2022

Efficient Charging System for Hyperloop Application

B. Gillespie

A. Kidd

B. Thompson

P. Vachharajani

presents

EFFICIENT CHARGING SYSTEM FOR HYPERLOOP APPLICATION

by

B. Gillespie
A. Kidd
B. Thompson
P. Vachharajani

(Faculty Advisor: Dr. R. Varma)

ECE 4415/4416 Computer/Electrical Engineering Design Project

Final Report

Submitted in partial fulfillment
of the requirements for the degree of
Bachelor of Engineering Science

Department of Electrical and Computer Engineering
Western University
London, Ontario, Canada
April 15th, 2022

Winner of the Global Undergraduate Awards 2022 Engineering Category

Abstract

Conventional transportation methods are a major contributor to the climate crisis and Hyperloop systems are a proposed form of novel transportation that can provide a fast and energy-efficient alternative. However, current research in Hyperloop technology has neglected the development of the charging infrastructure to facilitate repeated and high-speed charging cycles that minimize battery degradation. To address this, an efficient charging system for Hyperloop application is presented in this paper. Starting with a pre-existing electric vehicle (EV) charging model in MATLAB Simulink, the design was validated and scaled for the Hyperloop application to support a Lithium Iron Phosphate (LiFePO4) battery pack with parameters of 800V and 185Ah. This model was used to test the performance of the proposed charger and observe battery degradation in a variety of scenarios. Upon testing, the system parameters were tuned to successfully charge the specified battery pack in 42 minutes with the potential to decrease the charge duration by up to four times by using a higher charging current. The design was then scaled down to supply a LiFePO4 battery pack with 6.6V and 2.3Ah, which could be built in hardware to safely test and validate the design. The results of the small-scale prototype model were then compared with the full-scale model results. This yielded the same charge curve characteristics with only a 6% difference in the voltage magnitude, thus validating the scalability of the charging system. Finally, to minimize the effects of battery degradation, a temperature control system was designed to keep the battery pack at its ideal temperature (25°C) and was simulated at extreme temperatures of -30°C and 45°C. The results of the temperature control system showed a 7.2% reduction in battery degradation when compared to a system without a temperature control system.

Nomenclature

Table 1: List of symbols used in the report

Symbol	Meaning
$^\circ$	Degree symbol
Ω	Ohms (electrical resistance unit) symbol
P	Power
I	Current
V	Voltage
R	Resistance
W	Watts

Acronym Definitions

Table 2: List of acronyms used in the report

Acronym	Definition
AWG	American Wire Gauge
DoD	Depth of Discharge
EV	Electric Vehicle
LiFePO4/ LiPO4	Lithium Iron Phosphate
NMC	Nickel, Manganese and Cobalt
SOA	Safe Operating Area
SoC	State of Charge
SoH	State of Health

Table of Contents

List of Figures

List of Tables

1. Introduction/Background

1.1 Problem Statement

Hyperloop is a novel transportation system with the potential of creating fast and energy-efficient transportation between cities less than 1500 km apart [1]. A US Department of Energy paper states that Hyperloop transport of passengers could save up to 20% of energy compared to other passenger travel modes [2] such as cars, trains, and planes. As current transportation methods account for 25% of the world's energy usage [3], Hyperloop has the potential to drastically decrease current global energy usage. Also, Hyperloop aims to reach speeds up to 1200 km/h [1], which would significantly reduce travel time between major cities. It has been estimated that a 540 km trip between Toronto and Montreal could be completed by a Hyperloop pod in just 39 minutes [4].

Many groups that are currently designing Hyperloop pods are in various phases of pod and propulsion development with the ultimate goal of providing high-speed intercity service. The current focus of Hyperloop research is aimed at achieving a target speed, levitating the pod, and supporting passengers onboard, which has resulted in little focus on the battery or its charging system. To achieve the rapid acceleration required for a Hyperloop pod, a large battery pack is needed to fulfill the onboard power requirements. In order for Hyperloop to be integrated as a mass transit system, an efficient battery charging system that minimizes battery degradation must exist. Currently, there is no efficient charging system that would meet the criteria of a Hyperloop system; therefore this project will include the design of a Hyperloop charging system to address this gap in technology.

1.2 Background Information and Detailed Literature Review

Hyperloop is a proposed method of transportation that utilizes a system of vacuum-sealed tubes with minimal friction to allow for extremely fast travel. Hyperloop was popularized as a concept by Elon Musk in 2013 through his white paper "Hyperloop Alpha," [1], but the idea behind the Hyperloop came long before that. In the late seventeenth century, the invention of the world's first artificial vacuum led to designs for "underground rapid transit systems" powered by pneumatics. If Hyperloop were to become popularized, these systems could provide a new form

of transportation that is much faster and more energy-efficient than current alternatives. These systems have the potential to become a staple of travel and allow for easy, frequent, and fast travel between major cities.

The current popular design for Hyperloop technology focuses on pods propelled through linear induction motors, which are powered by Lithium Iron Phosphate (LiFePO4) battery packs. These onboard battery packs store power to supply all of the electrical systems on the Hyperloop pod. These battery packs would require frequent and efficient charging if Hyperloop were to be commercialized and used frequently. The average Hyperloop pod is estimated to require 450kW of power at peak consumption and in excess of 700V [5]. A major Hyperloop company provided an estimate for one of their systems that operates at a voltage of approximately 800V and a capacity of 185Ah.

Currently, companies and student design teams that are developing technology for Hyperloop are focused mainly on creating systems for testing Hyperloop technology rather than considering long-term use and efficient charging. In the few instances that teams have explored efficiency and long-term use of Hyperloop, they have focused their attention on power conditioning, battery monitoring, and system safety as opposed to enabling Hyperloop pods to operate over multiple cycles in real-world use.

As an example, prolonged operation of Hyperloop pods can cause harmful effects on the battery pack. A cause of this is increased cell temperatures as well as cell expansion, which can lead to eventual damage in the battery as a result of electrical arcing [6]. To address this, an engineering team, rLoop, utilized a carbon dioxide filled vessel to encapsulate the battery pack and reduce the likelihood of arcing [7]. Other teams such as Georgia Tech [8] and UC Santa Barbara [9] designed a water-cooling system and safety control system, respectively.

The Hyperloop prototype pods that have been developed to date have not required fast turnaround times between tests and thus have not required efficient chargers, but this will change as the technology develops. A common system that is currently used in Hyperloop prototypes is splitting the battery pack into smaller modules and charging each one individually with a low voltage rated battery module charger [5]. This, however, is undesirable since it is not easily scalable to high voltage packs and requires additional time and effort.

A similar industry that utilizes high-voltage charging is Electric Vehicles (EVs); however, not on the same power scale as Hyperloop. EV charging infrastructure is broken into three classes: level one, level two, and level three chargers. Level one generally operates at 120VAC and involves plugging vehicles into a standard wall socket. Level two chargers can be specifically installed in a home and operate at 240VAC, providing faster charging. Finally, there are level three DC fast chargers that operate at voltages exceeding 400VDC with power outputs that are in the range of 120-350kW [10]. The basic blocks of an level three EV charger [11] can be seen in the figure below.

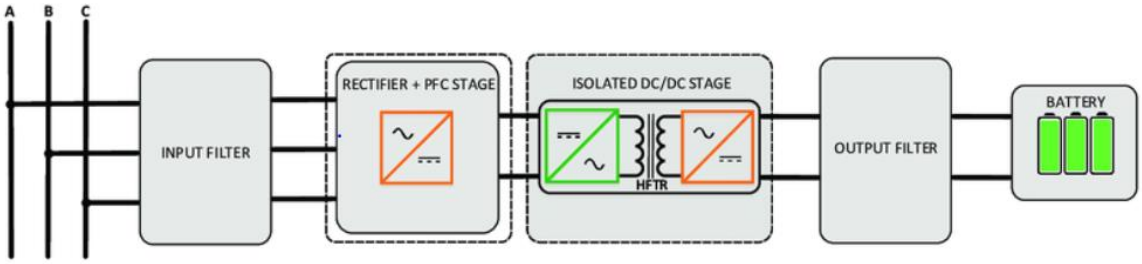


Figure 1: Basic blocks of an EV fast-charging system [11]

EV and Hyperloop batteries have similar needs; however, Hyperloop batteries are predicted to operate at around two times the voltage of an EV [5] and, if popularized, will be used more frequently. The strategies used in EV DC Fast Charging Systems will therefore be utilized as a starting point for the Hyperloop charger design, which can then be scaled and modified to meet the different requirements.

1.3 Project Objectives

The overall objective of this project is to design a charging system that satisfies the needs of a Hyperloop pod. There are three additional sub-objectives to optimize the system, as explored below.

The first sub-objective is to minimize the charging time of the battery pack to reduce the downtime of pods and thus allow for continuous intercity round trips. As an example from the Hyperloop Alpha concept paper [1], Hyperloop transportation could be used to travel between Toronto and Montreal. Trips between these two cities are estimated to take 39 minutes to travel each way, and they would only stop to be unloaded and reloaded at each end. Fast charging is

essential for Hyperloop to act as a main form of public transit to ensure that as many trips as possible can be completed.

Another sub-objective is to reduce the stress placed on the battery pack during charging. Lithium Iron Phosphate cells are very vulnerable to charging and discharging temperatures outside of their safe operating area (SOA), so ensuring that the cells stay within this region during the charging process is extremely important. If cells are not kept within their SOA, it can cause variations in the voltage, reduced charge capacity, and increased internal impedance of the cells. All these factors lead to reduced capacity and lower efficiency of the battery pack [12].

The final sub-objective is to validate the scalability and real-life performance of the charging model design. Depending on the intended use and manufacturer, a Hyperloop pod's power ratings could vary, and thus validating the scalability of the model is critical. Showing various scaled models of the system as well as implementing a physical hardware prototype will verify the system is able to satisfy the needs of any potential pod model.

1.4 Project Constraints

The constraints which must be satisfied for this project are listed below.

1. Budget - The project budget for prototyping is set at $500, so the hardware components for the project must cost less than the budget.
2. Power - The maximum voltage that can be used in the prototype is 40VDC or 30VAC and under 100VA power. These constraints are put in place for the safety and protection of design team members.
3. Timeline - This project must be completed between September 8th, 2021 and April 4th, 2022.
4. Safety - The large-scale design must be created with proper safety considerations made, especially as the power specifications are very large. The hardware prototype will also be implemented safely to ensure that risks are minimized for any users or designers. Safety measures are further explored in Section 4.2 Testing Strategy/Validation and Safety Protocols.

2. Design Approach

2.1 Concept Generation

Two main concepts needed to be determined for starting this project. The first was finding a quality reference paper to gain an understanding of DC chargers and to use as a starting point, seen in sections 2.1.1a-d. The second was to determine methods to reduce battery degradation, seen in sections 2.1.2a-c. The sections below will provide an overview of the technical papers and their content.

2.1.1a Design and simulation of a fast DC recharging station for EV

This paper is an IEEE conference paper that overviews the design of a DC ultra-fast recharging station for electric vehicles. This charge station is designed to be capable of conventional vehicle charging, as well as Vehicle to Grid (V2G) operation. The paper provides an explanation regarding the components that will be needed for an ultra-fast DC charger, specifically reviewing; LCL filter, DC/AC converter, DC/DC converter, and the required capacitors. It also provides all the equations needed to calculate the values for all of these components. It is worth noting that while this paper is said to demonstrate forward vehicle charging as well as V2G charging, the forward vehicle charging circuits are not fully described. This paper also describes the control systems used for the DC/AC and the DC/DC converters. Finally, simulation results are demonstrated for forward charging operation as well as V2G operation. Output display plots include SoC, current, voltage, voltage/current ripple, real power, and reactive power. [13]

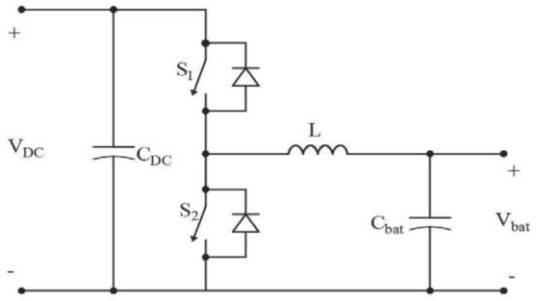


Figure 2: IEEE Paper - Half-Bridge DC/DC converter [13]

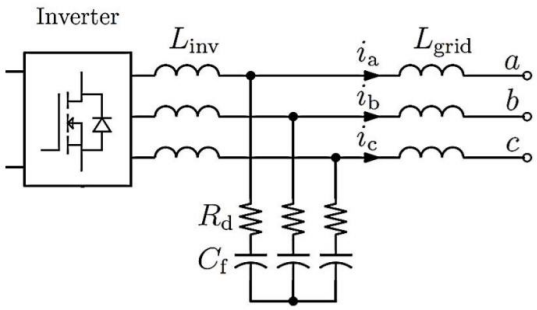


Figure 3: IEEE Paper - Three-phase LCL filter [13]

2.1.1b Modeling of an electric vehicle charging station for fast DC charging

This paper is an IEEE conference paper that describes a charging system that is designed to charge multiple vehicles from a single grid connection. The charging station is capable of conventional vehicle charging as well as Vehicle to Grid (V2G) operation. This paper provides an explanation of capacitance, the battery internals, the three-phase inverter, the LCL filter, the buck and boost charger operation, and control systems. Finally, simulation results are demonstrated for forward charging operation and plots are shown of current, voltage, real power, and reactive power. [14]

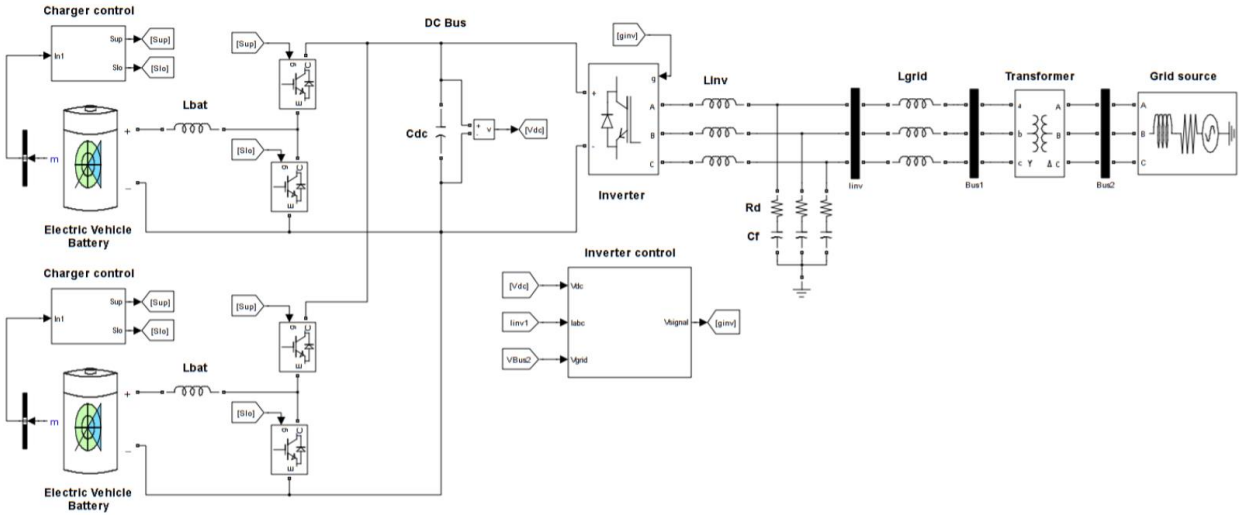


Figure 4: IEEE Paper - Matlab/Simulink SimPowerSystems implementation of the EV charging station [14]

2.1.1c Bharat DC001 Charging standard Based EV Fast Charger

This is an IEEE transaction paper that discusses a two-port DC fast charger based on the Bharat DC001 standard. The main components of this approach are a Delta-Vienna AC/DC rectifier, and a DC/DC converter. This paper provides insight into individual circuit components and the equations needed to calculate values. The simulation results demonstrate the SoC, current, and power, as well as many of the input signals to various stages in the circuit. [15]

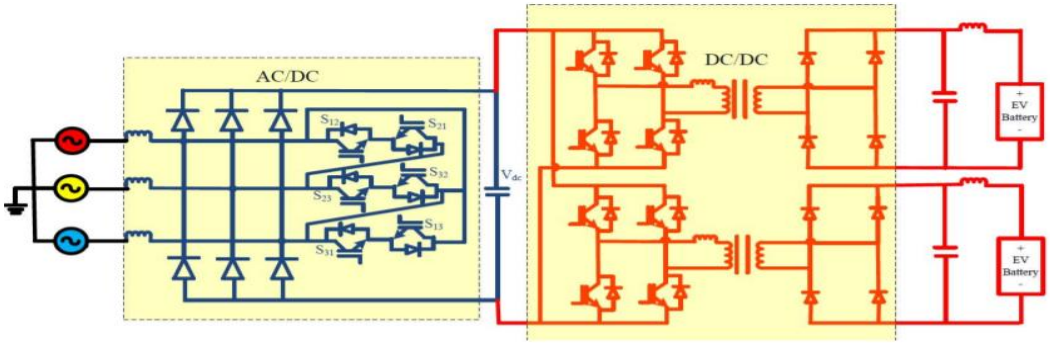


Figure 5: IEEE Paper - Proposed Bharat DC001 EV fast charger [15]

2.1.1d An Output Ripple-Free Fast Charger for Electric Vehicles Based on Grid-Tied Modular Three-Phase Interleaved Converters

This is an IEEE transaction paper that describes the design of a DC fast EV charger that aims to output ripple-free current. The design uses an active interleaved rectifier to convert to DC and then a DC/DC interleaved chopper to regulate the voltage into the battery. This paper is focused on the control system and design parameters that would result in a minimized current ripple and did not go into extensive detail on the setup of the DC fast charger itself. The paper was physically implemented, and simulation results and physical oscilloscope results were shown in the paper. [16]

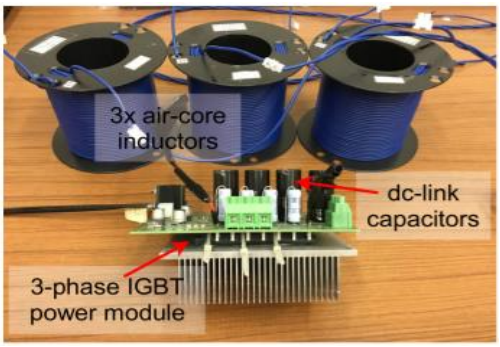


Figure 6: IEEE Paper - Basic three-phase power cell and tower arrangement of three cells [16]

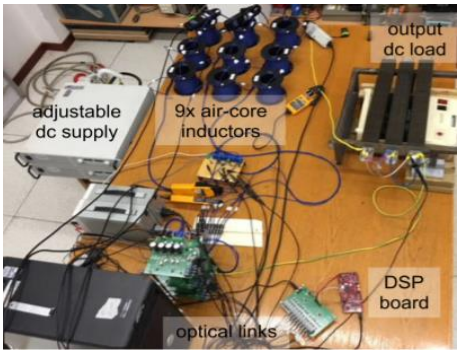


Figure 7: IEEE Paper - Full view of experimental setup [16]

2.1.2 Methods of Minimizing Battery Degradation

To achieve the goal of minimizing battery degradation, methods including temperature control, depth of discharge limitations, and proper charging currents are explored below.

