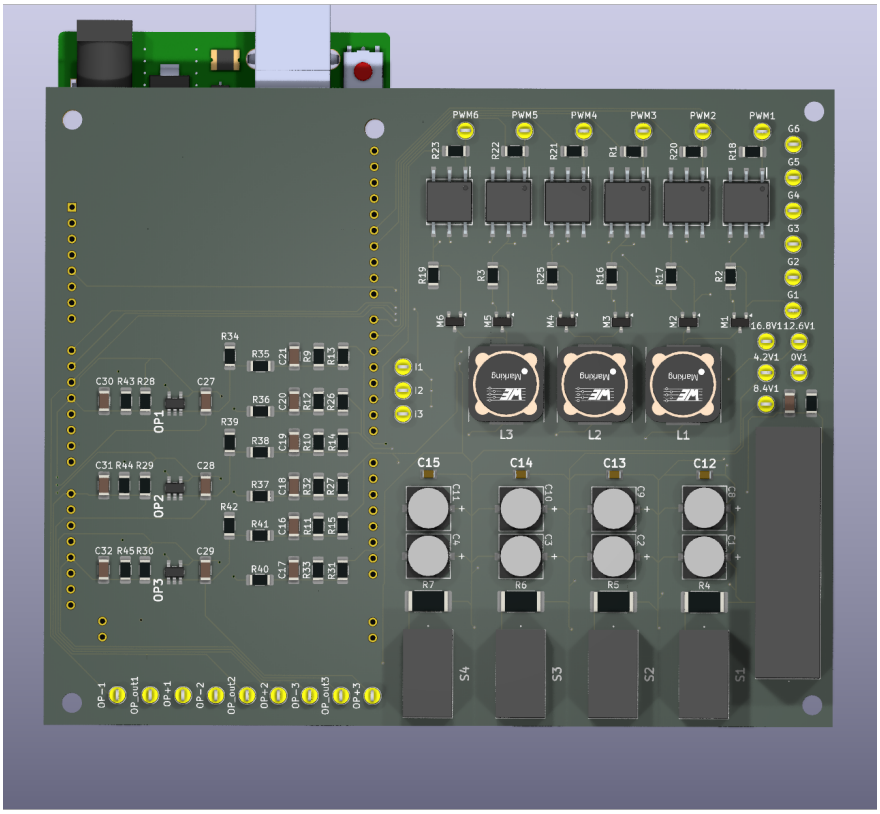
CHALMERS
UNIVERSITY OF TECHNOLOGY



Development of an active battery cell balancing circuit

Bachelor's thesis in Electrical Engineering

Elin Johansson, Henrik Larsson Hjorth, Samuel Rosén, Linus Sundberg, Johan Stridh & Hugo Åberg

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Bachelor's thesis 2025

Development of an active battery cell balancing circuit

Elin Johansson
Henrik Larsson Hjorth
Samuel Rosén
Linus Sundberg
Johan Stridh
Hugo Åberg

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2025

Development of an active battery cell balancing circuit
Elin Johansson, Henrik Larsson Hjorth, Samuel Rosén, Linus Sundberg, Johan Stridh & Hugo Åberg

Supervisor: Stefan Lundberg & Vaishnavi Ravi, Department of Electrical Engineering
Examiner: Jimmy Ehnberg, Department of Electrical Engineering

Bachelor's thesis 2025
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: 3D model of the PCB design.

Typeset in LaTeX
Gothenburg, Sweden 2025

Development of an active battery cell balancing circuit
Elin Johansson, Henrik Larsson Hjorth, Samuel Rosén, Linus Sundberg, Johan Stridh & Hugo Åberg
Department of Electrical Engineering
Chalmers University of Technology

Abstract

To further the research regarding battery balancing technology, this project evaluated various battery cell balancing topologies with the aim of identifying an alternative to passive battery cell balancing. Using a modified Pugh matrix, the multiple switched inductor topology was selected for further analysis, due to its high performance in the criteria: Balancing time, efficiency, and low complexity. A circuit implementing the selected topology was designed using MathWorks Simulink, LTspice and KiCad. Simulink was used to simulate on a system level, where ideal components were used to test the operation of the circuit and control system. The LTspice simulation was used for component selection. Building on the simulations, KiCad was then used to design a physical balancing circuit to be evaluated. The tests yielded that the active battery cell balancing circuit was capable of moving charge bidirectionally between two cells. Moreover, the balancing efficiency, at a reference current of 0.68 A, was determined to be 46.9% in simulations and 41.0% in testing.

Sammanfattning

För att främja forskningen kring batteribalansningsteknik utvärderade detta projekt olika topologier för battericellsbalansering, i syfte att identifiera ett alternativ till passivbattericellsbalansering. Med hjälp av en modifierad Pugh-matris valdes topologin med flera omkopplade induktorer för vidare analys, tack vare dess höga prestanda inom kriterierna: balanseringstid, verkningsgrad och låg komplexitet. En krets som implementerar den valda topologin designades med hjälp av MathWorks Simulink, LTspice och KiCad. Simulink användes för simulering på systemnivå, där ideala komponenter användes för att testa kretsens och reglersystemets funktion. LTspice-simuleringen användes för komponentval. Utifrån simuleringarna designades därefter en fysisk balanseringskrets i KiCad för utvärdering. Tester visade att den aktiva battericellsbalanseringskretsen kunde förflytta laddning i båda riktningarna mellan två celler. Dessutom fastställdes balanseringseffektiviteten, vid en referensström på 0,68 A, till 46,9 % i simuleringar och 41,0 % i praktiska tester.

Keywords: active, Arduino, balancing, battery, inductor, lithium-ion, PCB, switching.

Acknowledgements

We sincerely thank Vaishnavi Ravi and Stefan Lundberg for their crucial guidance throughout this research. Their competence and assessment were a significant help during the project.

Elin Johansson, Henrik Larsson Hjorth, Samuel Rosén,
Linus Sundberg, Johan Stridh and Hugo Åberg
Gothenburg, May 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

BESS	Battery Energy Storage System
DoD	Depth of Discharge
IMC	Internal Model Control
I/O	Input/Output
LED	Light Emitting Diode
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
NMOS	n-channel MOSFET
OCV	Open Circuit Voltage
OP amp	Operational amplifier
PCB	Printed Circuit Board
PI controller	Proportional Integral controller
PMOS	p-channel MOSFET
PWM	Pulse-Width Modulation
RMS	Root Mean Square
SoC	State of Charge
SoH	State of Health

Contents

1
Introduction

1.1 Background

Energy crisis, a condition where the demand for energy sources is higher than the available supply, is a pressing and widely discussed issue of today [1], [2]. The expansion of renewable energy is important for tackling the ongoing energy crisis [3]. Renewable energy sources typically rely on wind and solar power, making electricity generation vulnerable to weather fluctuations and therefore more volatile [4]. As renewable energy makes up a larger part of generated power, the volatility increases. To reduce the power fluctuations caused by weather changes, a technology called a Battery Energy Storage System (BESS) could be used. The BESS takes charge from a grid or a power plant and stores it in batteries in order to later discharge energy when needed [5].

