fault18	4.042	4.042	4.042	0	Fault 19	4.0549	0	4.0549	4.0549	Fault 17
18	1.6083	1.6083	1.6083	1.6083	No Fault	0	4.0418	4.0418	4.0418	Fault 17	4.0418	4.0418	0	4.0418	Fault19	4.042	4.042	4.042	0	Fault 20	4.0549	0	4.0549	4.0549	Fault 18
19	1.7995	1.7995	1.7995	1.7995	No Fault	0	4.0418	4.0418	4.0418	Fault 18	4.0418	4.0418	0	4.0418	Fault20	4.042	4.042	4.042	0	Fault 21	4.0549	0	4.0549	4.0549	Fault 19
20	1.9918	1.9918	1.9918	1.9918	No Fault	0	4.0418	4.0418	4.0418	Fault 19	4.0418	4.0418	0	4.0418	Fault21	4.0421	4.0421	4.0421	0	Fault 22	4.0549	0	4.0549	4.0549	Fault 20
21	2.166	2.166	2.166	2.166	No Fault	0	4.0418	4.0418	4.0418	Fault 20	4.0418	4.0418	0	4.0418	Fault22	4.0421	4.0421	4.0421	0	Fault 23	4.0549	0	4.0549	4.0549	Fault 21
22	2.3234	2.3234	2.3234	2.3234	No Fault	0	4.0418	4.0418	4.0418	Fault 21	4.0418	4.0418	0	4.0418	Fault23	4.0421	4.0421	4.0421	0	Fault 24	4.055	0	4.055	4.055	Fault 22
23	2.4657	2.4657	2.4657	2.4657	No Fault	0	4.0418	4.0418	4.0418	Fault 22	4.0418	4.0418	0	4.0418	Fault24	4.0421	4.0421	4.0421	0	Fault 25	4.055	0	4.055	4.055	Fault 23
24	2.5944	2.5944	2.5944	2.5944	No Fault	0	4.0418	4.0418	4.0418	Fault 23	4.0418	4.0418	0	4.0418	Fault25	4.0421	4.0421	4.0421	0	Fault 26	4.0551	0	4.0551	4.0551	Fault 24
25	2.7722	2.7722	2.7722	2.7722	No Fault	0	4.0418	4.0418	4.0418	Fault 24	4.0418	4.0418	0	4.0418	Fault26	4.0422	4.0422	4.0422	0	Fault 27	4.0551	0	4.0551	4.0551	Fault 25
26	2.9246	2.9246	2.9246	2.9246	No Fault	0	4.0418	4.0418	4.0418	Fault 25	4.0418	4.0418	0	4.0418	Fault27	4.0422	4.0422	4.0422	0	Fault 28	4.0552	0	4.0552	4.0552	Fault 26
27	3.0547	3.0547	3.0547	3.0547	No Fault	0	4.0418	4.0418	4.0418	Fault 26	4.0418	4.0418	0	4.0418	Fault28	4.0422	4.0422	4.0422	0	Fault 29	4.0553	0	4.0553	4.0553	Fault 27
28	3.1655	3.1655	3.1655	3.1655	No Fault	0	4.0418	4.0418	4.0418	Fault 27	4.0418	4.0418	0	4.0418	Fault29	4.0423	4.0423	4.0423	0	Fault 30	4.0554	0	4.0554	4.0554	Fault 28
29	3.2602	3.2602	3.2602	3.2602	No Fault	0	4.0418	4.0418	4.0418	Fault 28	4.0418	4.0418	0	4.0418	Fault30	4.0423	4.0423	4.0423	0	Fault 31	4.0554	0	4.0554	4.0554	Fault 29
30	3.3639	3.3639	3.3639	3.3639	No Fault	0	4.0418	4.0418	4.0418	Fault 29	4.0418	4.0418	0	4.0418	Fault31	4.0423	4.0423	4.0423	0	Fault 32	4.0555	0	4.0555	4.0555	Fault 30
31	3.449	3.449	3.449	3.449	No Fault	0	4.0418	4.0418	4.0418	Fault 30	4.0418	4.0418	0	4.0418	Fault32	4.0424	4.0424	4.0424	0	Fault 33	4.0556	0	4.0556	4.0556	Fault 31

Figure 7: Collection of labelled datasets

Once all data was collected, I employed a profound understanding of the data and worked on identifying critical patterns. Using differential analysis and correlation analysis, I examined the relationship between various parameters and the output labels. The correlation analysis utilized statistical techniques to quantify the degree to which the input values and the output label were associated. The differential analysis observed how variations in one parameter relate to changes in others. Through this, I had a clear understanding of the circuit theory of Lithium-ion batteries. To achieve the goal of detecting faults, I wanted to approach with 2 different methods. Creating a Neural Network and Create a rule-based algorithm.