2.1.2a Temperature Effects on Battery Degradation

The reference, BU-808: How to Prolong Lithium-based Batteries, [17] describes the effects that temperature has on the health of batteries. Every battery has an optimal temperature listed on its datasheet which should be followed. Lithium-ion batteries suffer from stress on the batteries induced by over-temperature or under-temperature conditions. This can cause the internal resistance of the battery to increase which results in a decreased maximum capacity and a decrease in the SoC it is able to achieve. The following figure outlines how charging at high temperatures has a negative effect on batteries (this test was conducted on a general Lithium-Ion battery cell).

TEMPERATURE	40% CHARGE	100% CHARGE
0°C	98% (after 1 year)	94% (after 1 year)
25°C	96% (after 1 year)	80% (after 1 year)
40°C	85% (after 1 year)	65% (after 1 year)
60°C	75% (after 1 year)	60% (after 3 months)

Figure 8: Estimated capacity of Li-ion for one year at various temperatures [17]

2.1.2b Depth of Discharge Effects on Battery Degradation

The reference BU-808: How to Prolong Lithium-based Batteries [17] also outlines the effects that Depth of Discharge (DoD) can have on the overall health of batteries. It has been found that both partial discharges and partial charges of a battery can reduce stress on the battery (when compared to full discharge and charge cycles) and prolongs battery life. The following figure outlines how low DoD can extend the number of discharge cycles that the battery can handle while maintaining its useful life (this test was conducted on a Nickel, Manganese and Cobalt battery (NMC) and a Lithium Iron Phosphate battery (LiFePO4)).

Depth of Discharge	Discharge cycles	
	NMC	$LiPO_4$
100% DoD	~300	~600
80% DoD	~400	~900
60% DoD	~600	~1,500
40% DoD	~1,000	~3,000
20% DoD	~2,000	~9,000
10% DoD	~6,000	~15,000

Figure 9: Cycle life as a function of depth of discharge [17]

2.1.2c Charge/Discharge Current Effects on Battery Degradation

The reference BU-802: How does Rising Internal Resistance affect Performance [18] outlines how Lithium-Ion batteries available on the market have set limitations for their charging and discharging current. Different Lithium-Ion chemistries, including LiFePO4, will have different ratings that have been tested rigorously by manufacturers to determine what range of current the batteries can be operated in to not cause heightened battery degradation. When they are operated outside of these ranges there can be detrimental effects on the batteries including increased internal cell resistance and reduced capacity. Reduced capacity causes the battery to discharge faster and in the case of Hyperloop, it would limit the range of the pod. A higher internal cell resistance minimizes the maximum current that the battery is able to output. This makes it extremely important to adhere to these maximum current ratings, especially in the Hyperloop application.

2.2 Concept Evaluation and Selection

2.2.1 IEEE Base Design Selection

Overall, all the aforementioned IEEE papers provided different designs and concepts to use as a baseline for designing the Hyperloop charge system (some of the individual benefits can be seen in the decision matrix below). To optimize the design, some aspects from each paper were combined to fill in areas that were not included in others. The main paper used for the concept was *Design and Simulation of a Fast DC Recharging Station for EV*. This paper was selected as it included the system parameter values, the MATLAB Simulink design for the DC/DC converter and LCL filter, the DC/DC converter control system, and the design equations. Since the base charging system has been created, this allows for focus on innovating for the Hyperloop application, rather than re-inventing already solved problems. This paper provides a lot of detail about the system but is missing some important information such as the AC/DC converter and control system and transformer setup. To bridge gaps in the design, the paper *Modeling of an Electric Vehicle Charging Station for Fast DC Charging*, was used. This paper uses a similar approach for the charging system and provides additional resources to imitate the model, such as how to read the output of the battery and the setup of the grid supply. The metrics for the following decision matrix were chosen as the goal was to identify a design that showcased a design that has similarities to a Hyperloop system as well as providing a strong design process for the systems.

Table 3: IEEE reference concept selection decision matrix

	Designed Voltage	DC/DC design	AC/DC Design	Control system design	Score
Weight	1	4	3	5	
Design and simulation of a fast DC recharging station for EV [13]	3	5	1	4	46
Modeling of an electric vehicle charging station for fast DC charging [14]	3	4	1	4	42
Bharat DC001 Charging standard Based EV Fast Charger [15]	3	3	4	3	42
An Output Ripple-Free Fast Charger for Electric Vehicles Based on Grid-Tied Modular Three-Phase Interleaved Converters [16]	4	1	3	2	27

From the decision matrix, it can be seen that the *Design and simulation of a fast DC recharging station for EV*, *Modeling of an electric vehicle charging station for fast DC charging*, and *Bharat DC001 Charging standard Based EV Fast Charger* papers were the best references for this application. This is due to their in-depth designs of the specific systems that will be implemented in this Hyperloop charger design.

2.2.2 Battery Degradation Reduction

The three main contributors to battery degradation were found in the resources above, and for the design of an efficient Hyperloop charging system, some of these concepts need to be reduced to ensure the system aids in maintaining battery health. A decision matrix was used to weigh the effects of each contributor on a Hyperloop system. The metrics for the following decision matrix were chosen as the goal was to identify degradation contributors that had a strong impact on battery health as well as ones that were manageable in a charging system.

Table 4: Battery degradation concept selection decision matrix

	Effect on Pod Range	Ability to Control	Effect on internal resistance	Score
Weight	2	3	2	
Temperature Effects on Battery Degradation [17]	3	3	2	19
Depth of Discharge Effects on Battery Degradation [17]	1	1	2	9
Charge/Discharge Current Effects on Battery Degradation	3	3	1	17

From the decision matrix, it can be seen that the temperature and charge/discharge current effects on battery degradation score the highest as they have the largest negative effect on the internal resistance of the battery which leads to degradation, as well as they have parameters that can be controlled in this project. Controlling the depth of charge and depth of discharge can be beneficial; however, that is up to the Hyperloop operator in how they use the system. The recommendation given to an operator would be to not fully charge or discharge the battery system if possible. The designed system will consider temperature control as well as regulating the charge current for the Hyperloop battery system.

3. Design Analysis

3.1 Complete Analysis/Calculations

3.1.1 Validation of EV Charging Design

To understand the design behind a DC fast charging station, an EV DC Fast Charger Design simulation was recreated in MATLAB Simulink. Two main papers were used to create the test model, *Design and simulation of a fast DC recharging station for EV*, and *Modeling of an Electric Vehicle Charging Station for Fast DC Charging*. These papers outlined the major components of a DC charging station but were not complete with schematics and simulations which is why they were used in tandem to create a simulation model. As a note, this first simulation was a recreation of the IEEE paper and does not reflect the values or goal of a Hyperloop charging system, it is solely for proof of the design concept, and understanding of the required components. This EV charger model was created to serve as a base for adapting to the Hyperloop application; the Simulink block diagram can be seen below, as well as the battery status results.

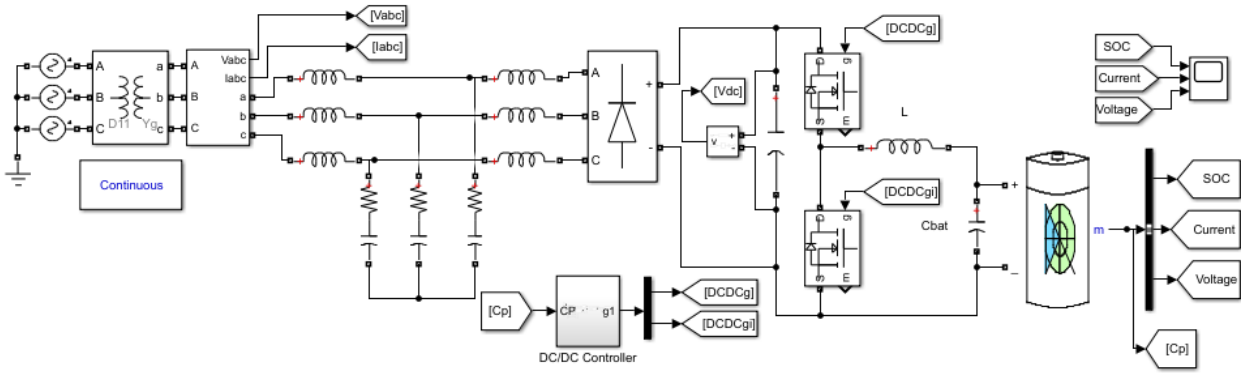


Figure 10: Full top-level schematic of the recreated charging system

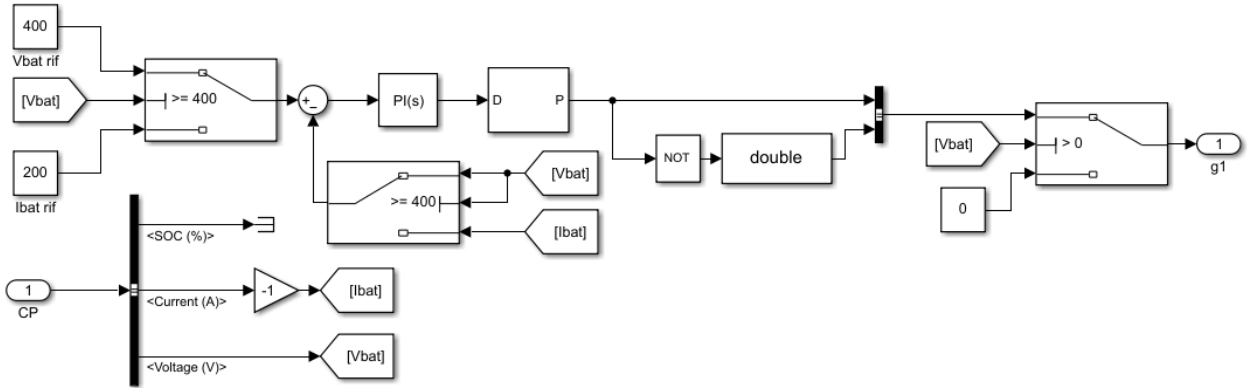


Figure 11: DC/DC converter control system schematic of the recreated charging system

The simulation paper was simulated for a duration of one second and the transient response was started at 0.2s. The following figures compare the results from the simulation paper and the recreation of the system.

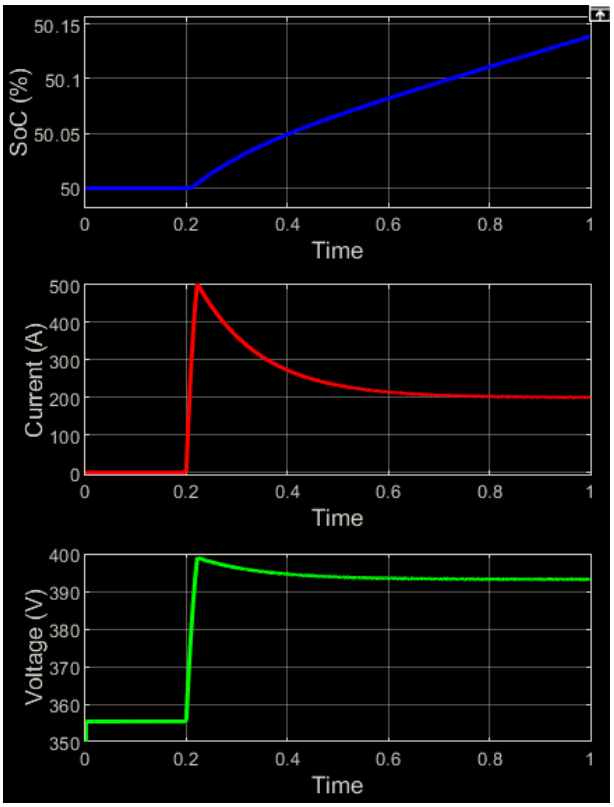


Figure 12: Simulation results from recreation

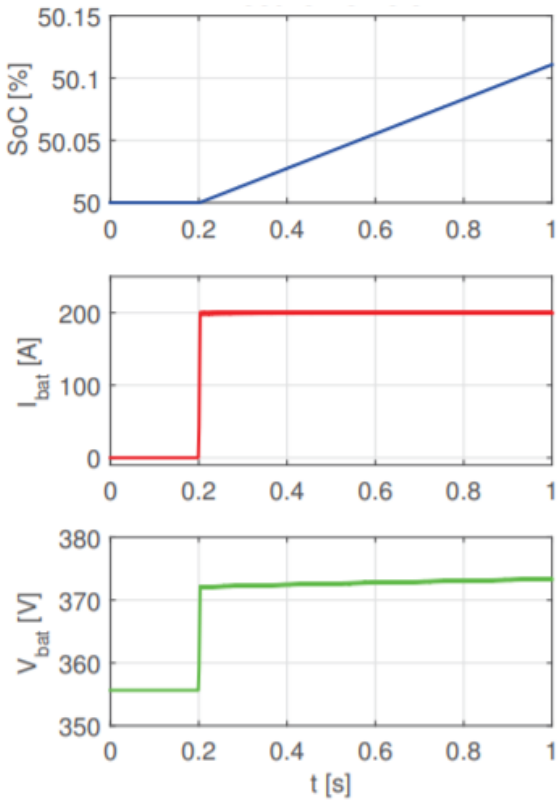


Figure 13: Simulation results from original IEEE Paper [13]

The results from the two simulations are very similar which demonstrates that the circuit and control systems were recreated correctly. The SoC and current plots of the two models show very similar steady-state outputs. The steady-state voltage of the battery pack in the recreated model (pictured on the left) is 393V, and in the original paper (pictured on the right) it is

approximately 373V. This small discrepancy is likely due to a slight difference in the transformer properties (not all transformer properties were shown in the paper, so data was taken from other sources), or due to internal impedances in the diode rectifier (this implementation was also not shown in the paper). It can also be noted that there are also discrepancies in the transient state, in which the recreated model has some overshoot, which can be attributed to the sampling time selected in the paper, while in the recreated model a continuous-time sampling was used.

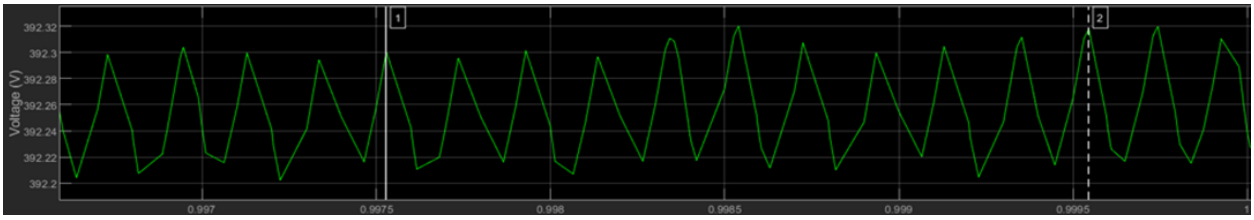


Figure 14: Simulation voltage ripple results from recreation

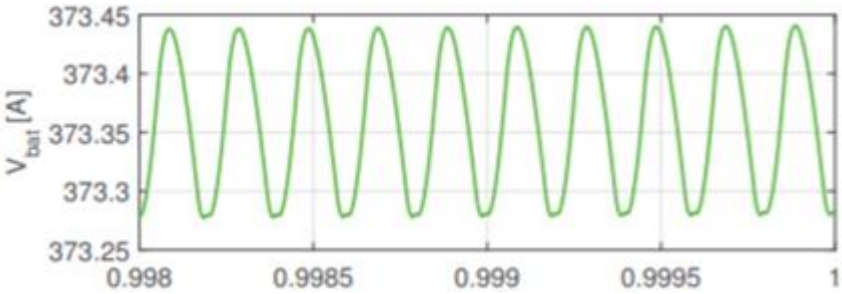


Figure 15: Simulation voltage ripple results from Original IEEE Paper [13]

$$Frequency\ of\ recreated\ Voltage\ oscillations\ =\ 1/(2.016ms/10)\ =\ 4960.32\ Hz$$
$$Frequency\ of\ original\ IEEE\ paper\ Voltage\ oscillations\ =\ 1/(2ms/10)\ =\ 5000\ Hz$$

It can be seen above that the voltage has a very similar ripple frequency; the small discrepancy is likely due to human error as a result of manual cursor placements. This further proves that the control system is operating correctly as described in the Simulation paper [13].