Due to the limitations in voltage and capacity of a single battery cell, BESSs often need many connected cells [6]. Many connected battery cells form a battery pack, and it is impossible to avoid differences between each cell in the pack. These differences are often caused by variations in the manufacturing process or the operating environment. Another cause of these differences is the batteries' State of Health (SoH), which describes the degradation of the batteries' capacity relative to their original value over their lifetime [7]. The variations between the cells can cause a reduction in the cells' charge efficiency and lifespan [8]. These reductions can impact the state of charge (SoC), which is the ratio of total available charge in a battery cell in relation to the maximum available capacity of the cell, to differ between the cells in a battery pack when it is in use [9]. To counter differences between the battery cells, a balancing circuit can be used. Balancing circuits come in a wide variety, but can be split into two main groups, passive and active.

Passive balancing functions by dissipating energy from the cell with the highest SoC into a resistor until the cell has the desired SoC [10]. It is the most commonly used balancing method due to low cost, small size, and easy control algorithms [11]. An active battery cell balancing circuit, active balancing circuit for short, instead redistributes the charge between the cells with the higher SoC to cells with the lower SoC [12]. Although active balancing is more complex and difficult to implement, the advantage of it is that more of the existing charge in the pack can be utilized instead of being dissipated as heat with the passive solution.

Active balancing circuits can be designed in many ways, depending on the choice and

arrangement of the components in the circuit, often referred to as circuit topology [13]. Due to different components and ways of redistributing the charge, topologies will vary in balancing time, complexity, size, cost, and efficiency. If active balancing circuits are to become a more common solution than the passive ones, the various existing topologies must be evaluated in order to identify the most promising one.

1.2 Purpose

The purpose of the project was to investigate and evaluate various active balancing topologies, to determine which topology performs best according to relevant criteria. The project also aimed to simulate, design, construct, and evaluate a circuit implementing the selected topology.

1.3 Scope

The circuit was designed for the lithium-ion battery cell INR18650 MJ1 3500 mAh from LG Chem [14]. Lithium-ion batteries were chosen since these are commonly used [15]. Consequently, no comparison between the circuit's performance with other types of battery cells was performed. The functionality of the active balancing circuit will be the same, but the performance can differ when changing to another lithium-ion battery cell.

The circuit was designed for a maximum of four battery cells, but due to safety reasons, only two battery cells were connected during testing. Furthermore, testing with two battery cells still provided proof of concept.

All necessary control functions were developed using the Arduino integrated development environment. However, no single program containing a complete balancing algorithm was developed. The project's aim was rather to prove the circuit functionality and that balancing could be done.

The effect the circuit has on the SoH for a battery pack is not quantified. This is because the project's aim was to evaluate an active balancing circuit's capability of balancing and not its impact on battery degradation. Although no investigation into battery degradation was made, a general discussion regarding the impact on SoH between passive and active balancing was done.

2

Battery cell balancing

In this chapter, some environmental effects of battery production and balancing will be presented. Furthermore, a brief examination of the differences between balancing topologies is conducted, examining important factors such as: cost, complexity, size, efficiency, and balancing time [16]. The selected topology and its theory are also presented, since the functionality of the selected circuit is essential for understanding the rest of the report.

2.1 Environmental effects

One of the most commonly used battery types in BESS applications is the lithium-ion battery [17]. However, the extraction of lithium has been associated with significant environmental damage in the regions where it occurs [18]. The world's largest lithium reserves are found in an area commonly referred to as the Lithium Triangle, located in Bolivia, Chile, and Argentina. In this region, lithium mining has contributed to more frequent droughts and the depletion of underground water sources. These environmental changes have adversely affected local farmers, limiting their ability to sustain crops and livestock. Additionally, studies have shown that lithium mining operations in China has caused chemicals to leak into the Liqi River, killing the aquatic life and local livestock [19].

Due to the negative impact of lithium extraction, it would be of interest to prolong the lifespan of lithium-ion battery packs. One of the factors that affects battery aging is Depth of Discharge (DoD), a measurement of how much of a battery's stored energy is depleted in one charge-discharge cycle [20]. DoD is in the range 0 to 1 and is calculated as

$$DoD = 1 - SoC. \tag{2.1}$$

Active balancing makes the SoC of the lowest charged cell approach the pack average [12]. Passive balancing, on the other hand, makes the pack average approach the cell with the lowest SoC, lowering the pack's SoC [10]. Due to the inverse relationship between DoD and SoC, the DoD will become lower as the SoC gets higher.

A high DoD will accelerate the battery cell's degradation process due to chemical,

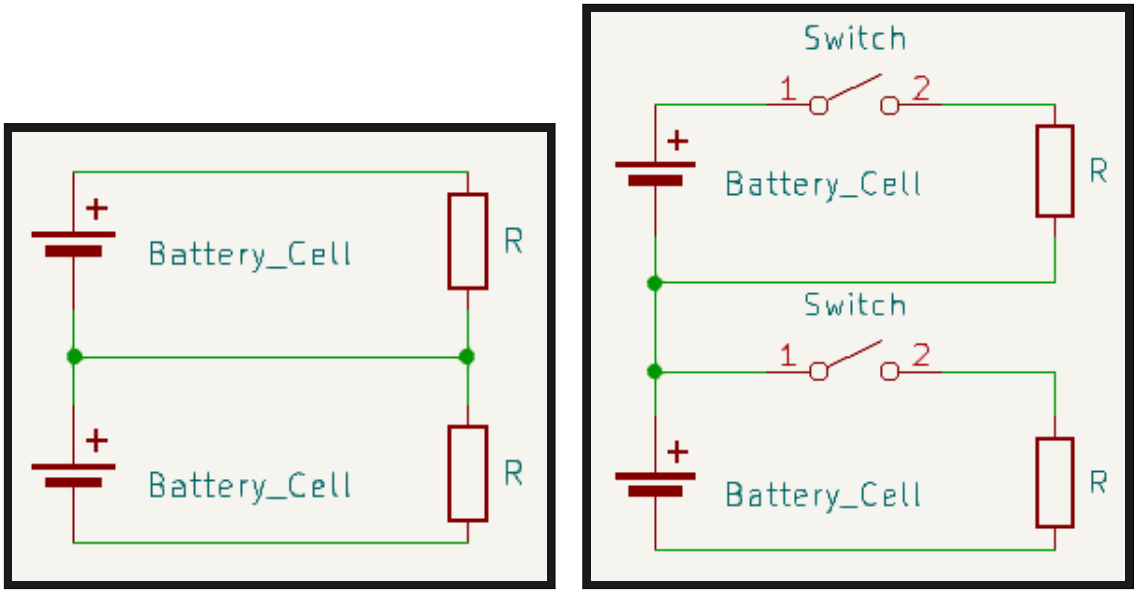
thermal and mechanical stress induced on the battery cells [20]. The accelerated degradation of the battery cells means that they will have less ability to hold and deliver energy, meaning they will have to be replaced earlier. By improving active balancing, more energy can be extracted from the pack, causing a lower DoD, which means less new lithium needs to be extracted.

On the other hand, active balancing circuits are generally more complex than passive ones [13], primarily due to the advanced circuitry and use of many semiconductor devices. The manufacturing of semiconductor components and chips leaves a large freshwater footprint [21]. Freshwater is not only used for electricity generation during manufacturing but also as a coolant. Therefore, minimizing the use of semiconductors in active balancing circuits becomes crucial for their sustainable and ethical development.