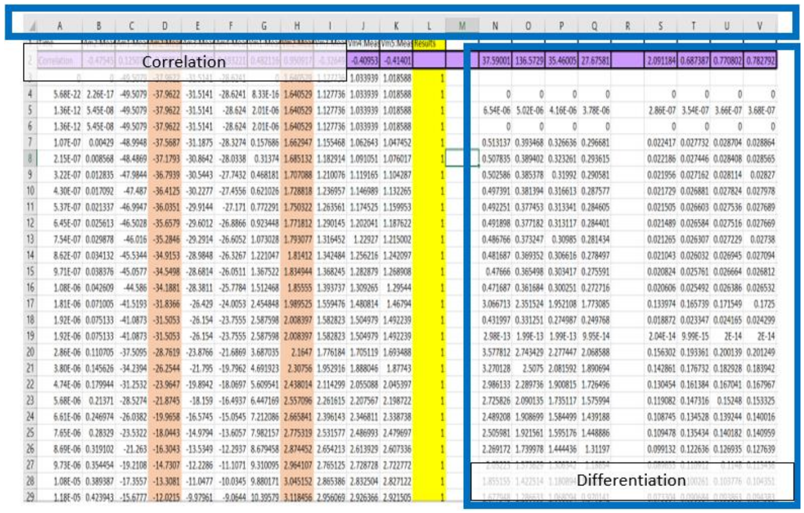


Figure 8: Differentiation and Correlation on dataset

Attempt on Neural Network

Using the individual voltmeter values as the input, I collected a large set of csv data and labelled the outputs as 0, 1, 2, 3, 4 which represents ‘No Error’, ‘Error at cell 1’, ‘Error at cell 2’, ‘Error at cell 3’, ‘Error at cell 4’ respectively.

The resulting technological complexities were resolved through the use of sophisticated Python packages like as pandas, torch, and sklearn. This permitted the creation of a complex neural network model capable of real-time defect prediction using the current dataset. Despite using a linear regression model as the underlying architecture, reaching ideal accuracy proved to be a difficult task, given the initial differences in predictions.

The final trained neural network model had above 80% accuracy where when a new dataset is tested, it predicted the output of 798 datasets correctly out of the 928 datasets. Figure 9 shows the output of the test. From figure 9, though the accuracy is decently high, it is <u>not</u> recommended to be used in real life applications as it is necessary for the prediction to be accurate and a need to not worry of mistake when a client refers to the program.

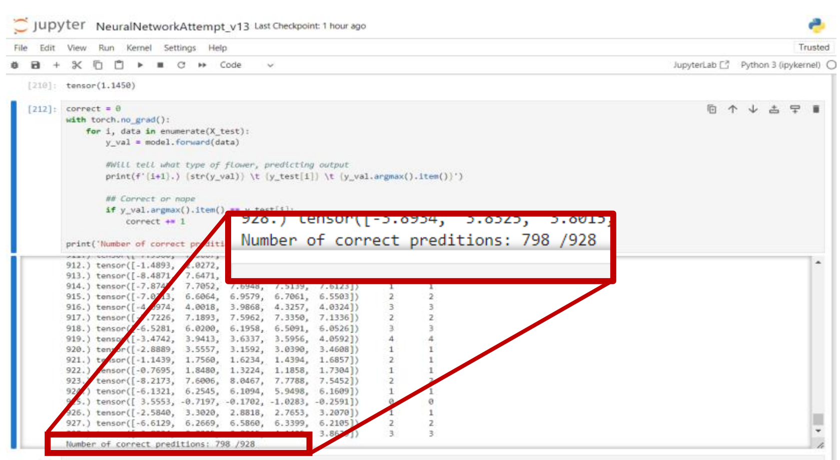


Figure 9: Test run of neural network.

Traditional Statistical Algorithm

A traditional statistical algorithm refers to using mathematical formulas on the context of machine learning or statistical modelling.

I meticulously formulated an algorithm based on the discerned pattern within the dataset compilation. The algorithm adapts and processes any incoming unlabeled data, enabling the prediction of the battery's status, alongside the identification of fault type and its precise location within the battery. This approach is called a "Traditional Statistical Algorithm". After getting a clear understanding of the cause-and-effect relationships within the battery system, an algorithm is created so that the probability of false prediction is highly unlikely.

I made equations to detect short circuits, open circuits and overcharging in all the locations. I later set it as a formula, and made it detect from reading live data.

With so far tests, the algorithm has not made much false predictions and is safe to say it's accuracy is almost 100%.

Figure 10: Differentiating affected and non affected voltmeters

Time Series Prediction

To create the digital twin, I need a model that acts exactly accurate to physical battery on a normal working condition. Through research, I realized, I needed to build a Long Short-Term Memory (LSTM) Model.

- ***About LSTM***
  LSTM stands for Long Short-Term Memory, which is a type of Recurrent Neural Network. This model is well suited for handling time-series data as it captures long-term dependencies.

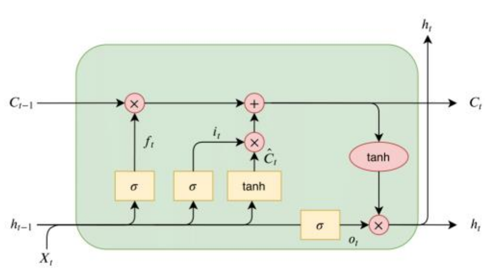


Figure 11: LSTM Model visualization

- ***Data Preparation***

  In the realm of predictive modelling, the focus is primarily on data representing normal battery operation of a working battery. This data is meticulously preprocessed to suit the requirements of LSTM networks.