3.1.2 Parameter Scaling Calculations

In order to obtain a full-scale Hyperloop model as well as a small-scale model for the prototype, the EV model needed to be converted to a per-unit system. In the table below, the parameters for the EV charger, per unit system, Hyperloop charger, and scaled-down charger are shown. This list is composed of all parameters as described in the *Design and Simulation of a Fast DC Recharging Station for EV* paper. The input parameters such as power requirements were then adapted for the subsequent systems. After determining these values, the models were simulated, and the values were tuned accordingly to achieve the desired voltage and current in the batteries.

Table 5: Per-unit calculations and scaling

	Original Values		Per Unit		Full Scale		Small Scale	
Base Calculation								
Vdesired	4.00E+02	V			8.71E+02	V	7.20E+00	V
Idesired	2.00E+02	A			2.00E+02	A	2.50E+00	A
Three-phase power	1.39E+05	VA			3.02E+05	VA	3.12E+01	VA
Base Impedance	1.15E+00	Ω			2.51E+00	Ω	1.66E+00	Ω
LCL Filter Values								
Lgrid	7.00E-04	H	6.06E-04	pu	1.50E-03	H	N/A	
Linv	5.00E-04	H	4.33E-04	pu	1.09E-03	H	N/A	
Cf	2.00E-04	F	1.73E-04	pu	3.27E-03	F	N/A	
Rd	1.50E+00	Ω	1.30E+00	pu	1.09E+00	Ω	N/A	
New Base Calculation								
Base Voltage on DC side	6.53E+02	V			1.42E+03	V	1.20E+01	V
New Base Impedance	3.08E+00	Ω			6.71E+00	Ω	4.43E+00	Ω
DC/DC Controller Values								
Vbat rif	4.00E+02	V	6.12E-01	pu	8.71E+02	V	7.20E+00	V
DC Side Values								
L into DCDC	1.00E-02	H	3.25E-03	pu	2.18E-01	H	1.44E-01	H
Cdc	4.70E-02	F	1.53E-02	pu	1.02E-01	F	6.76E-02	F
Lbat	9.37E-03	H	3.04E-03	pu	2.04E-02	H	1.35E-02	H
Cbat	1.25E-05	F	4.06E-06	pu	2.72E-05	F	1.80E-05	F

3.1.3 Power Requirement Calculations

As an example, to demonstrate that the charger can facilitate continuous intercity travel to meet the project objective of minimizing downtime, Toronto to Montreal travel was used. The following calculations were completed to determine the power requirements of this trip.

The power of the pod as described by a Hyperloop company that was consulted for the purposes of this project was determined as follows:

$$P = V * I = 800V * 185Ah = 148kWh$$

At 1200 km/h, the average power consumption of the system is estimated to be 100kW [1]. Based on that information the power consumption for a one-way Toronto to Montreal trip (39 min = 0.65h) would be as follows:

$$0.65h * 100kW = 65kWh$$

$$65kWh = 44\% \, of \, battery \, capacity$$

3.2 Simulink Model Design

3.2.1 Full-Scale Model

This model was created to demonstrate the effectiveness of the charger at the Hyperloop scale. A source at a major Hyperloop company provided the approximate battery pack specifications of a project which was rated for 800V and had a capacity of 185Ah. This was used to determine the charging parameters of 800V and a charge current of 185-740A. This charging current was chosen as there are 74 cells in parallel and 242 cells in series. Each cell has a standard charge current of 2.5A and a maximum charge current of 10A. Using the per-unit values calculated from the *Design and Simulation of a Fast DC Recharging Station for EV* paper, the model was scaled up and the components were tuned to obtain the desired voltage and current into the battery.

Additional modifications include using an LC filter to remove any harmonics coming out of the isolation transformer, modifying the AC/DC rectifier controller to operate in one direction to simplify the setup, and simplifying the DC/DC controller and tuning the PI values to obtain fast response times. The component values can be seen in the following figure.

Table 6: Component values for full-scale model

Component	Value
System	
Three-phase source	20000V, 60Hz
Three-phase transformer	Delta-Wye, 20000V to 1000V
Linv	1.2mH
Cf	200uF
Rd	1.5 Ω
Ldc	218mH
Cdc	47mF
Lbat	10mH
Cbat	5uF
Battery	LiFePO4, 798.6V nominal, 871.2V max, 185Ah, 10% initial SoC
AC/DC Controller	
Sawtooth Generator	5000Hz, 180 degree phase
Sine Waves	Amplitude 871.2V
DC/DC Controller	
PI1	Kp = 0.4, Ki = 4
PI2	Kp = 5, Ki = 500
PI3	Kp = 0.5, Ki = 4

The following figure shows the Simulink model of the full-scale charger model.

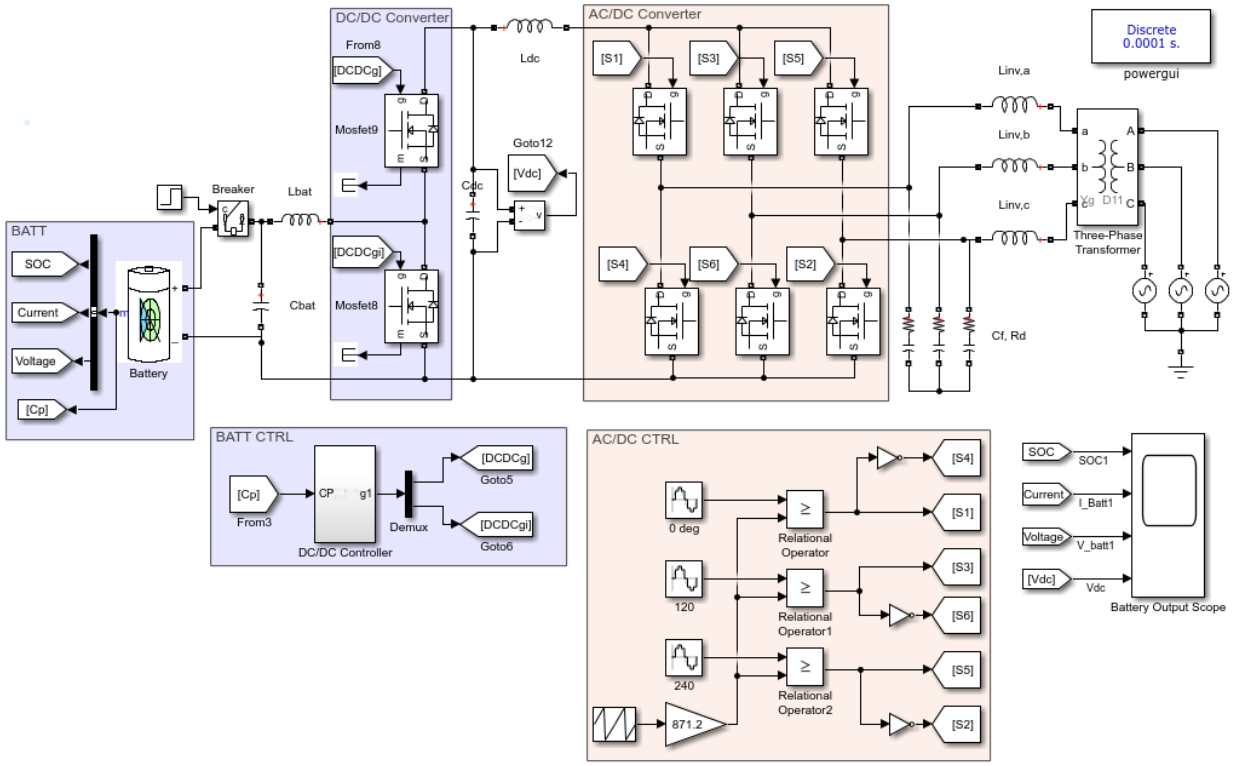


Figure 16: Full-scale Hyperloop charger model

The following figure shows the Simulink model of the DC/DC Controller for the full-scale charger.

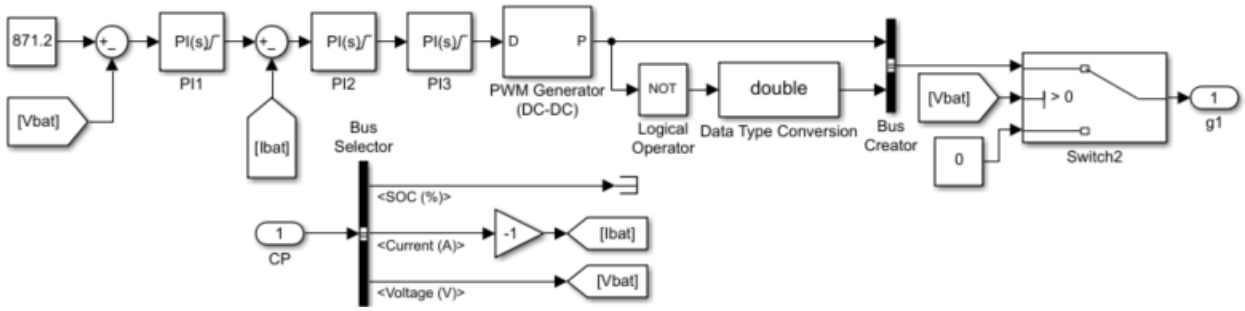


Figure 17: Full-scale Hyperloop charger model, DC/DC control system

The results from this model will be seen in section 4.3.1 Full-Scale Simulation of Hyperloop Charger.

3.2.2 Small-Scale Simulation of Hardware Prototype

The next model is the scaled-down charging system that was later implemented in hardware. The goal was to visualize a similar result to what was seen in the full-scale model, with the system using a battery pack made up of two 3.3V, 2.3Ah cells rather than the aforementioned high voltage pack. This pack size was chosen to stay within the power constraints of the project. The parameters of the pack are in total 6.6V, 2.3Ah, requiring a charge current of 2.5-10A. Using the per-unit values calculated from the *Design and Simulation of a Fast DC Recharging Station for EV* paper, the model was scaled down and the components were tuned to obtain the desired voltage and current into the battery.

The three-phase supplies that were available for use in this prototype had insufficient power available, and thus the three-phase portion of the model was unable to be built in hardware. This meant that the AC/DC rectifier was removed from this implemented design and the 2.5A charge current for the battery pack was supplied from a DC switching supply, see Section 7.3 Appendix C – Full Prototype Circuit Including Three-Phase Supply for a view of the design including the three-phase source (highlighted components were implemented). This design therefore modeled the DC/DC converter and its controller, as well as appropriate filtering. The component values can be seen in the following figure.

Table 7: Component values for small scale model

Component	Value
DC Source	13.5V
R	1.5 Ω
Cdc	24uF
Lbat	5mF
Cbat	6.8uF
Battery	6.6V nominal, 7.2V maximum, 2.3Ah (LiFePO4), 10% initial SoC
DC/DC Controller	
PI1	Kp = 0.4, Ki = 4
PI2	Kp = 5, Ki = 500
PI3	Kp = 0.5, Ki = 4

The following figure shows the Simulink model of the small-scale charger model.

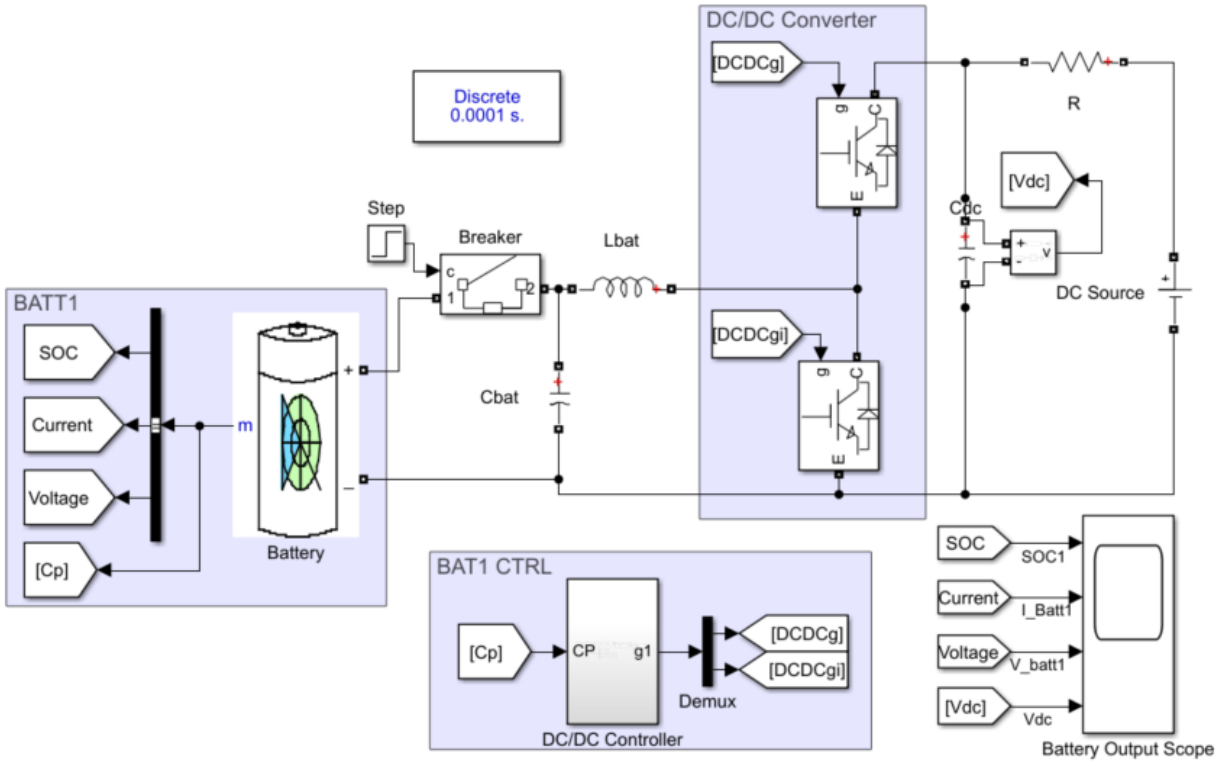


Figure 18: Small scale Hyperloop charger model

The following figure shows the Simulink model of the DC/DC controller for the small-scale charger model.

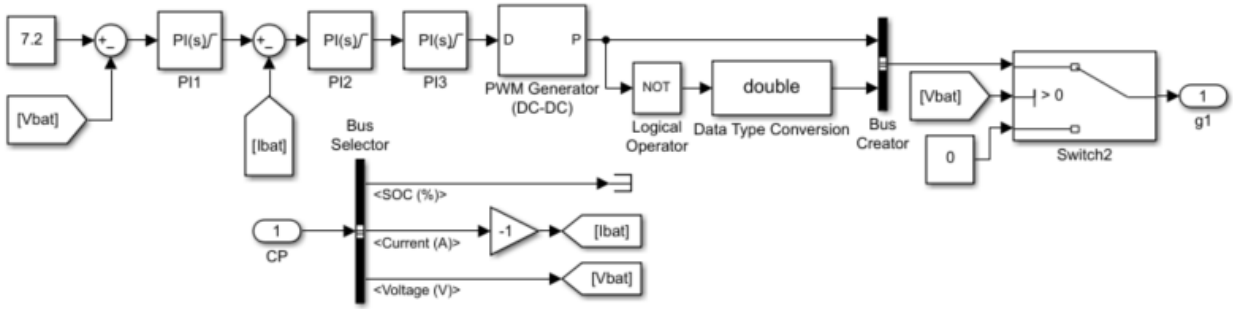


Figure 19: Small scale Hyperloop charger model, DC/DC control system

The results from this model can be seen in section 4.3.2 Small-Scale Simulation of Hardware Prototype, where they are compared with the data that was collected by running the prototype charger using the same parameters.

3.2.3 Small-Scale Simulation for Optimizing Battery Health

The next model is the scaled-down charging system that was used for optimizing the battery health and viewing degradation. The goal was to use this model to test multiple charging currents, test the effects of temperature on batteries, and determine a method of charging batteries to reduce degradation. The scaled-down model was chosen for these tests as the charging parameters are set the same as they would be at any scale, so the effects seen by the battery cells are identical and simulations are able to run much faster at this scale. The consistent system scaling of this model provided an accurate understanding of the metrics and required less computation power.

The design features four main modifications from the previous scaled-down model; a charging control system and a temperature control system. The charging control system allows for the battery to be cycled between 10% and 90% SoC. This range was selected to show 80% DoD cycles, which is common with LiFeO4 battery packs. When the battery reaches 10%, it switches to the charging circuit and when the battery reaches 90%, it turns off the charging circuit and connects to a discharge resistor. This process then repeats for a set number of cycles. The temperature controller keeps the temperature of each battery cell around 25°C, which is used to visualize the benefits of temperature control. The charge current was varied in this system between the rated values of 2.5A and 10A for fast charging depending on the simulation being run. Finally, the cell chemistry and rating parameters of the cells were set to be the same as in the datasheet of the ANR26650M1B [19] battery cells. This ensured that the battery health results seen in the simulation were the same as they would be in the hardware prototype.

Table 8: Component values for small scale model with temperature control

Component	Value
DC Source	13.5V
R	1.5 Ω
Cdc	24uF
Lbat	5mF
Cbat	6.8uF
Battery	3.3V nominal, 3.6V maximum, 2.3Ah

	(LiFePO4), 10% initial SoC
Battery Temperature	Ambient and internal temperatures are set to test values or set with a control system
DC/DC Controller	
PI1	Kp = 0.4, Ki = 4
PI2	Kp = 5, Ki = 500
PI3	Kp = 0.5, Ki = 4

The following figure shows the Simulink model of the small-scale charger with the additional control systems.