2.2 Resistor based balancing

The passive balancing method is implemented using resistors, which dissipate the extra energy as heat [22]. The simplest implementation of a resistor based circuit can be accomplished using fixed shunt resistors, which can be seen in Figure 2.1a. Equally sized resistors are connected in parallel over the cells so that cells with higher voltage drain faster than lower voltage cells. This is the least complex method, requiring no monitoring or control system. The drawback of this is that the cells will constantly be draining charge. This topology is therefore unsuited for lithium-ion batteries since they require a control system that ensures that the voltage levels remain within safe boundaries [16].

Another passive method can be implemented by using switched shunt resistors, which is shown in Figure 2.1b [22]. This topology is more complex than the previously mentioned fixed shunt resistor solution since it requires a control system to monitor the cells and control the switches. When the control system detects an imbalance between cells, it closes a switch until the cell voltages have equalized. The switched shunt resistor will therefore only discharge battery cells that require balancing, instead of continuously discharging all of the cells like the fixed resistor.



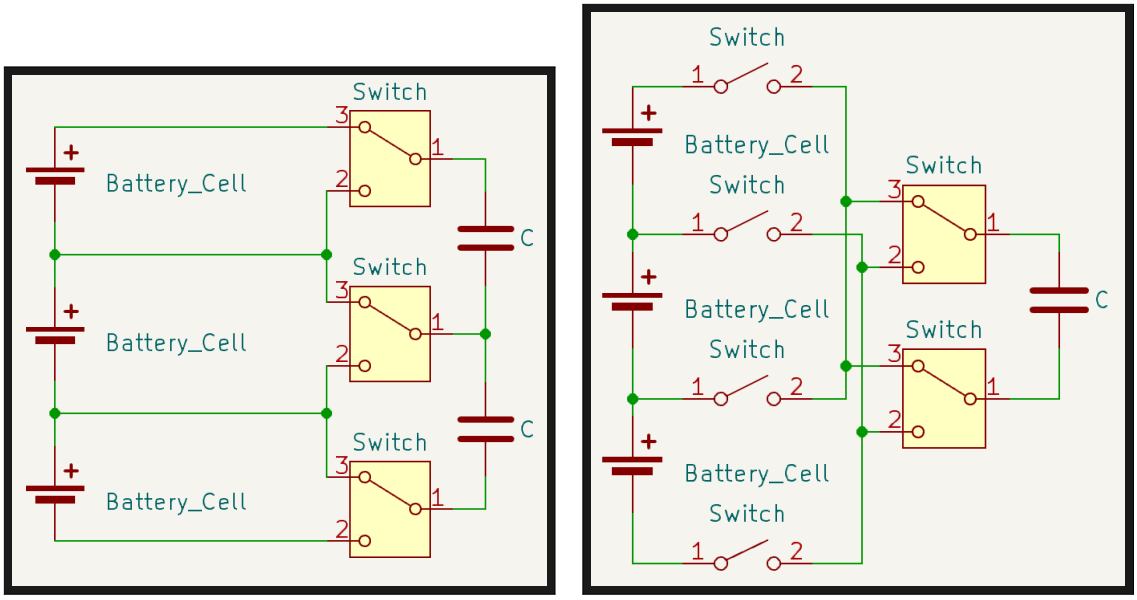
(a) Fixed shunt **(b)** Switched shunt

Figure 2.1: Resistor based balancing topologies.

The advantages of passive balancing are the simplicity and small size needed to implement it, in turn leading to a low cost [22]. The switched shunt design can also have a short balancing time depending on the choice of resistors. The disadvantages are that the extra heat generated can require the addition of thermal management and that the energy is lost, instead of being used more efficiently [23]. This leads to a trade off with a faster balancing time, meaning more heat generation and therefore necessitating better thermal management. In comparison to an active balancing topology, the battery will also be limited by the weakest cell in the pack instead of by the average cell, leading to quicker degradation of the battery [22].

2.3 Capacitor based balancing

An active balancing circuit can be constructed using capacitors to store and move charge between cells [23]. To balance the cells, switches are used to distribute the charge from one cell to the capacitor. The switches are closed and opened on alternating sides of the cells, allowing the charge to equalize. A common design is the multiple switched capacitor topology shown in Figure 2.2a [22]. It is constructed with capacitors connected in parallel between each cell. This lets the charge move between neighboring cells using the capacitors. The other design using capacitors, seen in Figure 2.2b, is a single capacitor connected in parallel with all cells, called a single switched capacitor [16]. This lets the controller move charge directly from the highest cell to the lowest, lowering balancing time.



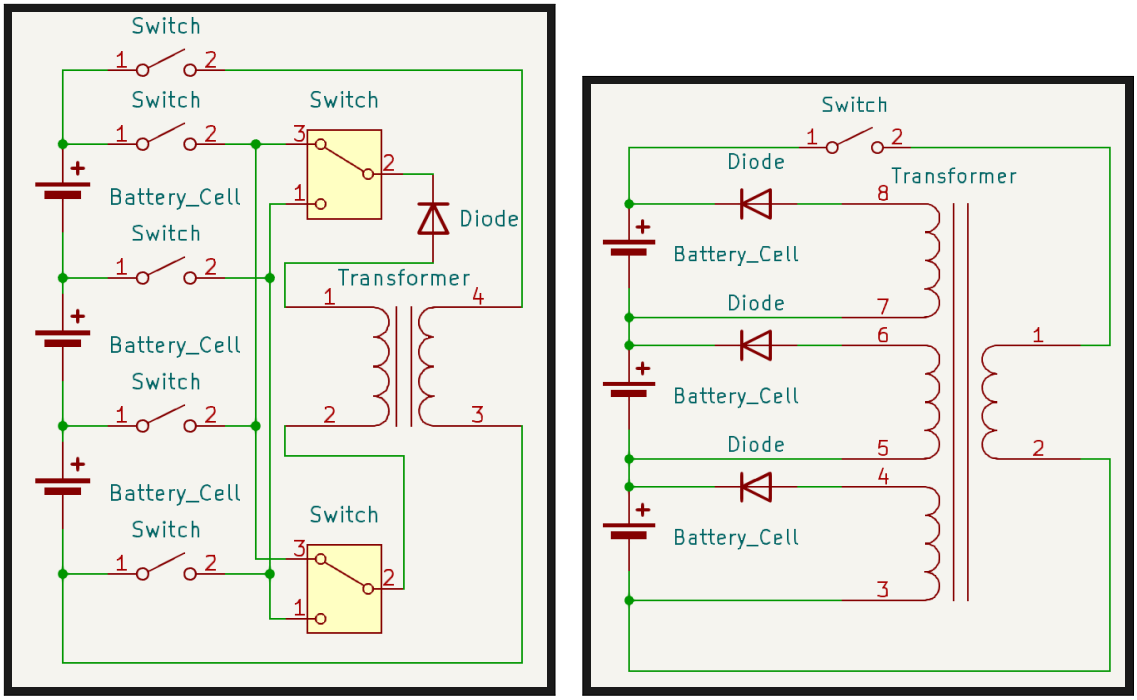
(a) Multiple switched **(b)** Single switched

Figure 2.2: Capacitor based balancing topologies.