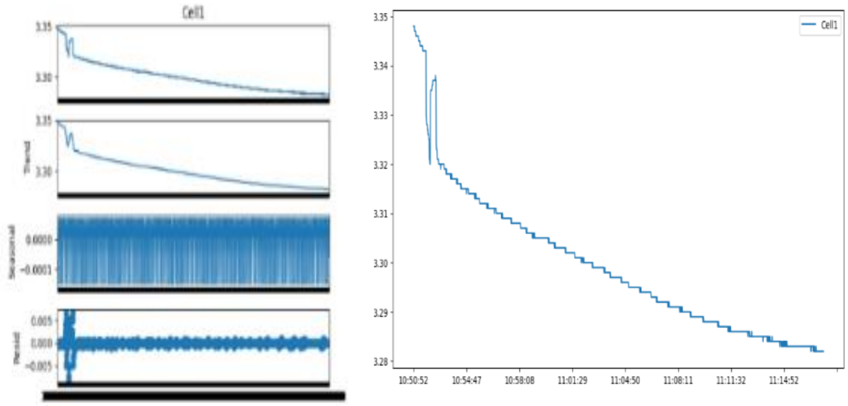


Figure 12: Training Data

The training data is input as ***n*** batches, and it is trained to predict the next value (***n + 1***). The number ***n*** is decided after several runs to get better accuracy.
for eg ***n*** = 3:

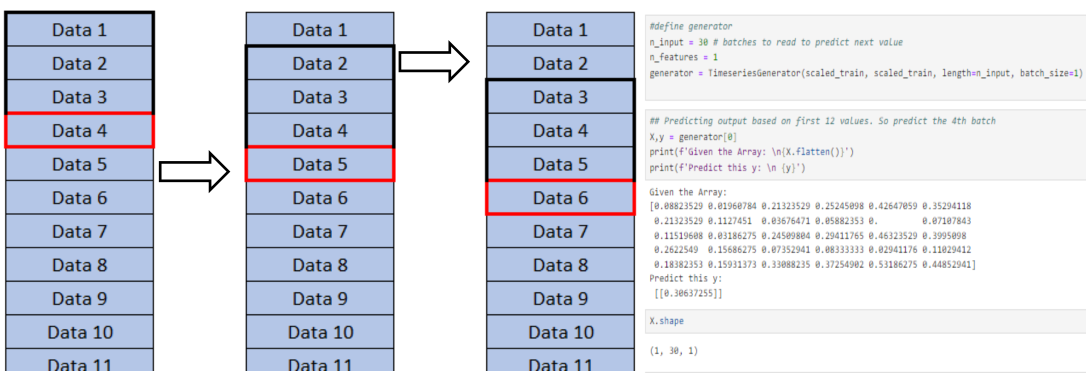


Figure 13: How data is being read

- ***Training and Optimization***

  The LSTM model is trained using the preprocessed dataset, with a specific emphasis on its capacity to predict future values of a healthy battery. The predictive accuracy of this model is enhanced by tuning in the hyperparameters, mainly the batch size and the learning rate.

```
Epoch 35/50
141/141 [==============================] - 4s 26ms/step - loss: 0.0045
Epoch 36/50
141/141 [==============================] - 4s 25ms/step - loss: 0.0033
Epoch 37/50
141/141 [==============================] - 4s 25ms/step - loss: 0.0037
Epoch 38/50
141/141 [==============================] - 4s 26ms/step - loss: 0.0043
Epoch 39/50
141/141 [==============================] - 4s 27ms/step - loss: 0.0030
Epoch 40/50
141/141 [==============================] - 4s 25ms/step - loss: 0.0038
Epoch 41/50
141/141 [==============================] - 4s 26ms/step - loss: 0.0025
Epoch 42/50
141/141 [==============================] - 4s 26ms/step - loss: 0.0028
Epoch 43/50
141/141 [==============================] - 4s 27ms/step - loss: 0.0035
Epoch 44/50
141/141 [==============================] - 4s 26ms/step - loss: 0.0027
Epoch 45/50
141/141 [==============================] - 4s 28ms/step - loss: 0.0046
Epoch 46/50
141/141 [==============================] - 4s 28ms/step - loss: 0.0023
Epoch 47/50
141/141 [==============================] - 4s 28ms/step - loss: 0.0026
Epoch 48/50
141/141 [==============================] - 4s 29ms/step - loss: 0.0024
Epoch 49/50
141/141 [==============================] - 4s 27ms/step - loss: 0.0024
Epoch 50/50
141/141 [==============================] - 4s 29ms/step - loss: 0.0020
<keras.src.callbacks.History at 0x254274fb550>
```

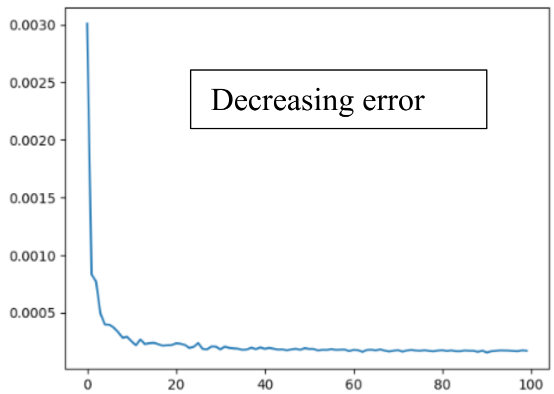


Figure 14: Running of epochs and error decreasing rate

- ***Evaluation and Validation***

  The performance of the LSTM model is evaluated and compared with the original dataset for comparison.