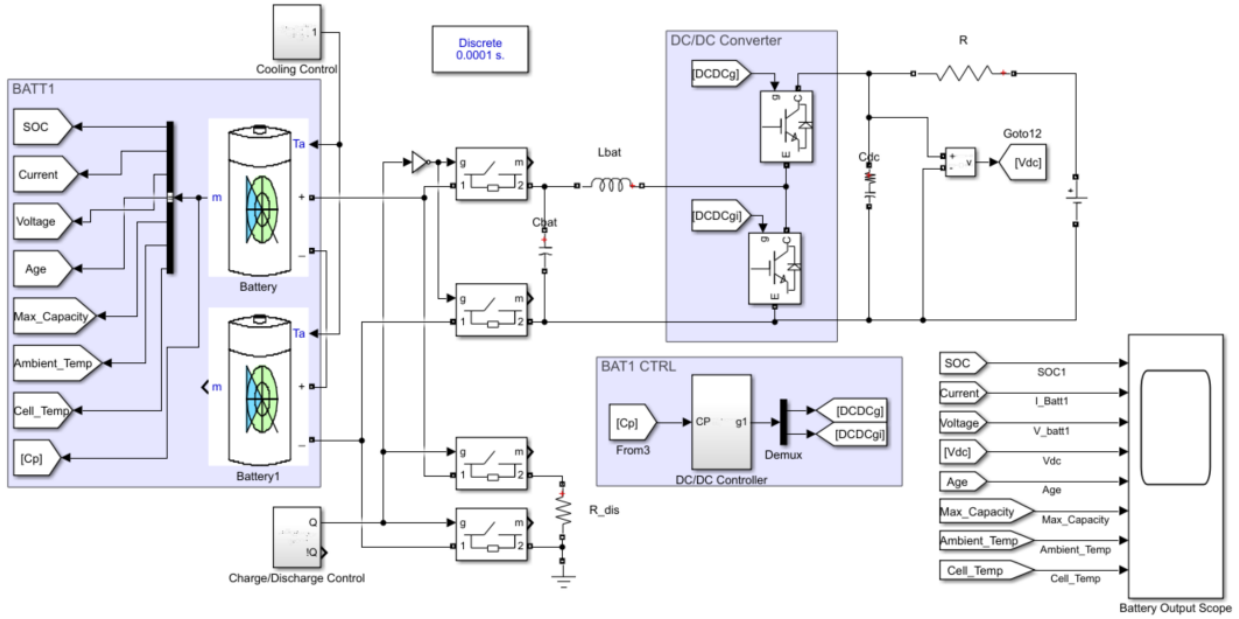


Figure 20: Small scale temperature control Hyperloop charger model

The following figure shows the Simulink model of the DC/DC controller for the small-scale charger model.

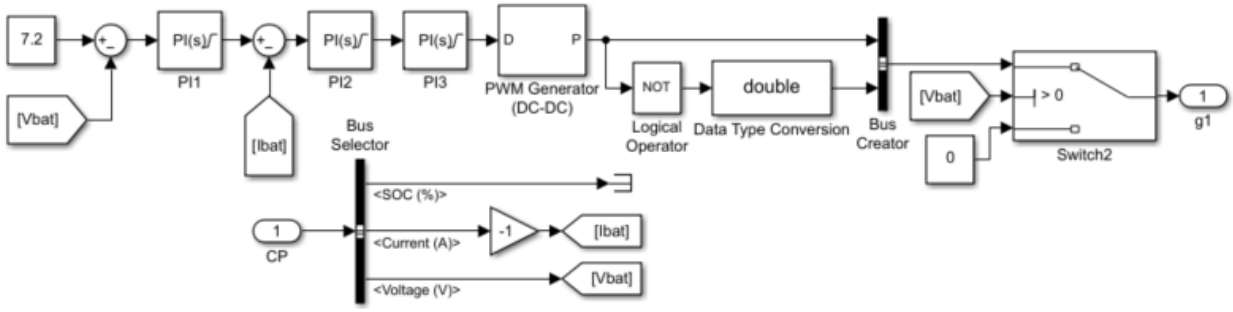


Figure 21: Small scale temperature control Hyperloop charger model, DC/DC control system

The following figure shows the Simulink model of the charge/discharge controller for the small-scale charger model.

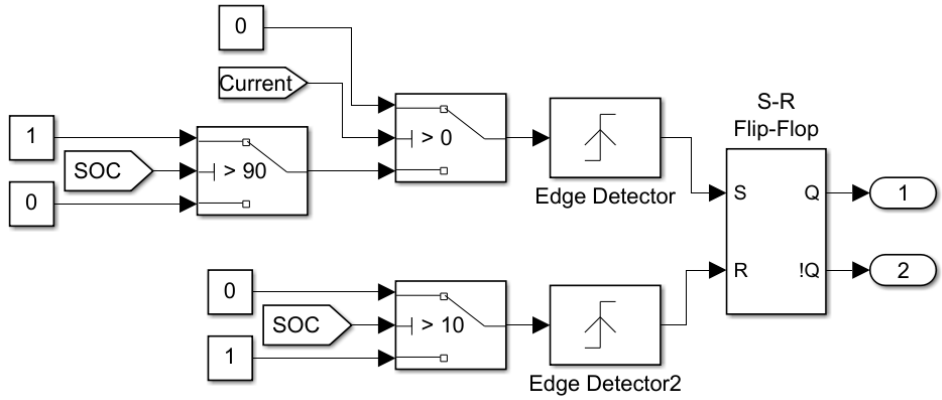


Figure 22: Small scale temperature control Hyperloop charger model, charge/discharge control system

The following figure shows the Simulink model of the temperature controller for the small-scale charger model.

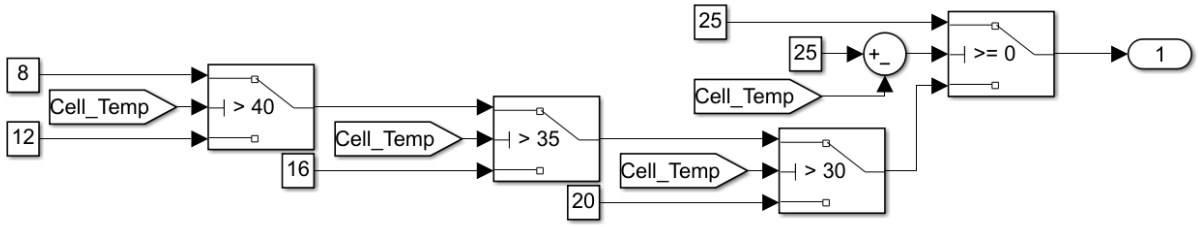


Figure 23: Small scale temperature control Hyperloop charger model, temperature control system

The results from this model can be seen in section 4.3.3 Small-Scale Simulation for Optimizing Battery Health.

4. Results and Validation

4.1 Prototype

4.1.1 Hardware Design

In order to validate that the simulated charger design would be valid in real-life conditions, a hardware prototype was implemented. This was scaled down from the full-scale Hyperloop battery pack to an acceptable size for the safety constraints of this project prototype. As a result, the design was built for a two-cell battery pack with a nominal voltage of 6.6V and a 2.3Ah capacity. This prototype utilized an Arduino UNO to control the sensors, actuators and charging circuit to match the parameters of the simulations.

The sensors that were included are a current sensor, a voltage sensor, and a temperature sensor. The inputs from these sensors were used to control the safety of the charging circuit and display charging status information to the user. For each of the current, voltage, and temperature, an SOA range was set in the program. If any of these conditions exit the range, the charging stops.

The actuators that were auxiliary to the charging circuit included a liquid crystal display with an I2C adapter (LCD screen), DC fan, and tri-coloured LED. The LCD screen was used to display instructions to the user to start charging, as well as display the values from the sensors outlined above. The DC fan was used to demonstrate the functionality of the cooling system, as it turned on when the temperature reading exceeded the SOA. Finally, the tri-coloured LED was used as a status light to indicate the charging state. The final actuator was a relay that controlled the high-current charging circuit from the low-current Arduino circuit.

To ensure this circuit would accurately represent the simulation models, the same component values and battery cells were used (LiFePO4).

The high-current charging circuit is powered by a DC source, which represents the voltage exiting the AC/DC rectifier in the simulated models. This adjustable DC supply was set to 13.5V and 2.5A to imitate the output of the AC/DC rectifier stage.

The table below outlines all the components used in the hardware prototype. A table that outlines all components that were purchased (including spares) and the total cost of the prototype can be found in section 7.2 Appendix B – Complete Hardware Cost Breakdown.

Table 9: Hardware prototype components

Part	Part Number	Specifications	Quantity Used	Price/unit [$]	Notes
Battery Charging (power supply, DC/DC)					
DC power supply	1608A	BK precision DC regulated power supply	2	0.00	For Vdc and powering BJTs
LiFePO4 Batteries	ANR 26650 M1B	3.25V, 2500 mAh	2	13.50	Two batteries in series
Diode	SB520-E3/54	Maximum repetitive peak reverse voltage = 20V (14Vrms), max current = 5A	2	0.76	For DC/DC converter
Inverter logic gate	CD74HCU04E	Propagation Delay: 6ns at VCC = 5V (Arduino period is 62.5ns)	1	0.90	To invert the PWM signals so the MOSFETs can be controlled in inverted pairs from the same Arduino pin
BJT	BC548CTA	General purpose NPN	2	0.44	To amplify the MOSFET gate voltages to 10V (5V from Arduino pin)
MOSFET	IRF520NPBF	Vgs(thres) = 2-4V, Vgs(max) = +/-20V, Id = 9.7 @ T=25deg	2	1.37	For DC/DC converter
Control					
Arduino	SMD R3 ATMEGA328	Arduino UNO	1	30.14	To control the charging
Sensors					
Hall-effect current sensor	ACS712	T= -40 to 125 degrees, low-voltage analog output with amplifier	1	3.00	To determine current entering battery
Voltage sensor	GR0430X10	0-25V range (charging voltage: 3.6 * 3 = 10.8)	1	1.78	To determine voltage across battery terminals
Temperature sensor	TMP36GT9Z	Plus/minus 2 degrees C over the -40 to 125 range	2	2.52	To monitor battery temperature
Pushbuttons	N/A	General-purpose	2	0.00	For the start and stop charge button
Actuators					
Tri-colour status LED	N/A	Common anode	1	0.00	Display charging status
LCD and I2C converter	N/A	16x2	1	0.00	To display SoC and prompts

Fan	N/A	Two-wire +/- DC motor	1	0.00	To cool off batteries if they get too hot
Relay	CA-MW-CB063	5V, NO/NC	1	2.60	To control the high current circuit from the Arduino
BJT	PN2222ABU	General purpose NPN	1	0.51	To amplify the Arduino signal to actuate the relay
Diode	1N4007	High voltage/current rated	1	0.30	Across relay to protect MCU from back EMF voltage spikes
Other					
14-gauge wire, black	N/A	The shop is providing	A/R	0.82	For ground wires
14-gauge wire, red	N/A	The shop is providing	A/R	0.82	For power wires
14-gauge wire, blue	N/A	The shop is providing	A/R	0.82	For signal wires
Battery holding units	N/A	For 26650	2	0.45	To hold the cells together
Bus bar for batteries	N/A	For 26650, 0.15*10mm	A/R	2.76/m	To connect the negative of one battery to the positive of the other (series connection)
Heat sink	N/A	Metal fins for heat dissipation	3	0.00	To dissipate heat on power resistor and MOSFETs
Breadboard	N/A	For AWG22 wire	2	0.00	To host the low-current connections
Solder	N/A		A/R	0.00	
Heat shrink	A580	Assorted	A/R	0.03	For soldering
Screw-on wire connectors	25-AWC	Assorted	A/R	0.39	Instead of soldering some connections
Circuit Components					
Lbat	B82615B2302M001	2.5mH, rated for 3A	2	13.54	5mH needed, so two are placed in series
Cbat	EKYB101ELL6R8ME11D	6.8uF, rated 100V	1	0.52	
Cdc	400AX24MEFC12.5X16	24uF, rated 400V	1	1.45	
R (into DC/DC)	MPT100T1R5F	1.5Ω, rated 100W	1	5.51	

The figure below shows a circuit schematic of the hardware. This includes all components listed above hosted on the microcontroller.

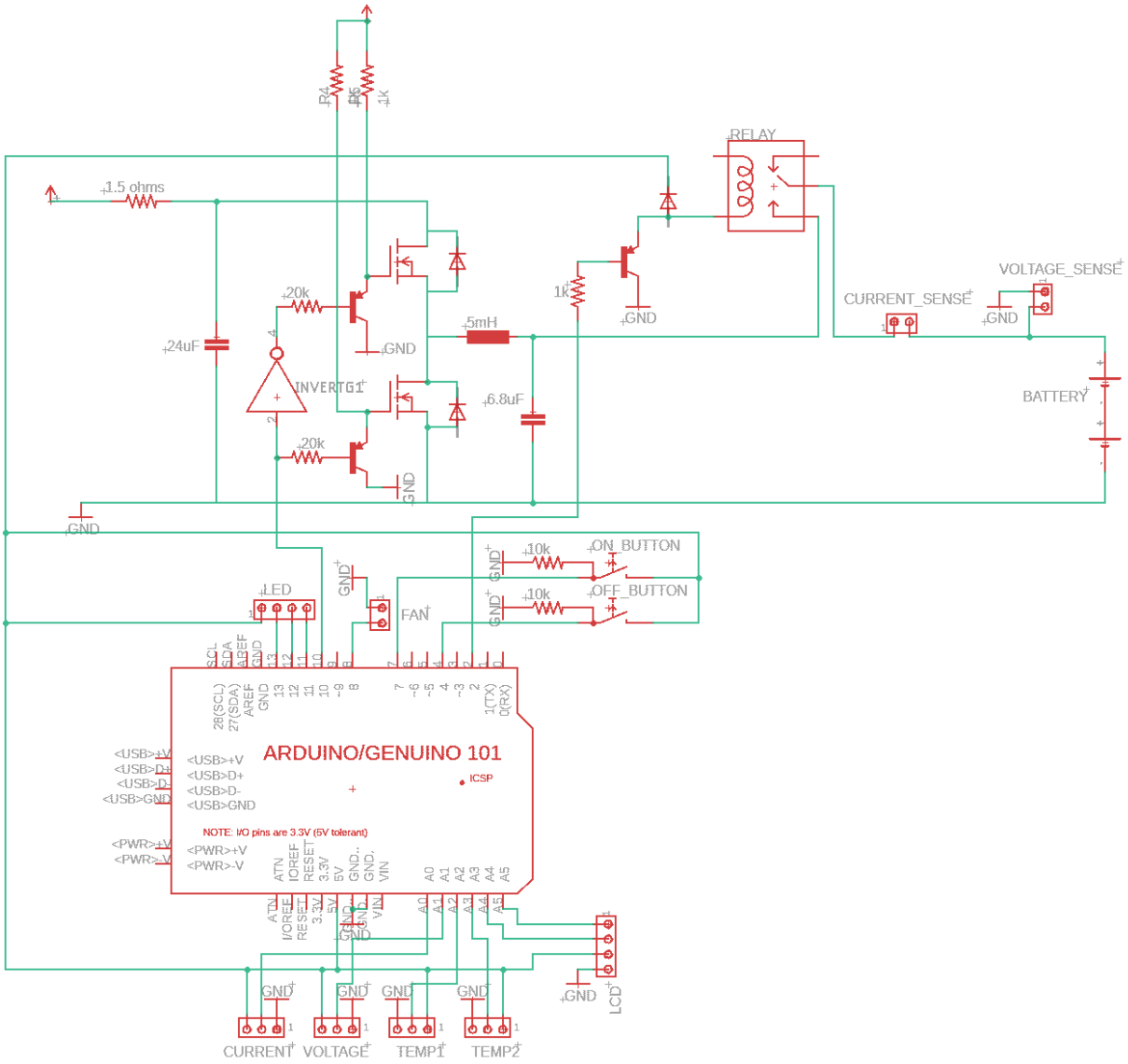


Figure 24: Schematic of hardware prototype

4.1.2 Software Design

The control of the hardware prototype was executed using an Arduino UNO microcontroller. In order to develop the software structure, the flow chart outlining the necessary logic can be seen in Figure 25 below.

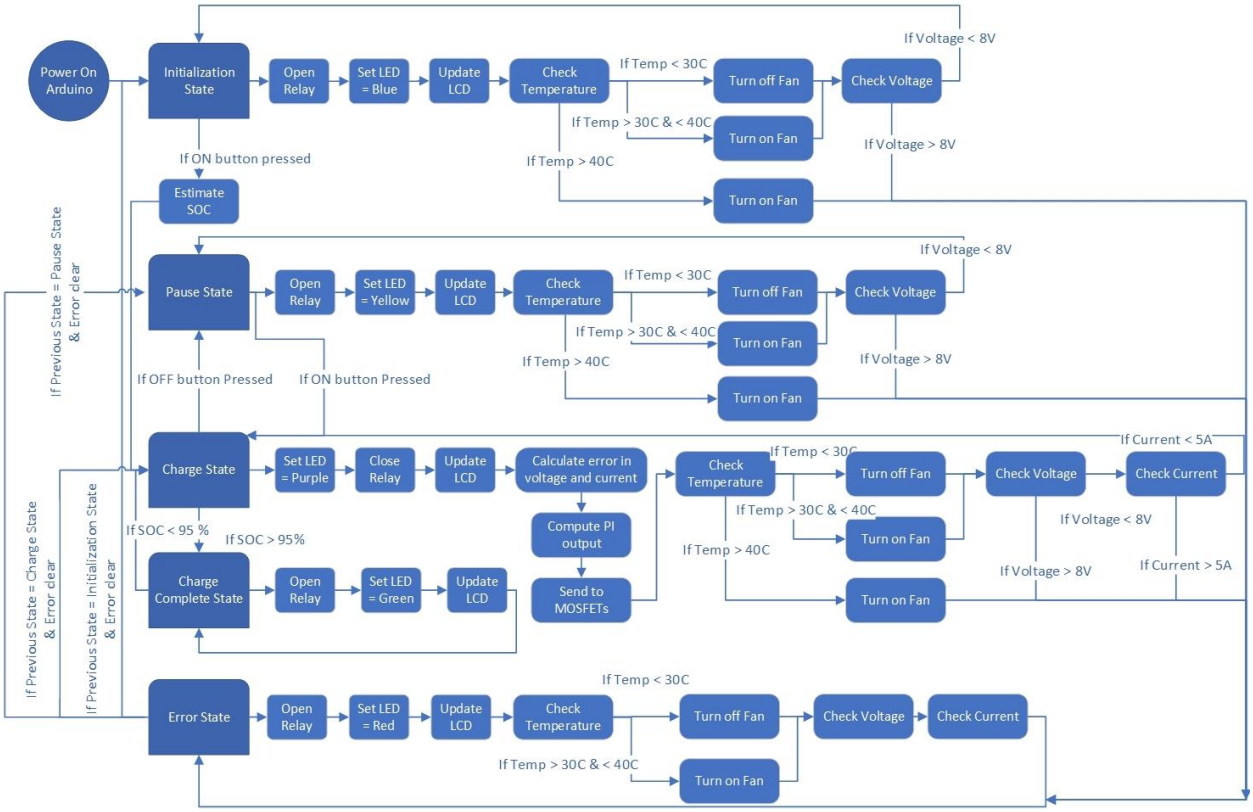


Figure 25: Software state diagram and overview

The overall structure of the software algorithm implemented in this project is modeled after a finite state machine. Five states including: initialization, charging, paused, charged, and error are defined. The switching between these states is automatically controlled by the microcontroller based on the current state, as well as input from external users and the sensors integrated with the prototype.