The capacitor based balancing is small and easy to implement compared to many other active balancing methods. It excels in balancing cells with high voltage differences, while it is less effective at balancing cells with low voltage differences [22]. The multiple switch topology also becomes quite slow when more cells are added. This is due to the fact that the multiple switch topology is only capable of balancing adjacent cells. The single switched topology, in comparison, requires more switches and therefore requires a more complex control system [16].

2.4 Transformer based balancing

Transformers can be used to move charge quickly between cells. Several different transformer topologies can be employed. One example is the switched transformer, seen in Figure 2.3a, where the pack is connected to one side of the transformer, and the other is connected to a single cell [22]. The control system switches rapidly between the battery pack and cells, moving charge from the pack to those cells with the lowest SoC. A diode is used on the cell side of the transformer to ensure that charge is only added to the cell. The design in Figure 2.3b, uses a more complex transformer with multiple windings [22]. This topology is called a shared transformer and uses a separate winding and diode for each cell. It also uses rapid switching but can have a simpler control structure.



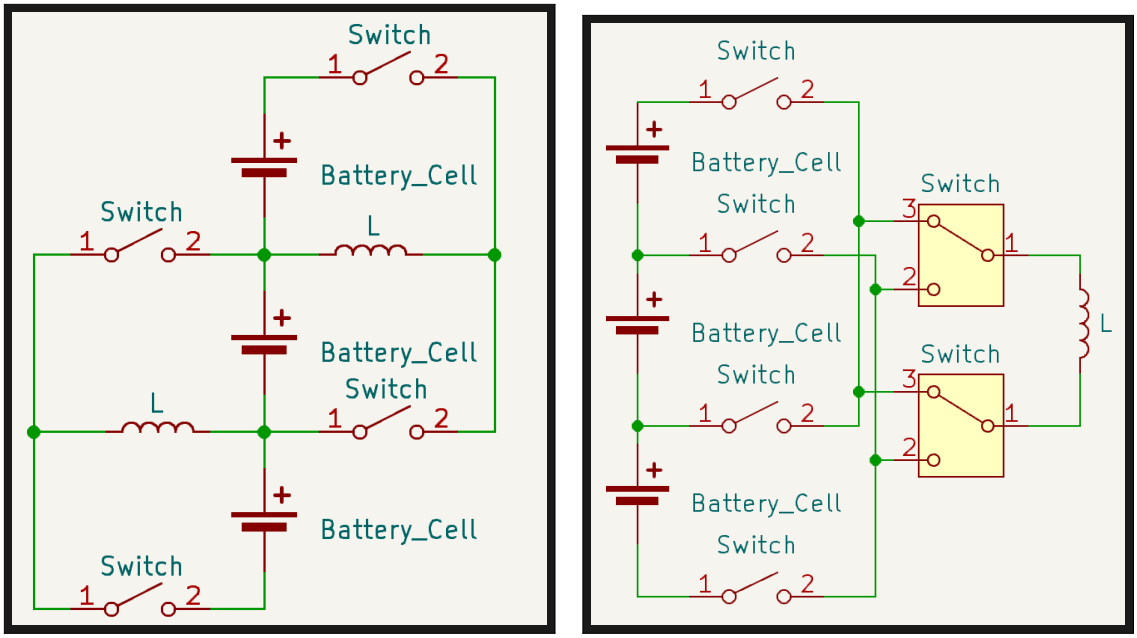
(a) Switched **(b)** Multiple winding

Figure 2.3: Transformer based balancing topologies.

The transformer based topologies have low balancing times and high efficiency [22]. Depending on the selected topology, the control system can be made very simple. The drawbacks are the high cost of the transformers and their size, as they take up more space than other balancing circuits. The rapid switching can also make it necessary to add filters to the circuit, to block the high frequency noise from interfering with the battery [16].

2.5 Inductor based balancing

The inductor topologies utilize inductors to balance cells. The design of these circuits is similar to the capacitor based ones but uses inductors to move charge between the cells instead of capacitors [23]. The control system switches the higher SoC cell on first and then the lower SoC cell to transfer energy, equalizing the cells [16]. A multiple switched design, which can be seen in Figure 2.4a, with inductors between each cell, is one possible implementation. Another implementation seen in Figure 2.4b is the single switched inductor design. This design uses one inductor but a more complex setup of switches to move charge directly from one cell to another.



(a) Multiple switched **(b)** Single switched

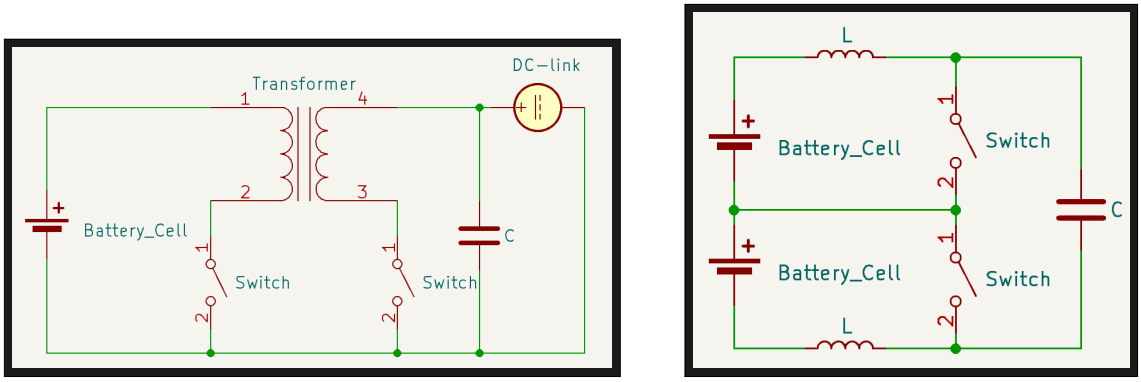
Figure 2.4: Inductor based balancing topologies.

The strength of these topologies is the high efficiency and fast balancing times [22]. Though for the multiple switched topology, the further apart the unbalanced cells are, the higher the required balancing time is [16]. This is due to the fact that the multiple switched topology is only capable of balancing adjacent cells. The single switched topology instead requires more complex control circuitry. Another drawback of the inductor based balancing technology is the high current stress on the switches [16]. High switching frequency due to the fast operation can also lead to a need to implement a filter in the circuit, further increasing the complexity [24].

2.6 Power converter based balancing

Converter topologies come in a wide variety and can use both capacitive and/or inductive components in their design [16]. One type of design, seen in Figure 2.5a, is a Flyback converter. It can be unidirectional, where charge is stored in the transformer and then transferred to the battery pack. The other configuration is bidirectional, allowing charge to move from both cell to pack and pack to cell.

Another converter based balancing topology is the bidirectional Cûk converter, the schematic of which can be seen in Figure 2.5b [16]. The balancing is similar to some of the previously mentioned topologies in that it balances the neighboring cells with one another. However, unlike the capacitive or inductive based alternatives mentioned earlier, it utilizes both types of components.



(a) Flyback **(b)** Cûk

Figure 2.5: Converter based balancing topologies.