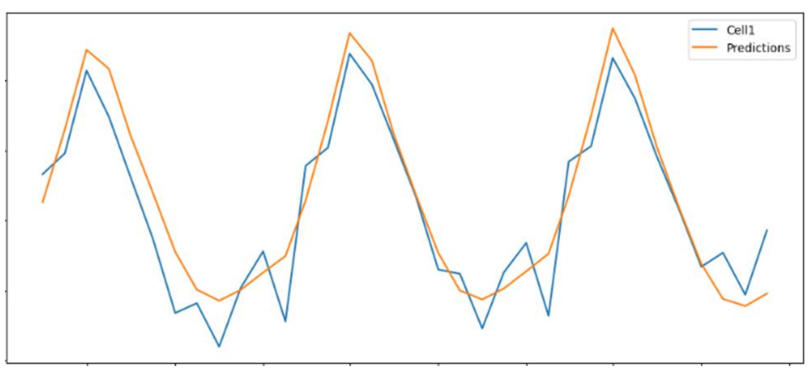


Figure 15: Comparison of predicted values and actual values

2.1.3. Integrate to Dashboard

After creating a good model, a teammate, Lau Jun Zhe passed me his program to upload data to dashboard. Jun Zhe created a program where it displays live data to the dashboard. I worked on combining both our programs and modified the dashboard so it could display the output of my model.

There were a handful of milestones faced in this process as I had to thoroughly understand the process on how my teammate created his dashboard so that I could make my desired changes correctly and neatly. This process took around a week, due to errors with the server that was used and format compatibility.

About used Dashboard

The dashboard created in this project was built with Grafana, which is a multi-platform open-source analytics and interactive visualization web applications. The web server provides charts and graphs which allowed us to create an eye appealing dashboard.

Originally, the team used JBD tools to read values from the battery. Since JBD tools only had csv to save and read data values, we used CSV as our preferred database. From the battery, there is a Universal Asynchronous Receiver/Transmitter (UART) box which is connected to the physical battery and a computer. Through this wired communication, we were able to read the sensor readings and send the data to our connected computer. The computer later sends the data to the web dashboard through Message Queuing Telemetry Transport (MQTT). The program constantly reads the last row of data in the csv file and sends those values to grafana web dashboard. This way the dashboard appears to show real time live data.

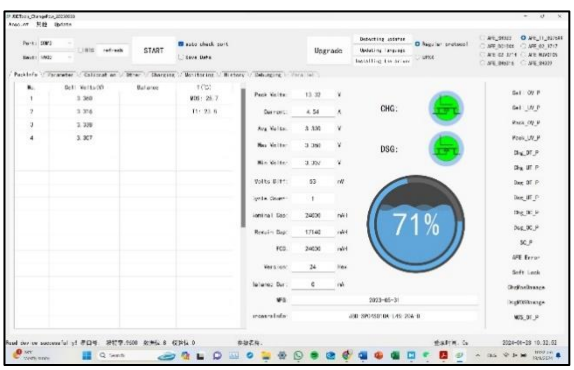


Figure 16: JBD Tools Application

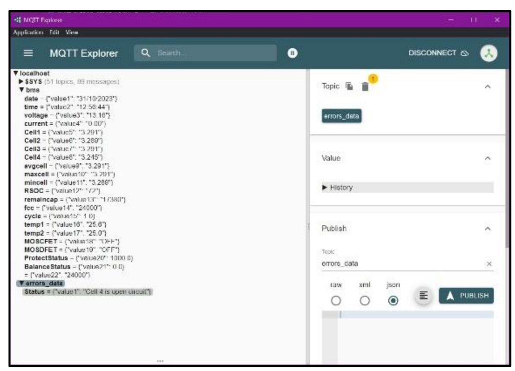


Figure 17: Viewing data transmitted through MQTT

Process of Integration

Since the program to build the dashboard reads from a csv file, it turned convenient to my project as my model reads from a csv file as well. Hence, using multiprocessing package, I combined my program to run concurrently with the dashboard creation program and both reads from the same database. I later created a new csv, where the output of my algorithm is saved and is being displayed back in the dashboard.

There were unexpected logical errors faced on the process of combining which required me to adjust my program and dashboard program. Nevertheless, it was later made possible to work seamlessly without crashes and logic errors.

Figure 18: Addition of error detection algorithm

Creation of 3D model

Using Autodesk Fusion 360 software, I designed a 3D model of the battery to be added in the dashboard. This is to create a more interactive and unique simulation.

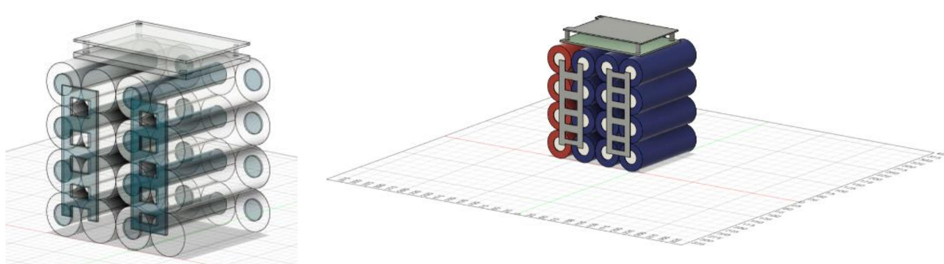
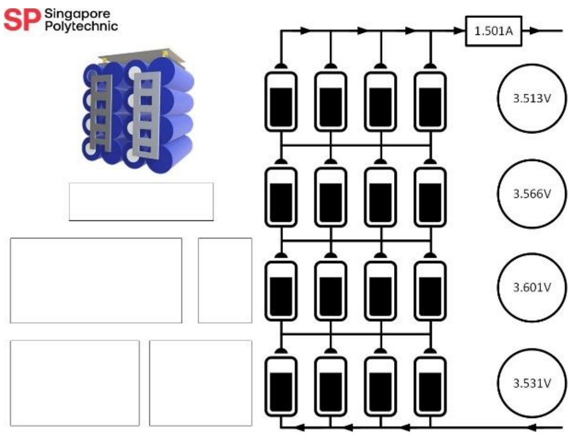


Figure 19: 3D model design of the battery

Difficulties Faced

As the dashboard was built in a web service, we were unable to add the customized design on grafana. Though some attempts and modifications were made, they were not eye appealing from a user's point of view. To have customized dashboard design, the dashboard has to be independently developed as using web frameworks provide limitations. However, due to the time frame at that moment, it was not possible to exclude grafana and code a dashboard from start.