When within each of the states, five main functions are called upon. The first function is a temperature control function which takes input from the temperature sensors and triggers a fan if the battery cell temperature exceeds the outlined threshold. The next two functions are voltage and current sensing functions that take analog readings from the external sensors and convert them to digital values for the microcontroller to process. Next, there is a DC/DC control function that

controls the switching of the MOSFETs in the DC/DC converter. The control system programmed into the microcontroller is an accurate representation of the control system seen in the Simulink model. This controller includes three cascaded PI controllers as seen in the simulation, relying on input from the voltage and current sensors compared to the setpoints to provide the desired output. Lastly, there is a state of charge (SoC) function. The SoC function relies on two different methods to provide an accurate SoC reading to the user. Upon initialization, the voltage of the battery is read to estimate the current SoC of the battery. Following this, a coulomb counting method is used to track increases and decreases in SoC during charge/discharge operations.

It is important to note, that in order to ensure the safe operation of the hardware prototype, error checking was completed within each state. This enables the microcontroller to quickly move to the error conditions state and open the relay within the charging circuit when any parameters were outside of the predetermined SOA. A full copy of the code implemented can be found in section 7.4 Appendix D – Prototype Code.

The image below shows the complete prototype design that was built-in hardware.

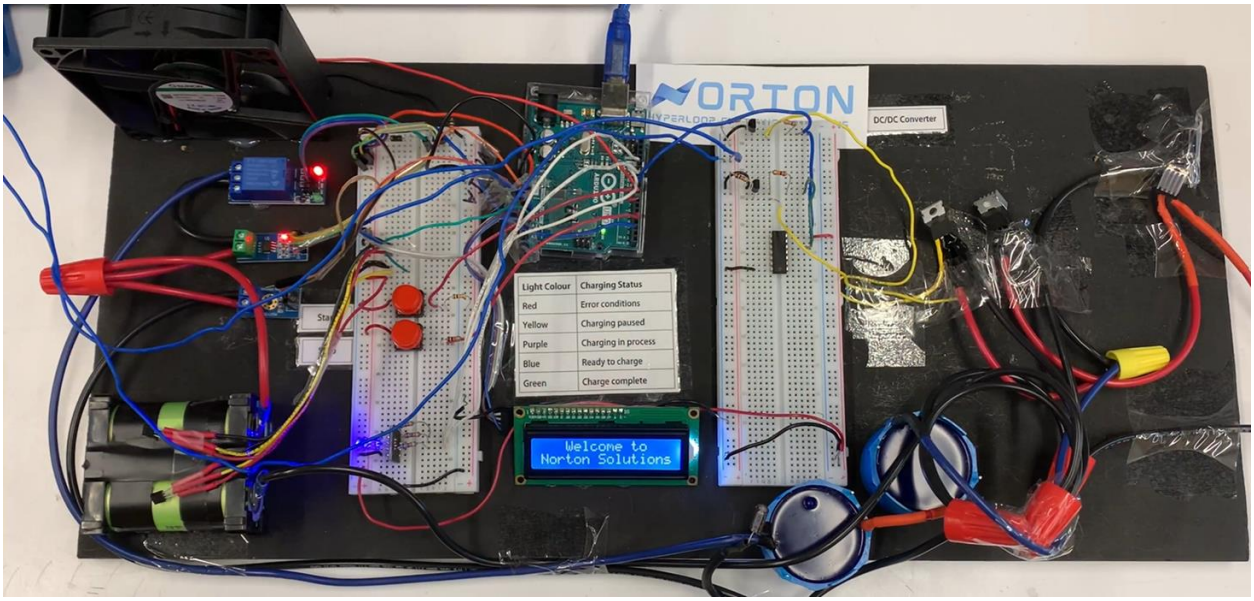


Figure 26: Hardware implemented prototype

4.2 Testing Strategy/Validation and Safety Protocols

The following test plan outlines how the prototype Hyperloop DC Fast Charging System was tested. It outlines the features and components of the design tested and the approach to testing each of them.

The overall approach to testing the scaled-down Hyperloop DC Fast Charging System began with component testing, followed by integration testing, and finally, charge cycle testing. Starting with component testing ensured that all purchased components performed as expected before assembly. Once all of the independent components were checked, they were integrated and tested to ensure proper communication with the software. Finally, the system as a whole was tested to collect data from the charging system and compared with the MATLAB simulations.

4.2.1 Phase One: Component Testing

The first part of testing ensured each component performed as intended. The detailed description of the steps performed can be seen in section 7.5.1 Phase One: Component Testing. A summary of the results can be seen in the table below.

Table 10: Results for component validation

Feature	Pass/ Fail
Measured resistor values are within a reasonable tolerance of colour code values	✓
Hall effect current sensor reads the accurate value, within a reasonable range of multimeter value	✓
The voltage sensor reads the accurate value, within a reasonable range of multimeter value	✓
LED can display all potential colours (for different charge states)	✓
The temperature sensor reads the accurate value, within a reasonable range of control value	✓
Push buttons close fully when pressed	✓
All LCD pixels are working	✓
Fan operated at expected RPM with given input	✓

All internal breadboard connections are proper	✓
Inverter logic gate inverts signals	✓
Arduino can read and write from all analog and digital pins	✓
Arduino clock cycles are fixed and accurate	✓
Wires are considered a direct connection and have negligible resistance	✓

Table 11: Results for battery validation

Results	Pass/ Fail
The internal resistance of the battery cells is what is expected (See Section 7.6 Appendix F – Battery Testing)	✓
Battery voltage at full charge is what is expected	✓
The battery is not physically damaged in any way	✓

4.2.2 Phase Two: Validation of Integration

After the individual components were validated and were operating correctly, the system was connected with tests completed to confirm the components interacted with the control system and with each other. Once completed using a breadboard, the system was soldered and rechecked to confirm the system is in working order. The steps performed can be seen in section 7.5.2 Phase Two: Validation of Integration.

Table 12: Results for AC/DC converter validation

Feature	Pass/ Fail
MOSFETs operate at a switching frequency of 5000Hz	✓
The DC voltage out of the AC/DC converter is as expected	✓
The DC current out of the AC/DC converter is as expected	✓

Table 13: Results for DC/DC converter validation

Feature	Pass/ Fail
MOSFETs operate at a switching frequency of 5000Hz	✓
The DC voltage out of the DC/DC converter is as expected	✓
The DC current out of the DC/DC converter is as expected	✓

Table 14: Results for cooling system (fan) validation

Feature	Pass/ Fail
The Fan turns on when the ambient temperature surpasses 30 degrees Celsius	✓
The Fan turns off when the ambient temperature drops below 30 degrees Celsius	✓

4.2.3 Phase Three: Validation of Entire System

After the integration testing, the whole system was tested. This ensured that communication between all components was working properly and that all of the functional and non-functional requirements were being met. The unit and integration tests were continuously tested as appropriate to ensure that combining the whole system did not cause any failures in individual systems. The steps performed can be seen in section 7.5.3 Phase Three: Validation of Entire System.

Table 15: Results for software validation

Feature	Pass/ Fail
Code uses object-oriented programming to increase efficiency	✓
Software writes progress updates to the console for debugging purposes	✓
The software can run charge/discharge cycles without error/stopping	✓

Table 16: Results for charging validation

Feature	Pass/ Fail
Battery charges to 95% in 50 minutes	✓
LED light displays 5 colours at the correct charge stages	✓
The battery can be efficiently discharged to 10%	✓

The completion of these tests validated that the prototype was correctly assembled and was safe to operate and model the simulation.

When completing each phase of validation, safety needed to be considered at all times. This began with the selection of individual components. It was ensured that each component was appropriately rated for the power requirements of the charging circuit. The wire gauge for the high-power components of the circuit was sized to be 14 AWG prior to the assembly of the components to allow for safe current flow for the battery charging circuit. These considerations allowed for the safe testing of individual components as seen in phase one of the validation process.

Next, during the second and third phases of the validation process, it was crucial that all component connections were solid and any live circuit sections were insulated. A multimeter was used to check the continuity of all connections, verifying that soldering was completed successfully. In addition, a normally open relay was included in the charging circuit. This allowed for the battery to be disconnected from the charging circuit in the event of any unsafe conditions.

Overall, these considerations allowed for the safe operation and testing of the hardware prototype.

4.3 Final Results and Validation

4.3.1 Full-Scale Simulation of Hyperloop Charger

A charging simulation of the full-scale Hyperloop charger seen in Figure 16 was run using Simulink to ensure that the model worked as intended. The large-scale battery pack was charged from 10% to 100% and the following data was obtained.

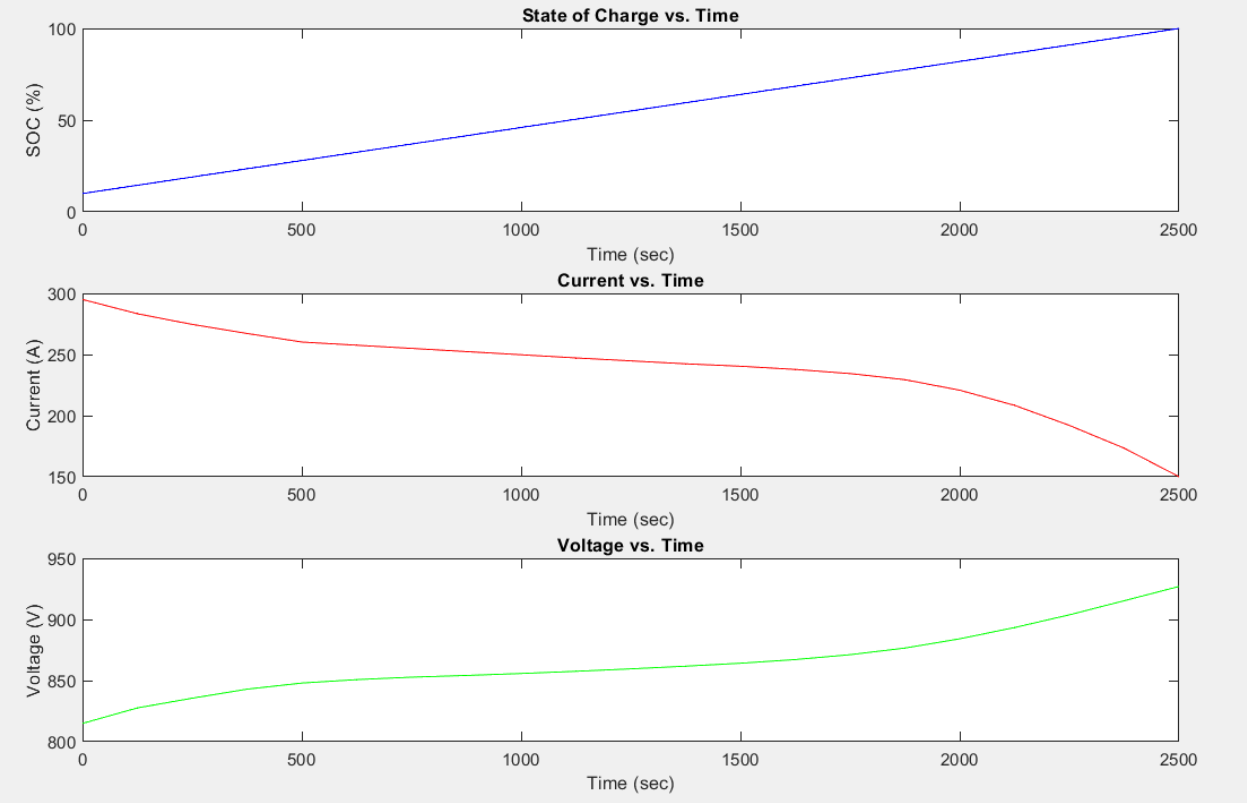


Figure 27: Full-scale Hyperloop charger simulation results

From the figure above, it can be seen that it takes approximately 42 minutes for the Hyperloop system to charge from 10% to 100% SoC at a charging current of 185A. The battery cells implemented are LiFePO4, which have a very similar charge curve to what is seen in the current and voltage plots generated from the EV charge paper model as well as the scaled-down model proving that they are behaving as expected.

This system operates at 800V and 185-740A to charge a full-scale Hyperloop battery pack. This satisfies the Hyperloop requirements set in the objectives from the Hyperloop company we received information from, to charge an 800V and 185Ah pack.

As set out in the objectives, this system should be able to satisfy the power requirements for continuous intercity round trips. To provide an example, a trip between Toronto to Montreal is explored. Each one-way trip between these cities would require 44% of the battery's charge, as seen in calculations in section 3.1.3 Power Requirement Calculations. Based on the 42-minute charge time of 10% to 100%, this charger could be used for 21 minutes at each stop to replenish the battery pack. The pod will need to be unloaded and reloaded at each end, so this charge time is reasonable to recharge the pack prior to the return trip. Also, the current used for this model is 185A, which is the minimum charging current. This could be increased to as high as 740A to further decrease the charge time by up to four times as necessary.

4.3.2 Small-Scale Simulation of Hardware Prototype

To validate the small-scale simulation (Figure 18) with the hardware prototype (Figure 26), charging data from both were collected and overlaid to be compared. Both the simulation and the hardware prototype were run at a charging current of 2.5A. The following plot shows the Hardware Prototype data vs. the Simulation results.

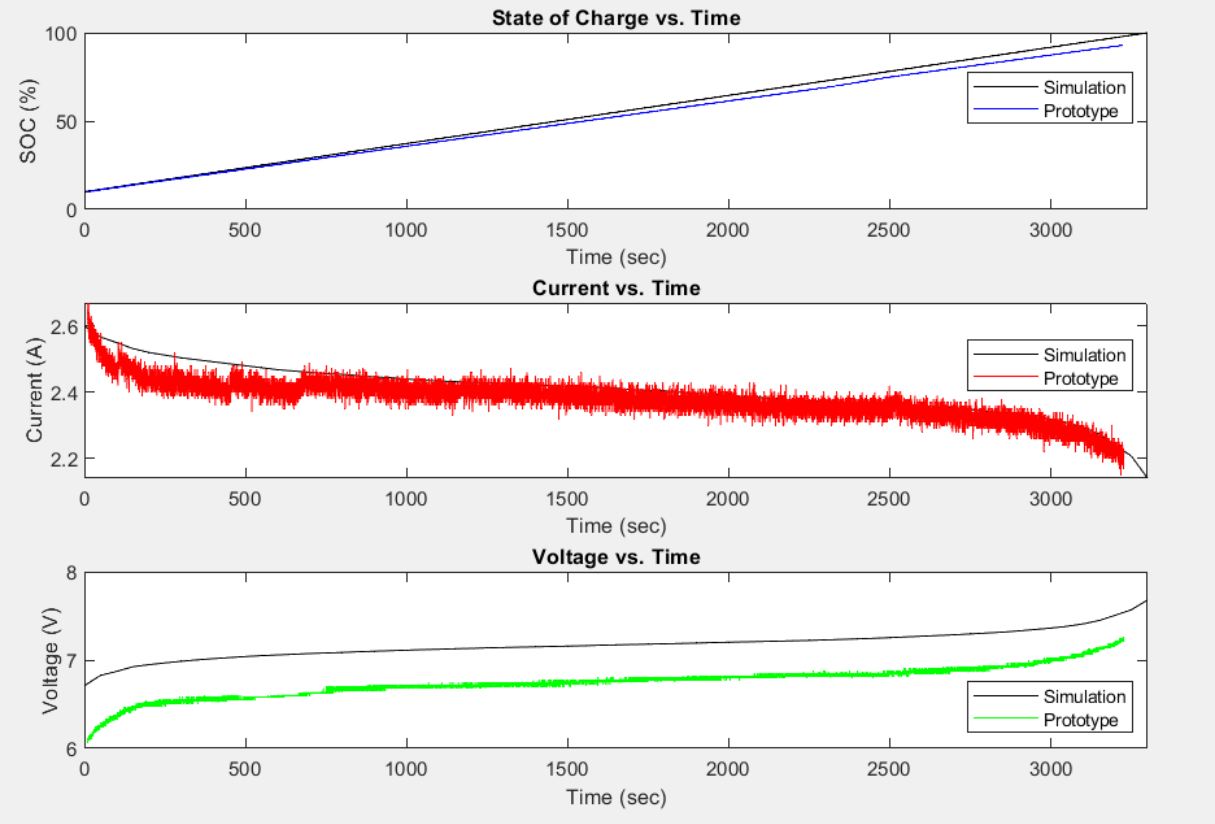


Figure 28: Small-scale Hyperloop charger simulation results overlaid with Prototype results

As seen in the plots above, the results of the hardware prototype demonstrate a consistent trend when compared to the small-scale simulation. Both plots follow the same general charge curves and display the same characteristics over the entirety of the charge cycle. Both of these also match the charge curve recommended in the battery product datasheet, as seen in section 7.1 Appendix A – Product Data Sheets.