The bidirectional Flyback converter has increased flexibility since energy from the battery pack can also be transferred to the cell [16]. It is also simple to implement for large battery packs [25]. The drawbacks of this converter are the uniform winding, the incurred magnetic losses in the transformer and it requires accurate voltage sensing [16].

The advantage of the Cûk converter is that it will have a continuous current at both input and output and a high efficiency [26]. Its drawbacks are the long balancing time for large battery packs, the high switch current, and the need for accurate voltage sensing [16], [27].

2.7 Summary of the topologies' advantages and disadvantages

The optimal choice of balancing topology depends on the manufacturers' requirements and limitations. Factors that weigh heavily are often cost, complexity, and size. This has favored the passive balancing topologies historically, and they are therefore the most common type today [28]. The large amount of different active balancing topologies also means that research must be made to determine which active balancing topology suits this project best. To provide an overview of the findings in previous sections, the advantages and disadvantages of the different topologies have been collected into Table 2.1.

Table 2.1: Overview of the different topologies' advantages and disadvantages [16], [22]-[27]

Topology	Advantages	Disadvantages
Fixed resistor	Cheap Simple Small form factor	Low balancing efficiency Thermal management
Switched shunt resistor	Cheap Simple Short balancing time	Low balancing efficiency Thermal management
Multiple switched capacitor	Simple Small form factor	Long balancing time Slow balancing for large battery packs
Single switched capacitor	Simple Small form factor	Long balancing time High amount of switches Complex control
Switched transformer	Short balancing time High balancing efficiency	Expensive Large form factor Complex control High frequency filtering
Multi-winding transformer	Short balancing time High balancing efficiency Simple control	Expensive Large form factor High frequency filtering
Multiple switched inductor	Short balancing time High balancing efficiency	Slow balancing for large battery packs High current stress on switches High frequency filtering
Single switched inductor	Short balancing time High balancing efficiency	Complex control High current stress on switches High frequency filtering
Flyback converter	Simple implementation for large battery packs Flexibility	Voltage sensing necessary Moderate balancing time Magnetic losses
Cûk converter	High balancing efficiency	Voltage sensing necessary Moderate balancing time

2.8 In-depth analysis of inductive balancing

In this section, a comprehensive analysis of the inductor-based balancing circuit shown in Figure 2.6 is provided. The circuit in Figure 2.6 has the same working principle as the circuit presented in Figure 2.4a, and is a simplified version of the active balancing circuit that was selected.

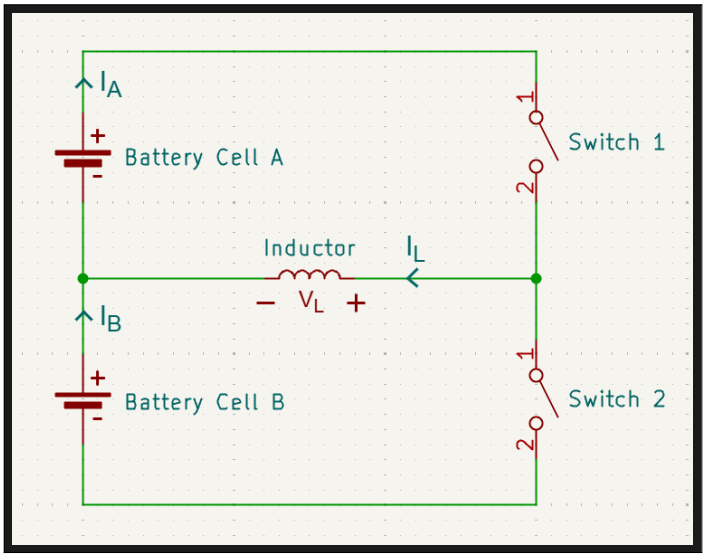


Figure 2.6: The simplified inductor-based balancing circuit that was analysed to describe the balancing process [16].

The circuit in Figure 2.6 consists of two battery cells, battery cell A and battery cell B, one inductor, and two ideal switches, switch 1 and switch 2. Redistributing charge from one cell to another uses the inductor's inherent ability to store magnetic energy [29]. For inductors to store magnetic energy, an electrical current must flow through it and when current flows through an inductor, the inductor voltages is

$$V_L(t) = L \cdot \frac{dI_L(t)}{dt}, \tag{2.2}$$

where L is the inductance, and $\frac{dI_L(t)}{dt}$ is the rate of current change. Equation (2.2) indicates that when a voltage is applied to an inductor, a current change will emerge. In the circuit shown in Figure 2.6, a voltage can be applied to the inductor by switching on either switch 1 or 2. Switching on switch 1 means it will start to conduct, and V_L will equal the voltage, V_A, across battery cell A. Similarly, switching on switch 2 will cause V_L to be equal to the negative of the voltage across battery cell B, i.e., $V_L = -V_B$. By toggling switches 1 and 2, such that only one switch is on at a time, V_L will resemble the waveform shown in Figure 2.7 [30].

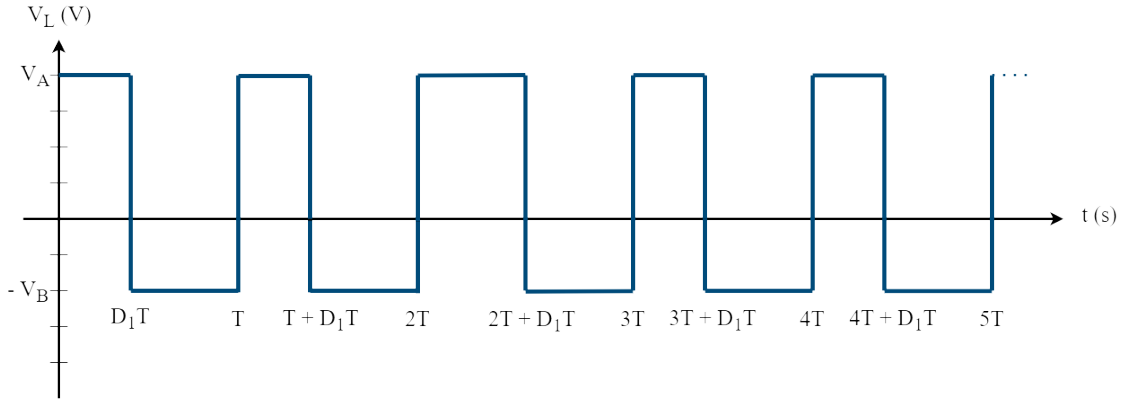


Figure 2.7: The ideal voltage across the inductor sketched over five switching periods [30].

The toggling seen in Figure 2.7 can be achieved with Pulse Width Modulation (PWM). PWM is a technique that allows analog signals, such as voltage and current, to be controlled digitally [31], [32]. The digital control is achieved by alternating a digital signal between a high and a low state. The portion of the period when the PWM signal is in its high state is called its duty cycle, and ranges between 0 and 1 [33]. In Figure 2.7, toggling is accomplished using two PWM signals that are out of phase, meaning they are never high simultaneously. This means that when a switch is on, its corresponding PWM signal is in the high state, and when a switch is off, the PWM signal is in the low state. Furthermore, the toggling in Figure 2.7 is repeated with a fixed period time, meaning the sum of the on time of switch 1 and the on time of switch 2 is constant and equal to the switching period time, T. The duty cycle, D_1, in Figure 2.7 is the duty cycle for switch 1, which means its on time, t_{s1}, can be calculated as $t_{s1} = D_1 T$. Similarly, the on time for switch 2, t_{s2}, can be calculated as $t_{s2} = D_2 T$, where $D_2 = 1 - D_1$.