Required to add 3D model of battery and 2D model of the schematic.

However, grafana was limited with its conditional formatting. Only string and integers were able to be sent.

Figure 20: Sketch of desired changes

Solution Method

I created a new html file, used iframe which acts as a screen sharing, where loads and displays the grafana web framework. I later created gif animations of the 3D battery model and 2D schematic model for each error scenarios. Afterwards, additional backend database was created to ensure the new animations and grafana displays are synchronized.

Figure 21: Updated Web Dashboard according to sketch

2.1.4. Writing data to the battery

On the successful completion of having a battery management system with available materials, the team wanted to improve the work by excluding JBD Tools and creating a program to write signals and read signals from the battery. This task was done by Jun Zhe, who later passed the program for signal controlling.

After receiving the program, I first worked on understanding thoroughly on how the program works and is able to send and receive signals from the battery. Later wards, I combined it with the project's dashboard and added buttons which send the signals.

The program is able to start reading, stop reading, turn on and off charging and discharging of the battery. Through this, I was able to automatically turn off charging and discharging without user interface. This happens in a case where a fault predicted on the battery, but the technician is not available to stop the charging status. When this is applied in real life, it could prevent explosions or risks of charging at short circuit.

Figure 22: Web dashboard with write signals to battery

2.1.5. Digital Circuit Simulation

We only had 3 batteries. 1 to display "Good Condition", another to display "Open Circuit" and the 3rd to display "Short Circuit". Hence, to display more errors such as "Overcharging" and errors at other locations of the circuit, I thought of combining the PLECS circuit simulation with the dashboard program by including an interface to automatically run the simulation and process the received data.

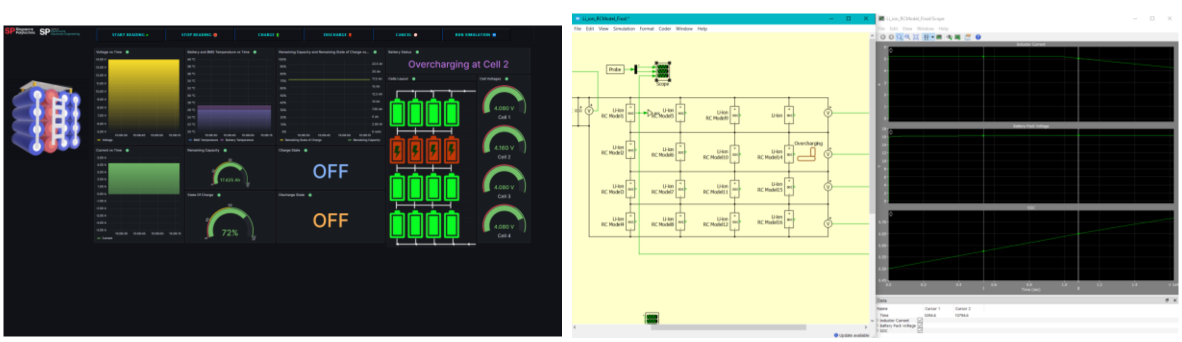


Figure 23: Run Plecs Simulation through Dashboard

2.1.6. Improvements till submission

Once the intended applications of the project was completed, the team successfully created a battery management system with digital twinning. As there was still time for submission, I decided to improve on features and user interface.

Creation of mobile app

I created a mobile app to have a concise dashboard on the hand. I established a connection to the Eclipse Mosquitto server using the MQTT protocol. The process involved initializing the web dashboard program as the client and subscribing to topics such as all sensor readings.

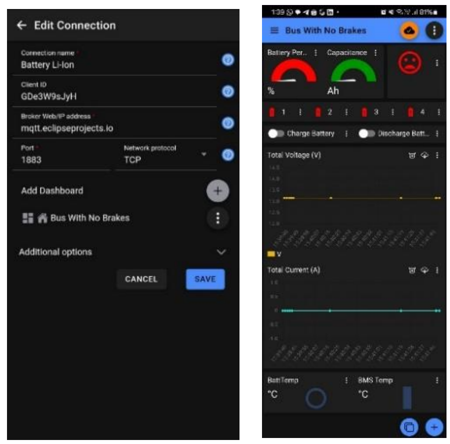


Figure 24: Mobile app configurations and set up

Optimized Backend Management

Through several testing I noticed the program starts tends to crash when left running for a long time. The circuit simulation values were not able to be processed properly either. I noticed that this error is caused as there's too much data held in memory. Hence, I modified the program to clear earlier data if there is more than 1000 rows of data, meaning more than 1000 iterations of battery readings. With this change, there were no crashes or troubles seen after. The program ran smoothly even after hours.

```
def cropFile():
    csv_file = r"C:\DigitalTwining_Practices\TestESCAPE\myflaskapp\AddMyOwn.csv"
    dataFile = pd.read_csv(csv_file)
    df_cropped = dataFile.head(5)
    df_cropped.to_csv(csv_file, index=False)

## Checking length of file:
csv_file = r"C:\DigitalTwining_Practices\TestESCAPE\myflaskapp\AddMyOwn.csv"
df = pd.read_csv(csv_file)
if(len(df) > 1000):
    cropFile()
```