When examining the SoC vs. time plot, it can be seen that the hardware prototype charges at nearly the same rate as the simulation. There is a slight reduction in charging speed seen in the hardware prototype. This is attributed to the slightly reduced current compared to the simulation as seen in the Current vs. Time plot. It is also important to note that the voltage seen in the hardware testing is 6% lower than in the simulations. Overall, these differences are very small and thus they can be attributed to the resistance and heat losses realized in real-world circuits that are not seen in MATLAB simulations, which assume ideal conditions.

From the above trends matching the expected simulation output, it can be concluded that the hardware prototype is validated. This is because charge curve characteristics match the simulated values within a tolerated margin of error. This satisfies the project objective of validating a real-world and scaled model of the system, which gives confidence that the design will function as expected at a variety of scales.

4.3.3 Small-Scale Simulation for Optimizing Battery Health

To understand the effects that the temperature control system (Figure 20) had on battery health, simulations were run on the charger in low and high-temperature environments. In these simulations, the batteries were fast-charged at their maximum rated charging current (10A) and discharged at their maximum rated continuous discharge current (50A). Charging at the maximum rated current minimizes charging time which helps to satisfy the project objective of designing a fast-charging system.

First, simulations were run on the small-scale charging model to understand the effects on a battery that was charged at a low temperature (-30°C) with and without a temperature control system. The temperature control system ensures that the cell temperature remains at its ideal value of 25°C [19] before charging begins and maintains this temperature throughout the charging. The test was run over 30 cycles of charge and discharge, looking at one LiFePO4 battery cell.

The following plots show the data from the low-temperature simulation compared to the controlled temperature simulation. The top plot shows the SoC of the battery over 30 cycles, and the second plot shows the degradation of the maximum battery capacity over 30 cycles.

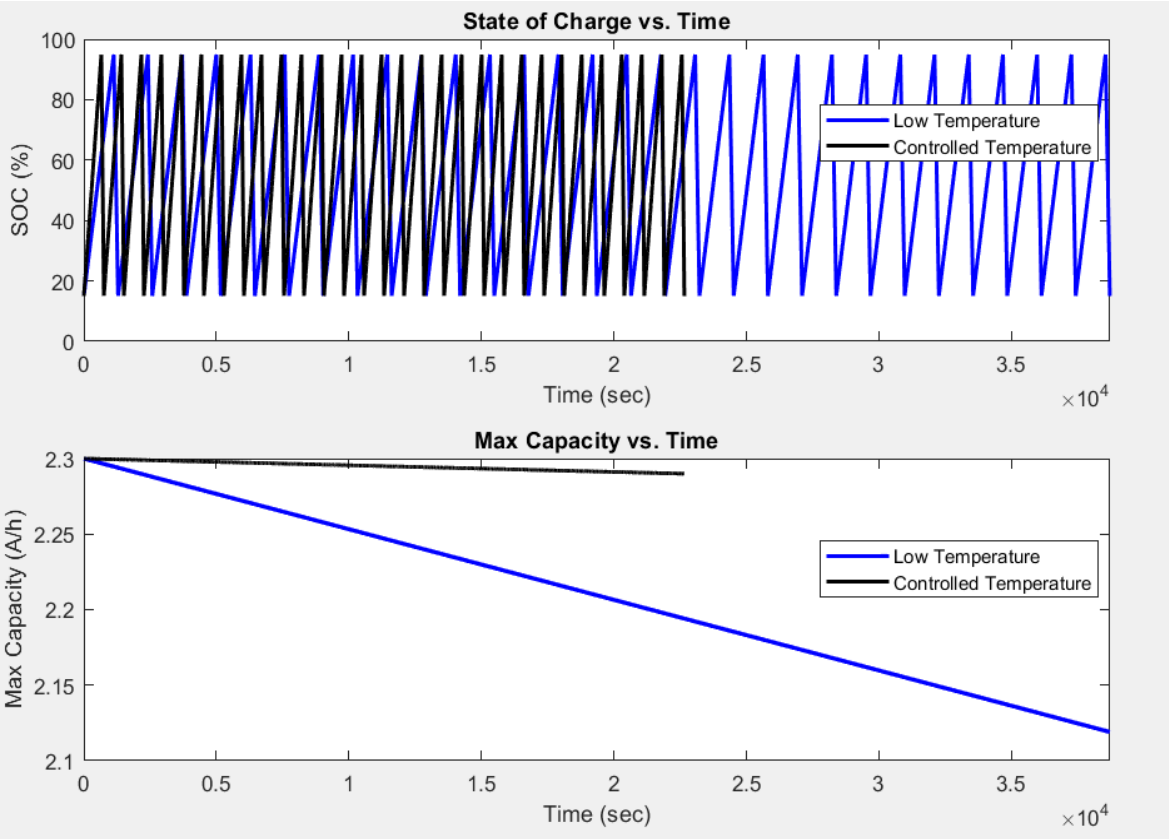


Figure 29: Low temperature vs. Controlled temperature degradation over 30 cycles

From the State of Charge vs. Time plot, it can be seen that when charging with the temperature-controlled system, it charges 42% faster than in low-temperature conditions. The second plot shows Maximum Capacity vs. Time under both conditions and it can be seen that the batteries saw greater degradation without a temperature control system implemented. Specifically, the batteries in the temperature-controlled system experienced 7.2% less degradation than in the low-temperature environment.

Next, simulations were run to understand the effects on a battery charged at a high temperature (45°C) with and without a temperature control system. The simulation was run over 30 cycles of charge and discharge looking at one LiFePO4 battery cell.

The following plots show the data from the high-temperature simulation with and without the control system. The top plot shows the SoC of the battery over 30 cycles, and the second plot shows the degradation of the maximum battery capacity over 30 cycles.

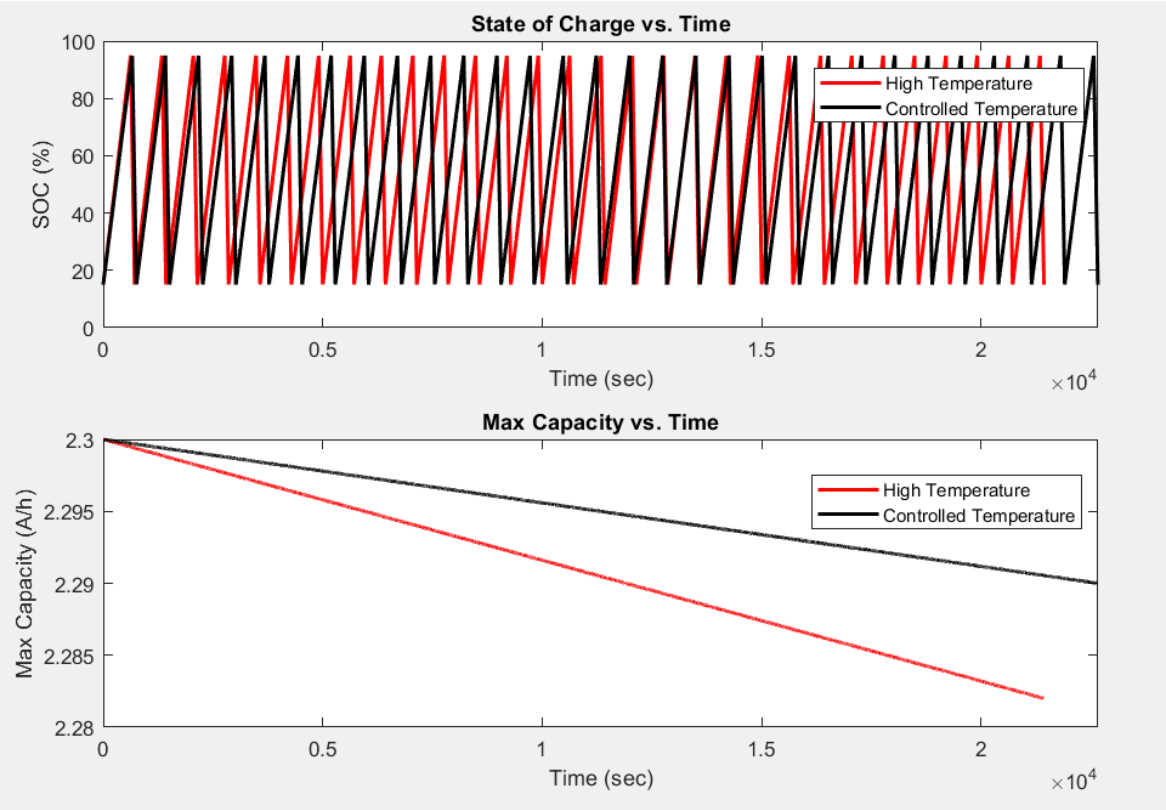


Figure 30: High temperature vs. Controlled temperature degradation over 30 cycles

From the State of Charge vs. Time plot, it can be seen that in the high-temperature environment, without the temperature control system, the battery charges 6% faster. This shows that charging and discharging are done slightly faster at higher temperatures. The second plot displays Maximum Capacity vs. Time which shows that over 30 cycles, the capacity of the batteries saw greater degradation in the high-temperature simulation than in the temperature-controlled simulation. Specifically, the simulation with temperature control experienced 3.3% less degradation in maximum capacity. Even though the high-temperature system experienced a slightly faster charge, the degradation is greater which is more important to the system as a whole.

Based on the results from the temperature simulations, the following conclusion can be drawn: maintaining batteries at their rated temperature (25°C) using a control system can reduce degradation in the maximum capacity of the batteries. This aids in satisfying the objectives of this project as implementing this temperature control system with a charger will reduce battery degradation. The control system becomes increasingly more important when working at extreme temperatures such as -30°C or 45°C, as degradation is amplified when the operating temperature is further from the ideal value. Over long periods of operation, the benefits of the temperature-controlled system will allow the batteries to have a longer useful life providing economic and environmental benefits, as they will not need to be replaced or disposed of as often. The control system also provides flexibility in the environments that Hyperloop can be operated.

5. Final Conclusions

This paper outlined the design of an efficient DC charging system for Hyperloop Application. The goal of this project was to design a charging system that satisfies Hyperloop power requirements. Also, to optimize the system, the following objectives were set: minimize charging time to reduce down-time of pods, minimize stress induced on the battery pack during charging, and validate the charging model at various scales.

To begin, an IEEE EV charging system paper was selected and verified to use as a base for the project. The system was then scaled up to Hyperloop parameters as well as scaled down to prototype parameters. The parameters were then tuned to obtain the desired output voltage and current, and modifications were made to the control systems to obtain fast response times.

The full-scale Hyperloop charger model was designed to charge an 800V, 185Ah battery pack. Performing simulations using the minimum charge current of 185A, it was determined that the system could effectively charge the battery pack in 42 minutes. The charge current of this system has the potential to be increased to as high as 740A to obtain even faster charge times. Given the expected size of the pack, it was determined that a one-way trip from Toronto to Montreal would drain 44% of the capacity of the battery pack, meaning that it would need 21 minutes to reach full charge before departing. This is a reasonable amount of time to fully unload and reload a Hyperloop pod, meaning that the designed charger satisfies the needs of a potential route that Hyperloop is expected to run.

A hardware prototype was also built based on the scaled-down model that was created in Simulink. This prototype was found to match the charge curves seen in the scaled-down system simulation when run at the same parameters (2.5A charge current). The SoC, current, and voltage plots matched the charge characteristics, with only a small voltage difference of 6% between the hardware prototype and the simulation. This was determined to be caused by the ideal conditions that were seen in the simulation which do not account for resistive and heating losses that would be seen in real life. Overall, the prototype was successful in validating the results seen in the small-scale simulation.

Finally, a temperature control system was implemented with the scaled-down Simulink model to ensure that the cells were kept at their ideal operating temperature. This model was tested in two different environments (all charging at 10A) with and without temperature control. The first was a low-temperature environment (-30°C), where it was seen that the control system decreased charge time by 42% and minimized battery degradation by 7.2%. The second was a high-temperature environment (45°C), where it was seen that the control system minimized battery degradation by 3.3%. This means that when the temperature control system was implemented, the batteries experienced a slower degradation of their maximum capacity.

From the results seen above, it was determined that the project accomplished the stated objectives. A charging circuit for Hyperloop battery pods was created, verified, and optimized across a variety of scales. Also, methods of reducing battery degradation were investigated and applied.

6. References

[1] "hyperloop-alpha.pdf." Accessed: Oct. 19, 2021. [Online]. Available: https://www.tesla.com/sites/default/files/blog_images/hyperloop-alpha.pdf
[2] N. None, "Effect of Hyperloop Technologies on the Electric Grid and Transportation Energy," DOE/EE--2328, 1773025, 8644, Jan. 2021. doi: 10.2172/1773025.
[3] "Transportation sector energy consumption," p. 11, 2016.
[4] "Toronto to Montreal in 39 minutes? Futuristic people mover zips to next stage | CBC News." https://www.cbc.ca/news/canada/ottawa/hyperloop-toronto-ottawa-montreal-route-winner-1.4291893 (accessed Apr. 15, 2022).
[5] C. Timperio, "Linear Induction Motor (LIM) for Hyperloop Pod Prototypes," Jul. 2018, doi: 10.3929/ethz-b-000379531.
[6] P. Saraf, N. Singh, P. Kattamuri, J. Karhade, and I. Bhattacharya, *On-board Electrical, Electronics and Pose Estimation System for Hyperloop Pod Design*. 2020. doi: 10.36227/techrxiv.13345376.
[7] "projects," *rLoop | Global, Crowdsourced Engineering*. https://www.rloop.org/projects (accessed Dec. 05, 2021).
[8] "GT HyperJackets PDB.pdf." Accessed: Dec. 05, 2021. [Online]. Available: https://hyperjackets.gatech.edu/resources/GT%20HyperJackets%20PDB.pdf
[9] "hyperloop.pdf." Accessed: Dec. 05, 2021. [Online]. Available: https://web.ece.ucsb.edu/Faculty/Johnson/ECE189/final/hyperloop.pdf
[10] "What Are The Different Levels Of Electric Vehicle Charging? - Forbes Wheels." https://www.forbes.com/wheels/advice/ev-charging-levels/ (accessed Dec. 05, 2021).
[11] D. Ronanki, A. Kelkar, and S. Williamson, "Extreme Fast Charging Technology—Prospects to Enhance Sustainable Electric Transportation," *Energies*, vol. 12, p. 3721, Sep. 2019, doi: 10.3390/en12193721.
[12] "Battery Management Systems for Large Lithium-Ion Battery Packs by Davide Andrea | Technical Books Pdf." https://www.technicalbookspdf.com/battery-management-systems-for-large-lithium-ion-battery-packs-by-davide-andrea/ (accessed Oct. 19, 2021).
[13] V. Castiglia, P. Livreri, R. Miceli, F. R. Galluzzo, G. Santelia, and G. Schettino, "Design and simulation of a fast DC recharging station for EV," in *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, Nov. 2017, pp. 1198–1203. doi: 10.1109/ICRERA.2017.8191243.
[14] A. Arancibia and K. Strunz, "Modeling of an electric vehicle charging station for fast DC charging," in *2012 IEEE International Electric Vehicle Conference*, Mar. 2012, pp. 1–6. doi: 10.1109/IEVC.2012.6183232.
[15] A. Sharma and R. Gupta, "Bharat DC001 Charging standard Based EV Fast Charger," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2020, pp. 3588–3593. doi: 10.1109/IECON43393.2020.9254949.
[16] K. Drobnic *et al.*, "An Output Ripple-Free Fast Charger for Electric Vehicles Based on Grid-Tied Modular Three-Phase Interleaved Converters," *IEEE Transactions on Industry Applications*, vol. 55, no. 6, pp. 6102–6114, Nov. 2019, doi: 10.1109/TIA.2019.2934082.
[17] "BU-808: How to Prolong Lithium-based Batteries," *Battery University*, Sep. 11, 2010. https://batteryuniversity.com/article/bu-808-how-to-prolong-lithium-based-batteries (accessed Apr. 15, 2022).
[18] "BU-802a: How does Rising Internal Resistance affect Performance?," *Battery University*, Sep. 27, 2010. https://batteryuniversity.com/article/bu-802a-how-does-rising-internal-resistance-affect-performance (accessed Apr. 15, 2022).
[19] "26650_datasheet.pdf." Accessed: Apr. 15, 2022. [Online]. Available: https://www.batteryspace.com/prod-specs/6610.pdf

7. Appendices

7.1 Appendix A - Product Data Sheets

Table 17: Product datasheets and links

Part	Link to Product Data Sheet
Battery Charging (power supply, DC/DC)	
DC power supply	https://assets.testequity.com/te1/Documents/pdf/bk/1670A_manual.pdf
LiFePO4 Batteries	https://www.batteryspace.com/prod-specs/6610.pdf
Diode	https://www.vishay.com/docs/88721/sb520.pdf
Inverter logic gate	https://rocelec.widen.net/view/pdf/qn8lxyyo0s/schs127d.pdf?t.download=true&u=5oefqw
BJT	https://media.digikey.com/pdf/Data%20Sheets/ON%20Semiconductor%20PDFs/BC546-50.pdf
MOSFET	https://www.infineon.com/dgdl/irf520npbf.pdf?fileId=5546d462533600a4015355e340711985
Control	
Arduino	https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf
Sensors	
Hall-effect current sensor	https://www.amazon.ca/COVVY-Current-Arduino-ACS712ELC-20A-Indicator/dp/B07TQ5M9MP/ref=sr_1_9?crid=3HUVXO4T1U8EN&keywords=current+sensor&qid=1643685181&sprefix=current+sensor%2Caps%2C79&sr=8-9
Voltage sensor	https://www.amazon.ca/Voltage-Measurement-Detection-Arduino-Geekstory/dp/B07FVVSYYH/ref=sr_1_5?crid=2WPAGNLH3I90&keywords=voltage+sensor&qid=1642988697&sprefix=voltage+sensor%2Caps%2C106&sr=8-5
Temperature sensor	https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf
Pushbuttons	N/A
Actuators	
Tri-colour status LED	https://www.nteinc.com/specs/30100to30199/pdf/nte30115.pdf

LCD and I2C converter	https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf
Fan	https://www.cuidevices.com/product/resource/cfm-40-series.pdf
Relay	https://www.handsontec.com/dataspecs/4Ch-relay.pdf
BJT	https://www.farnell.com/datasheets/2287718.pdf
Diode	https://www.farnell.com/datasheets/1689669.pdf?_ga=2.235854722.2094122556.1643571113-1312331928.1642692303&_gac=1.95394926.1643579113.Cj0KCQiAi9mPBhCJARIsAHchl1wQV1LjtXcTVN97GBNsYStGhCU4C2InzT9VwyGSFWAAHvU-DZv4_s4aAleTEALw_wcB
Other	
14 gauge wire, black	N/A
14 gauge wire, red	N/A
14 gauge wire, blue	N/A
Battery holding units	N/A
Bus bar for batteries	N/A
Heat sink	N/A
Breadboard	N/A
Solder	N/A
Heat shrink	N/A
Screw-on wire connectors	N/A
Circuit Components	
Lbat	https://product.tdk.com/system/files/dam/doc/product/transformer/transformer/pfc-choke/30/db/ind_2008/b82615.pdf
Cbat	https://www.chemi-con.co.jp/products/relatedfiles/capacitor/catalog/KYBLL-e.PDF
Cdc	https://www.rubycon.co.jp/wp-content/uploads/catalog-aluminum/AX.pdf
R (into DC/DC)	https://www.te.com/commerce/DocumentDelivery/DDEController?Action=srchrtrv&DocNm=1773180&DocType=DS&DocLang=English

As the battery cells are a critical component, in addition to being included in the above table, the product datasheet is included below [19].