Moreover, according to [29], the current through the inductor can be derived by solving (2.2) for the inductor current, $I_L(t)$, giving

$$I_L(t) = I_L(0) + \frac{1}{L}\int_0^{D_1 T} V_A(t)dt \quad \text{for } 0 \leq t < D_1 T, \tag{2.3}$$

where $I_L(0)$ is the initial inductor current and $D_1 T = t_{s1}$, and is the duration that switch 1 is on. When switch 1 switches off at $t = t_{s1}$, switch 2 switches on, meaning that the inductor current can be written as

$$I_L(t) = I_L(D_1 T) - \frac{1}{L}\int_{D_1 T}^{T} V_B(t)dt \quad \text{for } D_1 T \leq t < T, \tag{2.4}$$

where $I_L(D_1 T)$ is the current at the time $t = D_1 T$ and T is the time when switch 2 switches off again. Assuming $I_L(0) \neq 0$ means that $I_L(t)$ will resemble the current shown in Figure 2.8 if the voltage in Figure 2.7 is applied over the inductor [30].

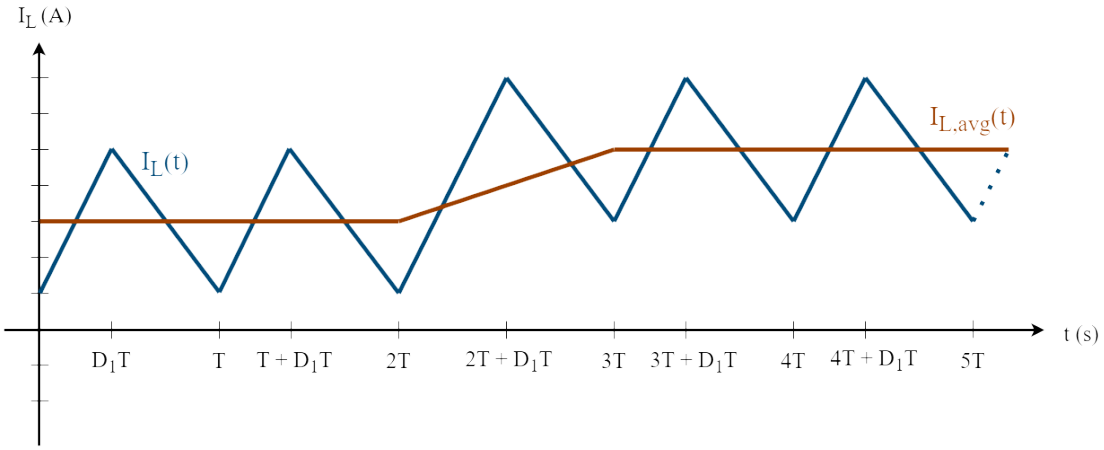


Figure 2.8: The ideal inductor current, derived from (2.3) and (2.4) and Figure 2.7 [30].

As can be noticed in Figure 2.8, the inductor current, I_L, increases and decreases according to (2.3) and (2.4) depending on the duty cycle, D_1, and the switching period time, T. The relation between inductor current and duty cycle can be seen between $2T$ and $3T$ in Figure 2.8, where D_1 is increased, causing I_L to increase.

Furthermore, the inductor current at the end of any arbitrary period n, can be calculated by combining (2.3) and (2.4) as

$$I_L(n) = I_L((n-1)T) + \frac{1}{L}\int_{(n-1)T}^{nT} V_L dt = I_L((n-1)T) + \frac{T}{L}\frac{1}{T}\int_{(n-1)T}^{nT} V_L dt, \quad (2.5)$$

where, $I_L((n-1)T)$ is the current at the end of switching period $(n-1)$, and

$$\frac{1}{T}\int_{(n-1)T}^{nT} V_L dt = V_{L,avg}, \quad (2.6)$$

where, $V_{L,avg}$ is the period average of the inductor voltage, meaning

$$I_L(n) = I_L((n-1)T) + \frac{T}{L}V_{L,avg}. \quad (2.7)$$

Solving (2.7) for $V_{L,avg}$ gives

$$V_{L,avg} = L \cdot \frac{I_L(nT) - I_L((n-1)T)}{T} \approx L \cdot \frac{dI_{L,avg}}{dt}. \quad (2.8)$$

Equation (2.8) shows that the change of the inductor current depends on the average inductor voltage. If the current should remain constant over the period, the average

inductor voltage must be zero. Similarly, if the current increases or decreases, the average inductor voltage must be positive or negative, respectively. From Figure 2.8 it can be noted that the average inductor current lies in the middle of the current ripple, since

$$I_{L,avg} = \frac{1}{T}\int_{(n-1)T}^{nT} I_L(t)dt = I_L((n-1)T) + \frac{\Delta I_L}{2}. \tag{2.9}$$

From (2.8) and Figure 2.8, it can be seen that the change in the average inductor current is dependent on the average inductor voltage. Furthermore, the average inductor voltage can be written as

$$V_{L,avg} = \frac{1}{T}\int_{(n-1)T}^{nT} V_L(t)dt = \frac{1}{T}\int_{(n-1)T}^{(n-1)T+D_1T} V_A dt - \frac{1}{T}\int_{(n-1)T+D_1T}^{nT} V_B dt. \tag{2.10}$$

Simplifying (2.10) gives

$$V_{L,avg} = V_A D_1 - V_B(1 - D_1), \tag{2.11}$$

from which it can be noted that the average inductive voltage can be zero, positive, or negative, with a maximum value of $V_{L,avg} = V_A$ at $D_1 = 1$ and a minimum of $V_{L,avg} = -V_B$ at $D_1 = 0$. The duty cycle of switch 1 can thereby be calculated as

$$D_1 = \frac{V_{L,avg} + V_B}{V_A + V_B}, \tag{2.12}$$

and to get a zero average inductor voltage, the duty cycle should be selected as

$$D_1 = \frac{V_B}{V_A + V_B}. \tag{2.13}$$

As noticed before, the relationship in (2.13) gives an inductor current that does not change over the switching period, i.e., $I_L(nT) = I_L((n-1)T)$.

The charge that is moved from one battery cell to another depends on the current from the battery cells. From Figure 2.6, it can be seen that when switch 1 is on, the current of battery cell A, I_A, is equal to the inductor current, I_L; otherwise, it is zero. Similarly, for battery cell B, when switch 2 is on, the current of battery cell B, I_B, is equal to the negative inductor current, otherwise, it is zero. Based on this analysis, the battery currents can be drawn as shown in Figure 2.9.