Figure 25: Crop data in database

In addition to this, the execution process was not effective. Multiple commands have to be executed in separate directories to run the whole application. Which would not be convenient for the client and the programmer. Hence I created a .bat file. It is a script file that stores commands to be executed in a serial order. With this way, the next time when a user intends to run the application, instead of manually typing in several command lines, all they had to do is to click on the .bat file and it will automatically run all the application needed.

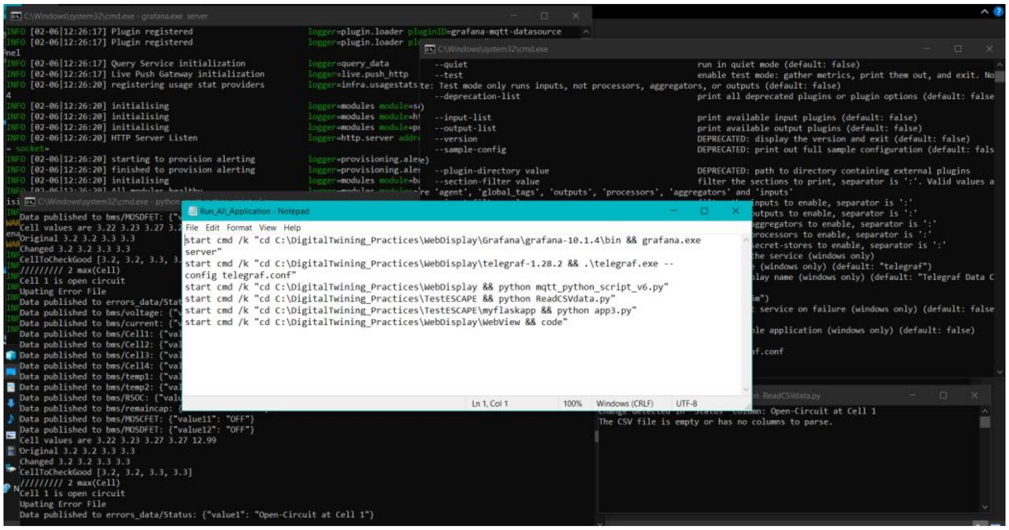


Figure 26: .bat file running all the terminals

Improved UI Design

Following enhancements to features and backend functionality to the best of my capability, I embarked on redesigning the dashboard to enhance its visual appeal for viewers. Through adjustments in color schemes, user interfaces, and iconography, I successfully crafted a professionally designed and meticulously organized web dashboard. Subsequent final modifications culminated in the successful completion of the battery component of the project.