+ Nanophosphate® High Power Lithium Ion Cell
ANR26650m1-B

A123 SYSTEMS

A123's high-performance Nanophosphate® lithium iron phosphate (LiFePO4) battery technology delivers high power and energy density combined with excellent safety performance and extensive life cycling in a lighter weight, more compact package. Our cells have low capacity loss and impedance growth over time as well as high usable energy over a wide state of charge (SOC) range, allowing our systems to meet end-of-life power and energy requirements with minimal pack oversizing.

APPLICATIONS

COMMERCIAL SOLUTIONS
Advanced lead acid replacement batteries for:
+ Datacenter UPS
+ Telecom backup
+ IT backup
+ Autonomously guided vehicles (AGVs)
+ Industrial robotics and material handling equipment
+ Medical devices

GOVERNMENT SOLUTIONS
+ Military vehicles
+ Military power grids
+ Soldier power
+ Directed energy

GRID SOLUTIONS
Versatile, flexible and proven storage solutions for the grid:
+ Frequency regulation
+ Renewables integration
+ Reserve capacity
+ Transmission and distribution

TRANSPORTATION SOLUTIONS
Hybrid, plug-in hybrid and electric vehicle battery systems for:
+ Commercial vehicles
+ Off-highway vehicles
+ Passenger vehicles

ANR26650m1-B TECHNICAL DATA

Cell Dimensions	ø26 x 65 mm
Cell Weight	76g
Cell Capacity (nominal/minimum) (0.5C Rate)	2.5/2.4 Ah
Voltage (nominal)	3.3V
Internal Impedance (1kHz AC typical)	6mΩ
Power*	2600 W/kg
Recommended Standard Charge Method	2.5A to 3.6V CCCV, 60 min
Recommended Fast Charge Method to 80% SOC	10A to 3.6V CC, 12 min
Maximum Continuous Discharge	50A
Maximum Pulse Discharge (10 seconds)	120A
Cycle Life at 20A Discharge, 100% DOD	>1,000 cycles
Operating Temperature	-30°C to 55°C
Storage Temperature	-40°C to 60°C

** ~200W as measured by A123 modified HPPC Method @ 23°C, 50% SOC, 10 second discharge*

www.a123systems.com

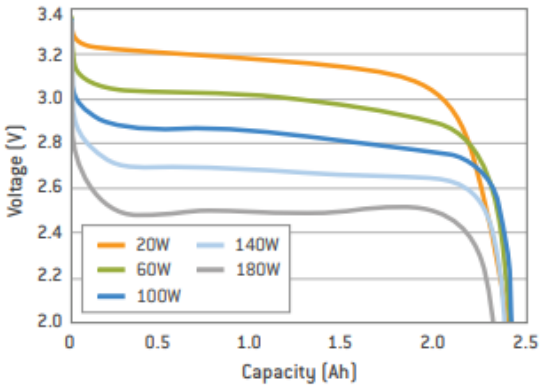
MD100113-02

Figure 31: Datasheet of ANR26650 battery cell (page 1)

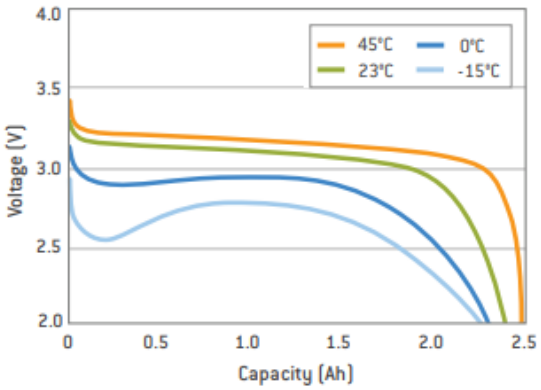
+ Nanophosphate® High Power Lithium Ion Cell ANR26650m1-B

A123 SYSTEMS

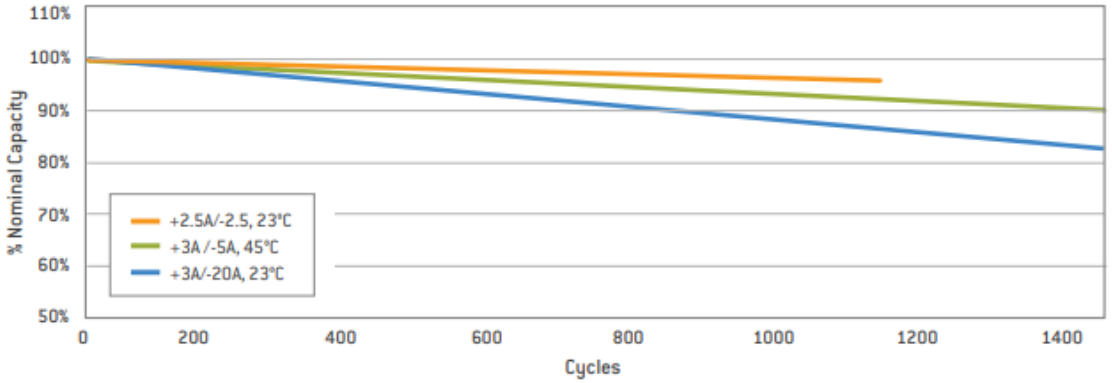
Constant Power Discharge Characteristics at 23°C



2.5C Constant Current Discharge at Various Temperatures



Cycle Life Performance, 100% DOD, Various Temperatures and Discharge Rates



This document represents typical data. Performance may vary depending on use conditions and application.
A123 Systems makes no warranty explicit or implied with this data sheet. Contents subject to change without notice.

CORPORATE HEADQUARTERS
A123 Systems, Inc.
200 West Street
Waltham, MA 02451

www.a123systems.com

POWER SAFETY LIFE ENERGY

MD100113-02

Figure 32: Datasheet of ANR26650 battery cell (page 2)

7.2 Appendix B - Complete Hardware Cost Breakdown

Table 18: Complete hardware cost breakdown

Part	Part Number	Quantity Ordered	Price/unit[1]	Total price[2]
Battery Charging (power supply, DC/DC)				
DC power supply	1608A	0	0.00	0.00
LiFePO4 Batteries	ANR 26650 M1B	5	13.50	127.37
Diode	SB520-E3/54	10	0.76	7.58
Inverter logic gate	CD74HCU04E	1	0.90	0.90
BJT	BC548CTA	10	0.44	4.40
MOSFET	IRF520NPBF	10	1.37	13.7
Control				
Arduino	SMD R3 ATMEGA328	1	30.14	30.14
Sensors				
Hall-effect current sensor	ACS712	2	3.00	18.99[3]
Voltage sensor	GR0430X10	10	N/A	17.76
Temperature sensor	TMP36GT9Z	5	2.52	12.60
Pushbuttons	N/A	0	0.00	0.00
Actuators				
Tri-colour status LED	N/A	0	0.00	0.00
LCD and I2C converter	N/A	0	0.00	0.00
Fan	N/A	0	0.00	0.00
Relay	CA-MW-CB063	5	N/A	12.99
BJT	PN2222ABU	1	0.51	0.51
Diode	1N4007	1	0.30	0.30
Other				
14 gauge wire, black	N/A	10ft	0.82	8.20
14 gauge wire, red	N/A	10ft	0.82	8.20
14 gauge wire, blue	N/A	10ft	0.82	8.20
Battery holding units	N/A	50	N/A	22.66

Bus bar for batteries	N/A	10m	N/A	27.65
Heat sink	N/A	0	0.00	0.00
Breadboard	N/A	0	0.00	0.00
Solder	N/A	0	0.00	0.00
Heat shrink	A580	580pc assorted	0.00	0.00
Screw-on wire connectors	25-AWC	25pc assorted	N/A	9.84
Circuit Components				
Lbat	B82615B2302M001	4	13.54	54.16
Cbat	EKYB101ELL6R8ME11D	3	0.52	1.56
Cdc	400AX24MEFC12.5X16	3	1.45	4.35
R (into DC/DC)	MPT100T1R5F	1	26.27	26.27
TOTAL COST				**434.32**

1 - Components that display N/A in the price/unit column were purchased in a bulk pack.

2 - Components that have $0 listed as the cost was previously owned and thus didn't affect the project budget.

3 - A different sensor was ordered, which accounts for the additional total cost.

7.3 Appendix C - Full Prototype Circuit Including Three-Phase Supply

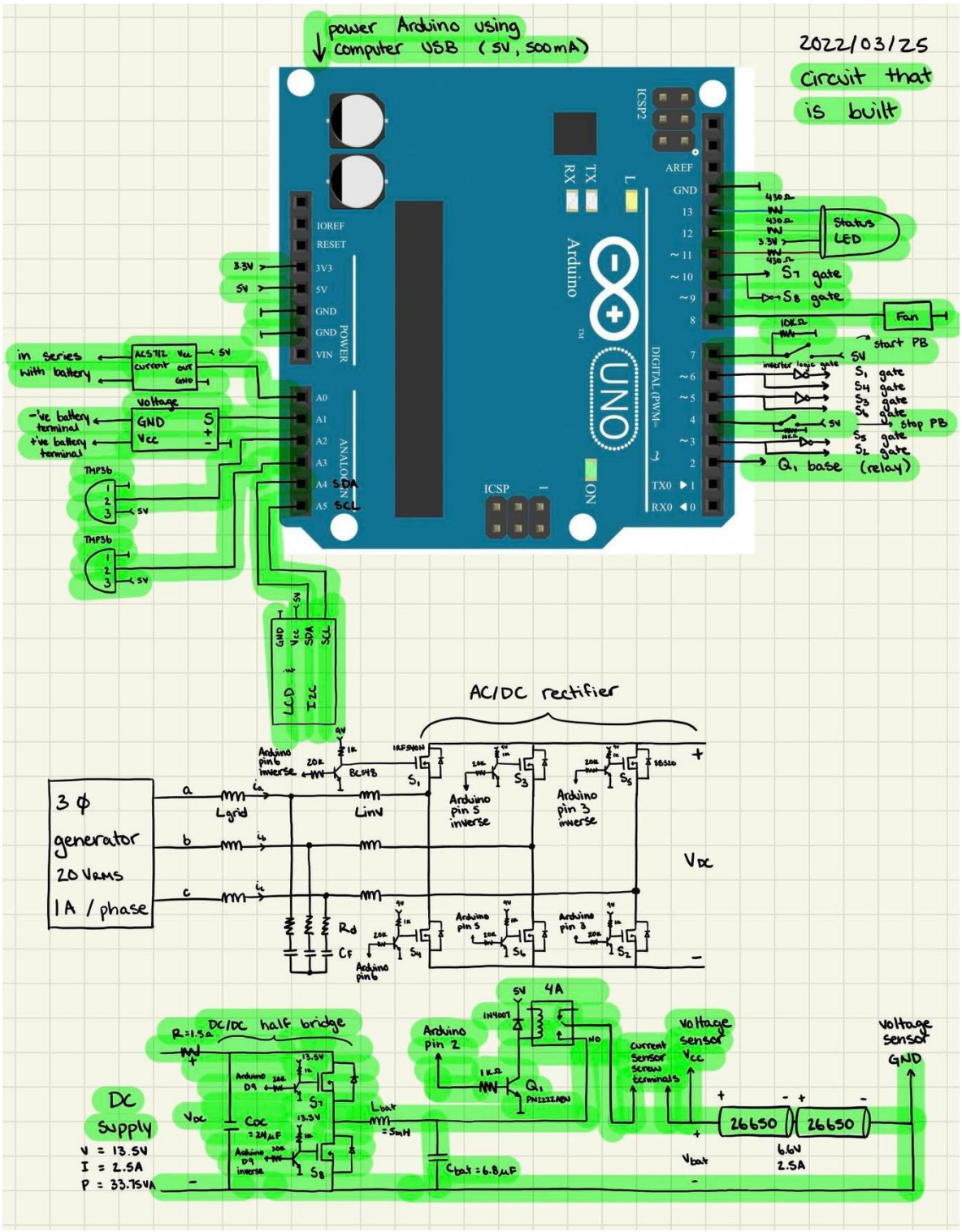


Figure 33: Full prototype circuit (including three-phase supply)