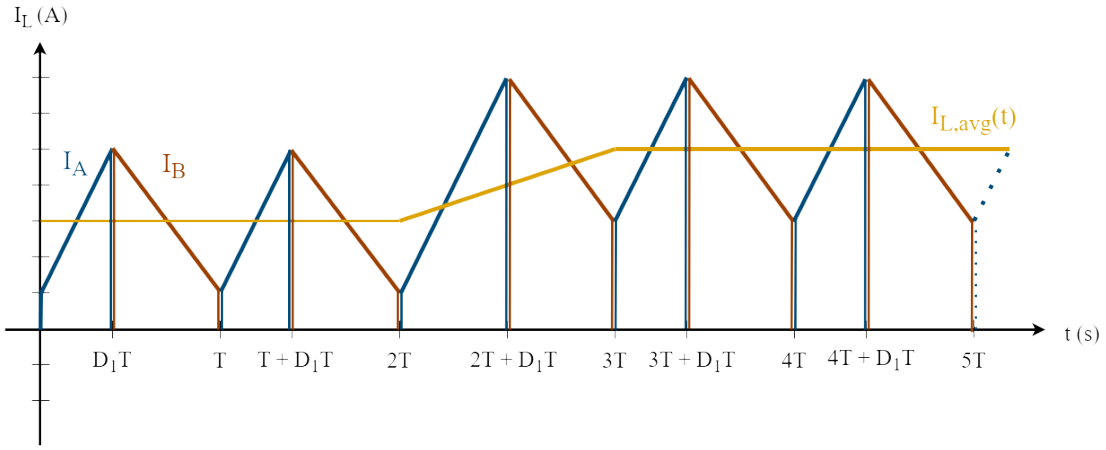


Figure 2.9: The current through battery cell A and battery cell B, I_A and I_B, along with the average inductor current.

The charge that is removed from battery cell A during switching period n can be described as

$$Q_A(nT) = \int_{(n-1)T}^{nT} I_A dt = \int_{(n-1)T}^{(n-1)T+D_1T} I_L dt, \tag{2.14}$$

where $(n-1)T + D_1T - (n-1)T = D_1T$, which means the integral containing I_L can be re written as

$$D_1T \frac{1}{D_1T} \int_0^{D_1T} I_L dt = I_{L,avg} D_1T. \tag{2.15}$$

Combining (2.14) and (2.15) means that the charge moved from battery cell A will be

$$Q_A(nT) = I_{L,avg} D_1T. \tag{2.16}$$

Similarly, the charge moved from battery cell B during switching period n is

$$Q_B(nT) = \int_{(n-1)T}^{nT} I_B dt = -\int_{(n-1)T+D_1T}^{nT} I_L dt = -I_{L,avg}(1-D_1)T. \tag{2.17}$$

From (2.16) and (2.17) it can be seen that with a positive average inductor current, charge is moved from battery cell A to battery cell B, and with a negative average inductor current, charge is moved from battery cell B to battery cell A. Since the cells are balanced by decreasing the SoC of one cell and increasing the SoC of the other, the balancing speed is directly controlled by $I_{L,avg}$, and this can be concluded by $|Q_A(nT)| + |Q_B(nT)| = I_{L,avg}T$. Moreover, the direction of charge transfer is

controlled by changing between positive and negative $I_{L,avg}$. Figure 2.10 illustrates how the charge would be redistributed between battery cell A and battery cell B given a positive average inductor current, when $SoC_A > SoC_B$.

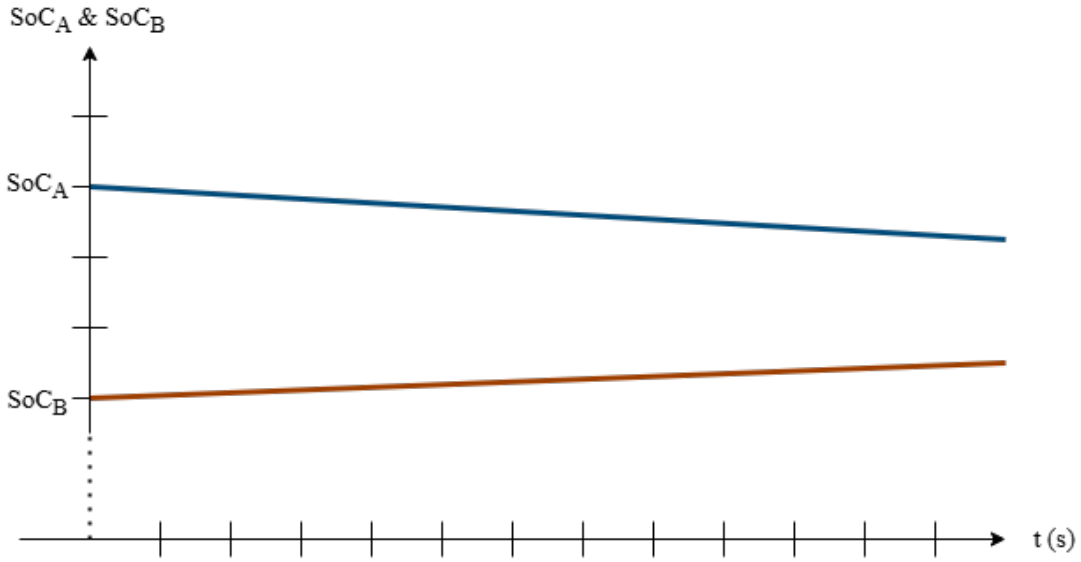


Figure 2.10: The SoC of battery cells A and B during the balancing process.

As can be seen in Figure 2.10, charge is moved from battery cell A and is redistributed to battery cell B. Thus SoC_A decreases and SoC_B decreases. A similar graph can be drawn for a negative average inductor current, where charge would be redistributed from battery cell B to battery cell A.

2.9 Current controller for inductive circuits

For facilitating control of the inductor current I_L, a proportional-integral controller (PI controller) can be used. A general PI controller has the form

$$F(s) = K_p + \frac{K_i}{s} \tag{2.18}$$

in which K_p is the proportional gain and K_i is the integral gain [34]. The term s^{-1} in the Laplace domain represents integration in the time domain, hence the second term integrates and accumulates the error [35]. The controller can be designed using internal model control (IMC), meaning that a mathematical estimation of the process to be controlled is used for the controller design [36]. Laplace transforming (2.8) yields

$$\frac{I_{L,avg}(s)}{V_{L,avg}(s)} = G(s) = \frac{1}{Ls} \tag{2.19}$$

in which L is the inductance. A PI controller controlling a purely inductive circuit can cause oscillatory behavior since there is no resistive damping [34]. This can be mitigated by adding active damping (a feedback loop having the same effect as resistance, without adding energy losses) R_a [34]. This model is illustrated in Figure 2.11.