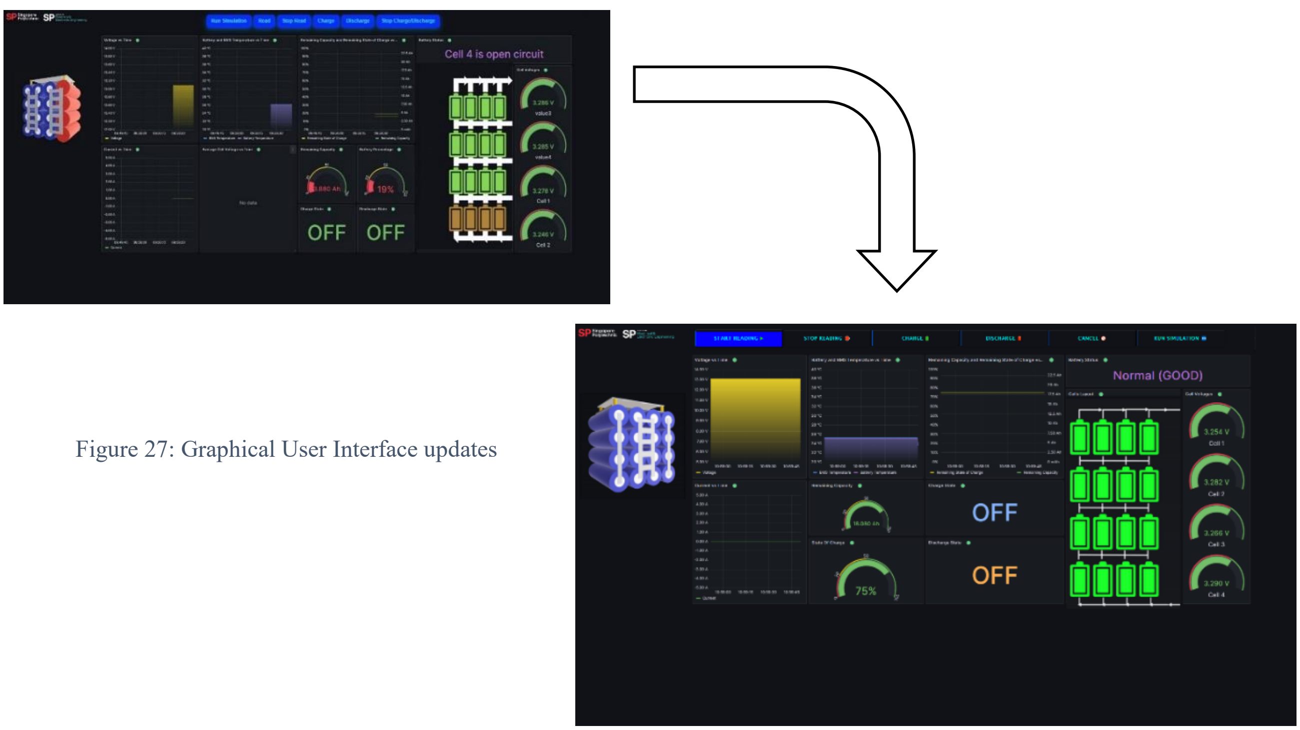


Figure 27: Graphical User Interface updates

3. CDIO (Conceive Design Implement Operate)

3.1.

In the conception phase, the team identified the need for creating a digital twin for both the battery and servo motor to enhance monitoring, control, and predictive maintenance. The primary objectives were to design circuitry for the battery, investigate its behavior, and develop a neural network for digital twinning. For the servo motor, the goal was to study and replicate the circuitry, implement a self-correction algorithm for key components, and integrate it into the digital twin system.

3.2. Design

3.2.1. Battery Schematic Circuitry

Initially, we were given a baseline schematic resembling the physical battery. As a team, we conducted a thorough analysis to identify areas for improvement, considering factors such as voltage regulation, charging dynamics, and discharging cycles. We worked on enhancing the original circuitry design to optimize the battery's performance and reliability. To communicate these modifications effectively, we crafted detailed schematic diagrams, illustrating the refined connections and the integration of specific components within the battery circuit. These diagrams served as a comprehensive visual representation of our team's advancements in the battery circuit design.

3.2.1. Neural Network and error detection algorithm for Digital Twin

For the development of the Digital Twin, I devised a recurrent neural network architecture meticulously crafted to encapsulate the intricate and dynamic behavior exhibited by the battery. I engaged in the collection of extensive training data, utilizing a combination of experimentation and simulations. This meticulous approach was undertaken to guarantee the neural network's ability to faithfully mirror the real-world characteristics of the battery. It also detects type of error, and location of error. Through this process, I aimed to enhance the accuracy and reliability of our Digital Twin model, ensuring it could effectively predict and represent the battery's behavior under varying conditions.

3.2.2. Dashboard for Battery Management System

In crafting the Dashboard for our Battery Management System, our focus was on user-friendliness and effective data visualization. We designed an interface that not only provides a comprehensive overview but also facilitates the efficient management of battery-related data. By seamlessly integrating key parameters such as voltage, current, and temperature, the dashboard enables real-time monitoring, empowering users with the insights needed to make informed decisions regarding the battery's performance and health.

3.3. Implement

3.3.1. Implementation for battery

For the Battery Circuitry, our team successfully translated the physical battery given by our vendors into a digital schematic in PLECS. Rigorous testing was conducted to validate its functionality and efficiency, ensuring that the implemented circuitry aligned seamlessly with the project's objectives. Additionally, we extended the implementation to user interfaces by developing a mobile app and a web dashboard. These interfaces serve as intuitive tools for users to interact with the battery system, providing real-time data insights and control options. Furthermore, the neural network, a crucial component for the Digital Twin, was seamlessly integrated into the system, enabling continuous learning and adaptation to the battery's dynamic behavior over time.

3.4. Operate

3.4.1. Battery Monitoring and Management:

The battery management system was operationalized, providing continuous monitoring and control through the implemented dashboard.

4. Exposures and Opportunities

4.1. Competition Exposure (Bus Tech Grand Challenge 2023)

On 8th November and 25th November 2023, as a team we represented Singapore Polytechnic (SP) by participating in Bus Tech Grand Challenge organized by Singapore's Land Transport Authority (LTA). I got the opportunity to demonstrate the battery management system and present in front of the judges.

Figure 28: 8th November demonstration

During the demonstration event, unfortunately the battery management system encountered unexpected errors at a critical juncture in front of the judging panel. Though the team managed to make reason by mentioning issues with Wi-Fi, it was unfavorable moment to have the project demo be not completely successful. However, after that moment, I tried to investigate the issue, realized it was caused because the program was running for a long time. Hence, I created an auto refresh to prevent this error from happening during finals.

Figure 29: Final day, presentation

The finals day would be memorable day. I represented a team of 4 in front of the judging panel for my presentation. Despite past presentation experiences, I was still anxious worrying about the turnout and the fear of not letting my team down by stuttering. Nevertheless, I was relieved to have a successful presentation. Though our team did not achieve the top 3 awards, the experience was one to be remembered.

4.2. SP Industries and Innovation

On 4th Jan to 6th Jan, I represented the project and SP in its open house. There were some industrial leaders who came to look for potential projects. A handful were impressed by our project idea and wanted to give the contact information to keep in touch with SP for the project of digital twin.

Figure 30: I&I event

4.3. Demonstration to guests and SP students

There were a handful of visits conducted, comprising individuals from various backgrounds, including students from the School of Electrical & Electronics Engineering, and representatives from industrial entities such as StarHub. These visitations provided an invaluable platform for heightened exposure, particularly through demonstrations where the digital twinning of the battery was presented with pride. The recognition garnered from esteemed companies and visiting guests served as a gratifying acknowledgment of the team's collective efforts and commitment invested in the project.