7.4 Appendix D - Prototype Code

```
/*
  Efficient DC Charging System for Hyperloop Application
  Arduino code for charging system prototype
*/

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 4);

//  Define variables
float maxVoltage = 8.0, maxCurrent = 5.0, minCurrent = -5;
float currentADC = 0.0, Samples = 0.0, avgADC = 0.0, currentValue = 0.0, voltageValue = 0.0;
float temp = 0.0, sense1Voltage = 0.0, sense2Voltage = 0.0, reading1 = 0.0, reading2 = 0.0;
float error1 = 0.0, error2 = 0.0, cumulError1 = 0.0, cumulError2 = 0.0, cumulError3 = 0.0;
float kp1, ki1, kp2, ki2, kp3, ki3;
float piout1, piout2, piout3;
float Setpoint = 7.2;
float totalCoulombs, totalAh, SOC = 0;

unsigned long currentTime, previousTime, elapsedTime, startMillis;
bool sw = 0;                        // 0 = off, 1 = on
int currentState  = 0;              // 0 = initialize, 1 = charging, 2 = done charging, 3 = error
int previousState = 0;

// Assign pins for external sensors and actuators
int temp1Pin          = A2;
int temp2Pin          = A3;
int fan               = 8;
int currentPin        = A0;
int voltagePin        = A1;
int relay             = 2;
int mf1               = 9;
int redLED            = 11;
int greenLED          = 12;
int blueLED           = 13;
const int onButton    = 7;
const int offButton   = 4;

void setup() {
  // Initialize digital input/outputs from microcontroller
  Serial.begin(9600);
  pinMode(relay, OUTPUT);
  pinMode(onButton, INPUT);
```

```
  pinMode(offButton, INPUT);
  pinMode(redLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(blueLED, OUTPUT);
  pinMode(fan, OUTPUT);
  pinMode(mf1, OUTPUT);
  digitalWrite(relay, LOW);                 //set relay to low as default state to ensure safe operation
  ledFunction();

  // Setup initial message on LCD.
  lcd.init();
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(3, 0);
  lcd.print("Welcome to");
  lcd.setCursor(0, 1);
  lcd.print("Norton Solutions");
  delay(2000);
}

void loop() {
  if (digitalRead(onButton) == HIGH) {      // Check if ON button is pressed
    sw = 1;
  }
  if (digitalRead(offButton) == HIGH) {     // Check if OFF button is pressed
    sw = 0;
  }

  // Loop in off state while switch is off ie. intialize state if previous state was zero or pause state if previous state
was anything else
  while (sw == 0) {
    digitalWrite(relay, LOW);               // Open relay
    if (digitalRead(onButton) == HIGH) {    // Check if ON button is pressed
      sw = 1;
    }
    if (digitalRead(offButton) == HIGH) {   // Check if OFF button is pressed
      sw = 0;
    }

    // Call functions for intialization state
    currentState = 0;
    ledFunction();
    coolingFunction();
    voltageFunction();

    //display temperature, voltage, current, and SOC on the serial monitor
```

```
  Serial.print(currentTime);
  Serial.print(", ");
  Serial.print(temp);
  Serial.print(", ");
  Serial.print(currentValue);
  Serial.print(", ");
  Serial.print(voltageValue);
  Serial.print(", ");
  Serial.println(SOC);
 }
 lcd.clear();
 voltageFunction();
 currentFunction();

 // Error state, automatically sets previous state to 1 so it does not revert to initialize during error conditions
 while ((voltageValue > maxVoltage) || (currentValue > maxCurrent) || (currentValue < minCurrent) || (temp >
24.0)) {                          // While there is an error condition
  currentState  = 3;               // Set current state to error
  previousState = 1;               // Set previous state to charging
  digitalWrite(relay, LOW);        // Open relay

  ledFunction();                   // Display appropriate LED colour based on state

  // Institute a delay so that error can be cleared prior to user attempting to resume charging
  for (int i = 0; i < 3000; i++) {
   delay(3);
   if (digitalRead(onButton) == HIGH) {       // Check if ON button is pressed
    sw = 1;
   }
   if (digitalRead(offButton) == HIGH) {       // Check if OFF button is pressed
    sw = 0;
   }
  }

  // Call functions for error state
  coolingFunction();
  voltageFunction();
  currentFunction();
 }

 // Charging state, if previous state was 0 then it uses voltage estimation to determine starting SOC
 while ( (sw == 1) && (voltageValue < maxVoltage) && (currentValue < maxCurrent) && (currentValue >
minCurrent) && (temp < 24.0) && (SOC < 95)) {         // While ON, no errors, and not complete
  currentState = 1;
  ledFunction();                                     // Display state using LEDs
  digitalWrite(relay, HIGH);                         // Close relay
```

```
  delay(3);

  if (digitalRead(onButton) == HIGH) {                    // Check if ON button is pressed
   sw = 1;
  }
  if (digitalRead(offButton) == HIGH) {                   // Check if OFF button is pressed
   sw = 0;
  }

  // Call functions necessary for charging
  dcdcFunction(voltageValue, currentValue);
  coolingFunction();
  voltageFunction();
  currentFunction();

  // Display battery paramters to user on serial monitor
  Serial.print(currentTime);
  Serial.print(", ");
  Serial.print(temp);
  Serial.print(", ");
  Serial.print(currentValue);
  Serial.print(", ");
  Serial.print(voltageValue);
  Serial.print(", ");
  Serial.println(SOC);

  if (previousState == 0) {
   SOCestimate(); // Estimates SOC based on initial battery voltage
   totalCoulombs = ((SOC / 100) * 2.5 * 3600); // Sets coulombs value to associated SOC to enable coloumb
counting as SOC tracking method
  }
  previousState = 1;                    // Set previous state to 1 to indicate charging had commenced
 }

 // Charging continues until SOC is 95, assuming no error or external input
 while (SOC > 95) {
  currentState = 2;
  digitalWrite(relay, LOW);             // Open relay
  ledFunction();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Charge Complete");
  delay(3);
 }

 delay(3);
```

```
 coolingFunction();
}

void coolingFunction() {
 // This function reads the temperature sensors and will turn on the fan if required
 reading1 = analogRead(temp1Pin) ;          // Read analog value from sensor
 reading2 = analogRead(temp2Pin);           // Read analog value from sensor
 sense1Voltage = (reading1 * 3.3) / 1024;   // Convert value to mV
 sense2Voltage = (reading2 * 3.3) / 1024;   // Convert value to mV
 temp = ((sense1Voltage - 0.25) * 100) + ((sense2Voltage - 0.25) * 100) / 2; // Compute average of sensors (500mV
offset for -50degrees and 10mV per degree)

 if (temp >= 30) {
  digitalWrite(fan, HIGH);                  // Turn ON fan if over temperature
 }
 else {
  digitalWrite(fan, LOW);                   // Turn OFF fan if not over temperature
 }
 delay(100);
}

void currentFunction() {
 // This function reads the current sensor, averages the value to reduce error, updates the LCD, and performs
coulomb counting
 for (int x = 0; x < 10; x++) { //Get 10 samples
  currentADC = ((analogRead(currentPin)) * (5.0 / 1024));      // Read current sensor values and convert to mV
  Samples = Samples + currentADC;                              // Add samples together
  delay (3);                                                   // Let ADC settle before next sample 2ms
 }

 avgADC = Samples / 10.0; // Compute average of samples
 currentADC = 0;
 Samples    = 0;
 // 2.5 is offset at 5V
 // 0.185v(185mV) is rise in output voltage when 1A current flows at input
 currentValue = (((avgADC - 2.5 ) / 0.185));
 // Serial.println (currentValue);

 if (currentState != 3) {
  lcd.setCursor(0, 0);
  lcd.print("I=");
  lcd.print(currentValue);
  lcd.setCursor(7, 0);
  lcd.print("       ");
 }
```

```
 currentTime = millis();  // Get the current "time" (actually the number of milliseconds since the program started)
 if (currentTime - startMillis >= 2000) {          // Test whether the period has elapsed
  totalCoulombs = totalCoulombs + currentValue * 2;
  totalAh = totalCoulombs / 3600.0;
  SOC = (totalAh / 2.50) * 100;                    // Check if max Ah of battery is correct here
                                                   // SOH = (max Ah of battery / new batt max AH)*100;
  startMillis = currentTime;                       // IMPORTANT to save the start time of the current state.
 }
 if (currentState != 3) {
  lcd.setCursor(8, 1);
  lcd.print("SOC=");
  lcd.print(SOC);
 }
}

void voltageFunction() {
 // This function reads the voltage sensor and updates the LCD
 voltageValue = (((analogRead(voltagePin)) * (5.0 / 1024)) * 5) - 0.16; // Voltage sensor is a voltage divider sensing
up to 5x terminal voltage
 if (currentState != 3) {
  lcd.setCursor(0, 1);
  lcd.print("V=");
  lcd.print(voltageValue);
 }
}

void dcdcFunction(float volt, float crnt) {
 // This function calculates the error from the PI controllers and writes to the MOSFETs
 kp1 = 0.4;
 ki1 = 1;
 currentTime = millis();                                  // Get current time
 elapsedTime = (double)(currentTime - previousTime);      // Compute time elapsed from previous computation
 error1 = 7.2 - volt;                                     // Determine error
 cumulError1 += error1 * elapsedTime;                     // Compute integral
 piout1 = kp1 * error1 + ki1 * cumulError1;
 piout1 = bound(piout1, 0.0, 2.5);

 kp2 = 5;
 ki2 = 100;
 error2 = piout1 - crnt;
 cumulError2 += error2 * elapsedTime;                     // Compute integral
 piout2 = kp2 * error2 + ki2 * cumulError2;
 piout2 = bound(piout2, 0.0, 1.0);

 kp3 = 0.4;
 ki3 = 1;
```

```
 cumulError3 += piout2 * elapsedTime;                    // Compute integral
 piout3 = kp3 * piout2 + ki3 * cumulError3;
 piout3 = bound(piout3, 0.0, 1.0)
 piout3 = map(piout3, 0.0, 1.0, 0.0, 255.0);

 analogWrite(mf1, piout3); // Write to PWM pin
}
float bound(float x, float x_min, float x_max) {          // Used to set the output bounds of the PI
controllers
 if (x < x_min) {
  x = x_min;
 }
 if (x > x_max) {
  x = x_max;
 }
 return x;
}
void ledFunction() {
 // This function sets the tri-colour LED based on the current state and updates the LCD
 if (currentState == 0 && previousState == 0) {           // Initialize State
  digitalWrite(blueLED, HIGH);
  digitalWrite(redLED,   LOW);
  digitalWrite(greenLED, LOW);

  lcd.clear();
  lcd.setCursor(2, 0);
  lcd.print("Please Push");
  lcd.setCursor(1, 1);
  lcd.print("start to begin");
 }
 if (currentState == 0 && previousState == 1) {           // Paused State
  digitalWrite(blueLED,   LOW);
  digitalWrite(redLED,   HIGH);
  digitalWrite(greenLED, HIGH);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Charging Paused");
 }
 if (currentState == 1) {                                // Charging state
  digitalWrite(blueLED, HIGH);
  digitalWrite(redLED,  HIGH);
  digitalWrite(greenLED, LOW);
 }
 if (currentState == 2) {                                // Charging complete state
  digitalWrite(blueLED,   LOW);
```

```
   digitalWrite(redLED,    LOW);
   digitalWrite(greenLED, HIGH);
  }
  if (currentState == 3) {                                    // Error state
   digitalWrite(blueLED,  LOW);
   digitalWrite(redLED,  HIGH);
   digitalWrite(greenLED, LOW);

   lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("Error Conditions");
  }
}
void SOCestimate() {
  // This function reads the voltage sensor and estimates the SOC of the battery at the start of charging
  // Limits are set based on the charge curve for the batteries in the data sheet
  if (voltageValue > 6.8) {
   SOC = 90;
  }
  else if ((voltageValue > 6.6) && (voltageValue < 6.8)) {
   SOC = 80;
  }
  else if ((voltageValue > 6.4) && (voltageValue < 6.6)) {
   SOC = 50;
  }
  else if ((voltageValue > 6.2) && (voltageValue < 6.4)) {
   SOC = 20;
  }
  else if (voltageValue < 6.2) {
   SOC = 10;
  }
}
// End of Arduino Code
```

7.5 Appendix E - Test Plan

7.5.1 Phase One: Component Testing

This phase of testing is to confirm that all components work as intended. This section will outline the criteria that each feature was tested on.

Table 19: Pass/Fail criteria for phase one testing

Feature	Testing Procedure	Criteria
Component Validation		
Measured resistor values are within a reasonable range of colour code values	Using a multimeter measure the resistance across each resistor	The measured resistor values must be within 5% of the expected value
Hall effect current sensor reads the accurate value, within reasonable a range of multimeter value	Using a current supply test the reading of the Hall effect sensor as well as the reading through a multimeter	The hall effect current sensor reading should be within 5% of the multimeter value
The voltage sensor reads the accurate value, within a reasonable range of multimeter value	Using a voltage supply test the reading of the voltage sensor as well as the reading through a multimeter	The voltage sensor reading should be within 5% of the multimeter value
LED can display all potential colours (for different charge states)	Connect tri-colour LED to Arduino and run code to display different colours	The LED should display all colours correctly
The temperature sensor reads the accurate value, within a reasonable range of control value	Using a heat gun, test the reading of the temperature sensor as well as the reading through an external thermometer	The temperature sensor reading should be within 5% of the thermometer value
Push buttons close fully when pressed	Use a multimeter set to continuity mode on the button, the multimeter will beep when the button pressed	The button should always close fully causing the multimeter to detect a short
All LCD pixels are working	Connect LCD to Arduino and run code to turn all pixels on	LCD should display all pixels on, no pixels should be dead
Fan operated at expected RPM with given input	Connect the fan to a DC supply that will output the voltage used in the final design, measure the RPM	The fans RPM should be within 5% of the rated RPM

All internal breadboard connections are proper	Use a multimeter set to continuity mode on breadboard connections, the multimeter will beep if it is a common point	All breadboard connections are connected as they should be and confirmed through the multimeter
Inverter logic gate inverts signals	Set the Arduino to output a series of high and low signals, measure the output using an oscilloscope and compare the input and output. Also, test switching speed to ensure the inverter can switch at the required duty cycle	The output from the signal inverter should be the opposite of the input (high → low, low → high)
Arduino can read and write from all analog and digital pins	Set the Arduino to write from all available pins, check using an oscilloscope, then set Arduino to read from all pins and input a signal from a waveform generator	The measured outputs from all pins should be the same as the Arduino is programmed to do. The inputs read by the Arduino should be the same as what is outputted by the generator
Arduino clock cycles are fixed and accurate	Write the output of the clock to an Arduino pin, use an oscilloscope to view the waveform and measure its accuracy	The clock cycles should be consistent and correspond with the given frequency of the Arduino clock
Wires are considered a direct connection and have negligible resistance	Use a multimeter set to continuity mode on all wires, the multimeter will beep if it is a common point	All wires are connected and confirmed through the multimeter
Battery Validation		
The internal resistance of the battery cells is what is expected	Use an internal resistance meter to measure the internal resistance of each battery	The internal resistance should be within 5% of the rated internal resistance
Battery voltage at full charge is what is expected	Charge the battery fully and measure voltage output from the battery	Battery voltage at full charge should be within 5% of the rated value
The battery is not physically damaged in any way	Visually inspect each cell looking for dents, or other damage	There should be no visual damage to any of the cells

Based on the tests above if any of the tests failed then the test would be repeated to check for potential human error. If that does not change the result, then an investigation into why the test failed would begin. This investigation would look at all aspects of the component that failed. Alternatives will be researched that can replace the current component.

7.5.2 Phase Two: Validation of Integration

This phase of testing is to confirm that all components work together in a system. This section will outline the criteria that the integrated system components were tested on.

Table 20: Pass/Fail criteria for phase two testing

Feature	Testing Procedure	Criteria
AC/DC Converter Validation		
MOSFETs operate at a switching frequency of 5000Hz	Connect MOSFET to the Arduino and switch at 5000Hz, measure switching frequency using an oscilloscope	The switching frequency measured should be exactly what was programmed
The DC voltage out of the AC/DC converter is as expected	Use a multimeter to measure the output voltage from the AC/DC converter	The measured voltage should be within 5% of the expected value
The DC current out of the AC/DC converter is as expected	Use a multimeter to measure the output current from the AC/DC converter	The measured current should be within 5% of the expected value
DC/DC Converter Validation		
MOSFETs operate at a switching frequency of 5000Hz	Connect MOSFET to the Arduino and switch at 5000Hz, measure switching frequency using an oscilloscope	The switching frequency measured should be exactly what was programmed
The DC voltage out of the DC/DC converter is as expected	Use a multimeter to measure the output voltage from the DC/DC converter	The measured voltage should be within 5% of the expected value
The DC current out of the DC/DC converter is as expected	Use a multimeter to measure the output current from the DC/DC converter	The measured current should be within 5% of the expected value
Cooling System Validation		
The Fan turns on when the ambient temperature surpasses 30 degrees Celsius	Use a heat source (blow dryer, heat gun) to increase the temperature above 30 degrees Celsius and monitor the fan	The fan should turn on as soon as the temperature rises above 30 degrees Celsius
The Fan turns off when the ambient temperature drops below 30 degrees Celsius	Use a fan to decrease temperature below 30 degrees Celsius and monitor the fan	The fan should turn off as soon as the temperature drops below 30 degrees Celsius

Based on the tests above if any of the tests failed then the test would first be repeated to check for potential human error. If that does not change the result, then an investigation into why the test failed would begin. This investigation would look at all aspects of the feature that failed. From iteration of the design will be considered to determine how to achieve the same desired result with a different setup or using alternate components (if available).

7.5.3 Phase Three: Validation of Entire System

This phase of testing is to confirm that the entire system works as intended. This section will outline the criteria that the system was tested on.

Table 21: Pass/Fail criteria for phase three testing

Feature	Testing Procedure	Criteria
Software Tests		
Code uses object-oriented programming to increase efficiency	Visually inspect the code to ensure it uses object-oriented programming	The code should use object-oriented programming where possible
Software writes progress updates to the console for debugging purposes	Run the system and check the output in the console	The console should display updates throughout its setup and runtime to show its progress
The software can run charge/discharge cycles without error/stopping	Run the system through charge/discharge cycles	The system should remain operational throughout the runtime
Charging Tests		
Battery charges to 95% in 50 minutes	Using a stopwatch or Arduino system track the amount of time for the battery to go from 10-90% of charge	The measured time should be within 5% of the expected time
LED Light displays 5 colours at the correct charge stages	Monitor the LED lights throughout the charging process and track when the lights change	The LED should display: Red: An error condition Yellow: Charging paused Purple: Charging in process Blue: Ready to charge Green: Charge complete
The battery can be efficiently discharged to 10%	Using the variable load discharge the battery and monitor components for heating	The batteries should reach 10% without any components overheating

Based on the tests above if any of the tests failed then the test would first be repeated to check for potential human error. If that does not change the result, then an investigation into why the test failed would begin. This investigation would look at all aspects of the feature that failed, at this stage since all the components were tested and all of the component integration has been tested, the error would likely be stemmed from either the software, the control system, or human error. Since these aspects are easily adjustable the software and control system would be reviewed followed by all the team members reviewing the setup to reduce the chance of human error.

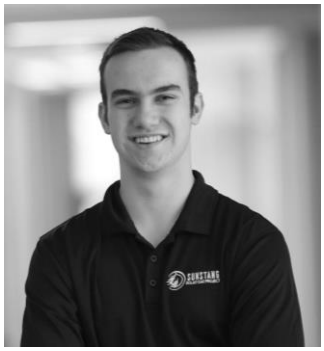
7.6 Appendix F - Battery Testing

Table 22: Battery internal resistance measurements

		Internal Resistance (mΩ)					
Battery	**Serial Number**	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Average**
1	FH_-1905613-AG01849	147	166	141	162	179	159
2	FH_-1905613-AG04989	109	100	100	103	100	102.4
3	FH_-1905613-AG05095	94	95	95	96	95	95
4	FH_-1905613-AG05108	95	100	95	96	95	96.2
5	FH_-1905613-AG05298	105	112	111	109	103	108

Batteries 3 and 4 were selected for two main reasons; first, they have the lowest average internal resistance of the 5 tested batteries, and second, they are the closest two average internal resistances out of the 5 tested batteries so they should experience similar effects.

7.7 Appendix G – Authors



Bruce Gillespie is currently completing a Bachelor of Engineering Science in Electrical Engineering at Western University. Throughout this project, Bruce focused on the simulation of the charger models. After graduation, he is going on to work as an ASIC Verification Engineer.

Adam Kidd is currently completing a Bachelor of Engineering Science in Electrical Engineering at Western University. Throughout this project, he focused on simulations of the charger models. After graduation, he will be pursuing a Master's degree in Analog and Digital Integrated Circuit Design at Imperial College London.

Bailey Thompson is currently completing her Bachelor of Engineering Science in Electrical Engineering at Western University. She was responsible for the hardware design of the prototype in this project. After graduation, she will be working as an engineering consultant in the controls and automation industry.

Parth Vachharajani is currently completing his Bachelor of Engineering Science in Electrical Engineering at Western University. For the duration of this project, he was a software designer for the physical implementation of the hardware prototype. Following graduation, he will be working in the electric power industry while completing an MEng degree part-time at the University of Toronto.