Figure 31: Demonstration to guests

5. Problem encountered & Solutions

When I embarked on this project, I faced numerous challenges, but I learned to overcome them, and the journey was an enlightening experience. The following are some of the challenges I faced during the 6-month internship.

5.1. Understanding Circuit Theory

Problem

As a Computer Engineering student, I had more experience of computing applications but very little towards, electrical knowledge. This project required heavy knowledge towards electrical and electronics. For an Electrical & Electronics Engineering (EEE) student, it would be easier to catch up as they would have background knowledge and use it to conduct thorough research, selecting and comprehending appropriate resources. However, as a Computer Engineer, since I lacked some of the basics and background knowledge in electronics, it was a struggle to find the right resources. While working on the battery project, it was my first time hearing the word “Lithium-ion” battery. Moreover, though I was able to adapt to PLECS software, I struggled understanding the theoretical part of it.

Solution

I borrowed “Principles of Electrical & Electronics” and “Electrical Installation Design” module books from a EEE friend and spent a week to have a quick study up on the basics and background knowledge of battery and its sensors. This later helped me to research online and comprehend helpful and correct resources. While referring to the book, I was able to understand the battery model and create a program to catch its faults.

5.2. Time Constraints

Problem

During the earlier phase of the internship, 1 week before the day of the demonstration for the competition. Several problems arose with the project. The initial design of the dashboard was not very appealing, and the original dashboard had limitations in design. In addition, the background process to run the application was not efficient either. Multiple programs need to be executed in separate directories to display the dashboard. This caused issues with the database too. I was worried about the project as I felt it would not impress the judging panel and there was a lot of room for improvement, but we were only 1 week away from the deadline for the demonstration.

Solution

I realized the cause of these problems were due to multiple people working on the same application but with different methods. Hence, I discussed with the team to change the approach and volunteered to work on improving the application. While I tasked the member who was in charge of the dashboard to change certain data panels and revert back to original program, I created a separate database to write the status outcome. Originally, the status outcome and the battery readings were written to the same database that arose multiple read and write issues. After this change the backend process did not overwrite or work in our intended flow. Once the backend was solved, I started working on improving the design to make it more appealing for clients and eased the process of execution.

6. Conclusion

Throughout my internship journey, I encountered a series of challenges that tested my skills and perseverance. From grappling with unfamiliar concepts in circuit theory to navigating time constraints and devising complex algorithms, each obstacle presented an opportunity for growth and learning.

Despite my initial lack of expertise in electrical and electronics engineering, I quickly adapted and sought out resources to fill the gaps in my knowledge. Collaborating with peers and leveraging their insights proved invaluable in overcoming these obstacles and making meaningful contributions to our project.

The experience of working within a team, particularly during the intense final stages leading up to the competition demonstration, taught me the importance of effective communication and teamwork. Through proactive problem-solving and a shared commitment to excellence, we were able to deliver a refined and functional product that exceeded expectations.

One of the most rewarding aspects of my internship was the opportunity to tackle a daunting challenge: developing a self-correction algorithm for the servo motor's RLC system. Despite initial setbacks, I persisted in my pursuit of a solution, drawing on my mathematical and analytical skills to eventually devise an effective algorithm that optimized performance and minimized sensor usage.

Reflecting on this experience, I am proud of the technical expertise and resilience I demonstrated throughout the project. Moreover, I am excited about the real-world applications of our digital twin-based battery management system and self-correction algorithm, which have the potential to revolutionize industries ranging from renewable energy to industrial automation.

In conclusion, my internship experience has not only strengthened my technical skills but also instilled in me a deeper appreciation for collaboration, innovation, and lifelong learning. I am grateful for the opportunity to contribute to such a impactful project and look forward to applying the lessons learned in future endeavors.

7. List of Key Skills Needed and References

To work on this project, some skills and references would be:

7.1. Understanding and using Plecs

PLECS application to download: https://www.plexim.com/download

Introduction to PLECS: https://www.plexim.com/support/videos/introduction-plecs-standalone

YouTube tutorial: https://www.youtube.com/watch?v=iK2ebV9XcyM&t=657s

Python Automation on PLECS: https://www.youtube.com/watch?v=v5kqeVm6eQA

7.2. Understand Digital Twinning concept

View pdf from ScienceDirect:
https://www.sciencedirect.com/science/article/pii/S2666546820300161

7.3. Understanding Lithium-Ion Battery

About Lithium-Ion batteries: https://en.wikipedia.org/wiki/Lithium-ion_battery

Fault detection: https://pubs.acs.org/doi/10.1021/acsomega.2c04991

7.4. Understanding RLC

Notes from unacademy: https://unacademy.com/content/jee/study-material/physics/rlc-circuit-in-series-with-an-ac-source/#:~:text=When%20the%20inductance%20L%2C%20resistance,but%20the%20voltage%20will%20vary.

Notes from LumenLearning: https://courses.lumenlearning.com/suny-physics/chapter/23-12-rlc-series-ac-circuits/

Notes from ElectronicsTutorials: https://www.electronics-tutorials.ws/accircuits/series-circuit.html

7.5. Web Development

Fron-End tutorial: https://youtu.be/qz0aGYrrlhU?si=dNwMfxSbN2eq9G48

Back-End (FLASK) tutorial: https://youtu.be/Z1RJmh_OqeA?si=o3jqBQKfUW4SlMYM