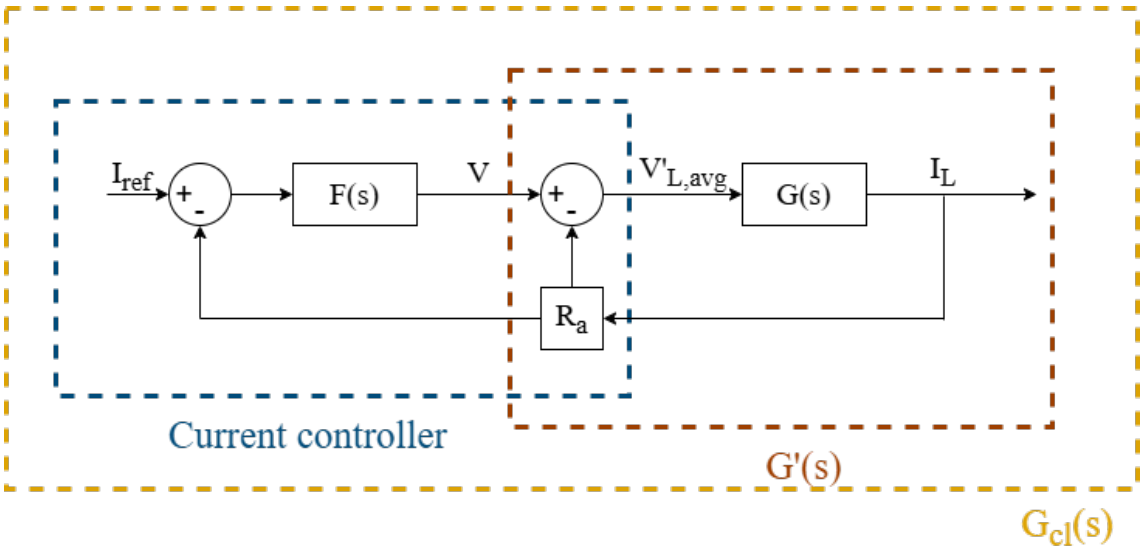


Figure 2.11: Current control loop and modified process model.

Figure 2.11 highlights the current controller (blue box) and the modified process model $G'(s)$ (red box) including active damping R_a. This process model gives

$$V'_{L,avg} = V - R_a I_L = LsI_L \tag{2.20}$$

and solving (2.20) for I_L leads to the modified process model

$$G'(s) = \frac{1}{Ls + R_a} = \frac{\frac{1}{L}}{s + \frac{R_a}{L}} = \frac{g\alpha_c}{s + \alpha_c} \tag{2.21}$$

with gain $g = R_a^{-1}$ and $\alpha_c = R_a L^{-1}$. Using the closed loop system $G_{cl}(s)$ (yellow box in Figure 2.11) selected as a first-order low pass filter with bandwidth α_c as

$$G_{cl}(s) = \frac{\alpha_c}{s + \alpha_c} = \frac{F(s)G'(s)}{1 + F(s)G'(s)} = \frac{\frac{\alpha_c}{s}}{1 + \frac{\alpha_c}{s}} \tag{2.22}$$

yields

$$F(s) = \frac{\alpha_c}{s} G'^{-1}(s) = \frac{\alpha_c}{s}(\hat{L}s + R_a) = \alpha_c \hat{L} + \frac{\alpha_c R_a}{s} = K_p + \frac{K_i}{s}. \tag{2.23}$$

in which L has been replaced by $\hat{L}$ since the estimation $\hat{L}$ is used instead of L when designing the controller using IMC . The controller parameters are $K_p = \alpha_c \hat{L}$ and $K_i = \alpha_c R_a$ [34]. The current control in the cell balancing circuit in Figure 2.6 amounts to adjusting the duty cycle of PWM signals, as illustrated in Figures 2.7 to 2.9, and the duty cycle is determined according to (2.12). Accordingly, the purpose of the current controller is to determine the voltage $V_{L,avg}$.

A further modification to the current controller can be made in order to limit a phenomenon called integrator windup. This occurs when the output voltage desired by the controller has a value $V_{L,avg}$ exceeding the maximum possible value $V_{L,max}$. This causes an overshoot above the reference current since the integration term accumulates, due to the current not following the reference as expected by the controller [34]. A PI current controller implemented with a back calculation loop limiting integrator windup is depicted in Figure 2.12.

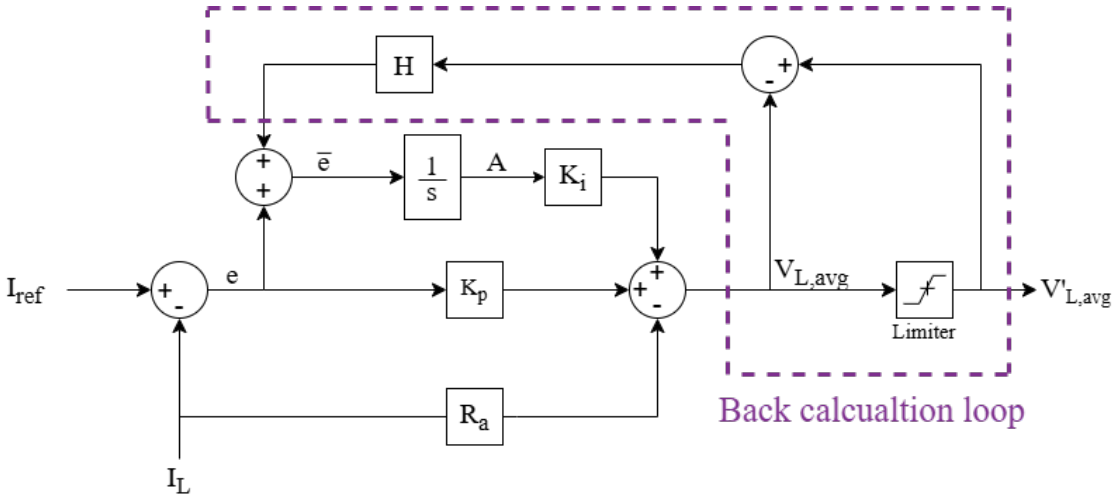


Figure 2.12: Block diagram describing the control system with the back calculation loop.

The back calculation loop in Figure 2.12, highlighted in the purple box, modifies the error

$$e = I_{ref} - I_L \tag{2.24}$$

sent to the integrator to a limited error term

$$\bar{e} = e + H(V'_{L,avg} - V_{L,avg}) \tag{2.25}$$

in which $V'_{L,avg}$ is the actual voltage output from the current controller and H is the gain for the signal $V'_{L,avg} - V_{L,avg}$. $V'_{L,avg}$ takes a value according to

$$V'_{L,avg} = \begin{cases} -V_{L,max}, V_{L,avg} < V_{L,max} \\ V_{L,avg}, |V_{L,avg}| < V_{L,max} \\ V_{L,max}, V_{L,avg} > V_{L,max}. \end{cases} \tag{2.26}$$

H is determined by studying the transfer function from $V'_{L,avg}$ to $V_{L,avg}$ and setting I_L and e to zero. This results in

$$V_{L,avg} = \frac{HK_i}{s + HK_i} V'_{L,avg} \tag{2.27}$$

and its bandwidth was chosen as $\alpha_c = HK_i$. Accordingly,

$$H = \frac{\alpha_c}{Ki} = \frac{1}{R_a} \approx \frac{1}{K_p}. \tag{2.28}$$

The current controller implementation then amounts to implementing the transfer function from I_{ref} to $V_{L,avg}$. From Figure 2.12, it can be noted that

$$V_{L,avg} = K_p e + K_i A - R_a I_L. \tag{2.29}$$

Furthermore, observing that the integrated area $A = \overline{e}s^{-1}$ and using (2.25), (2.29) can be rewritten as

$$V_{L,avg} = K_p e + \frac{K_i \overline{e}}{s} - R_a I_L \tag{2.30}$$

and subsequently multiplied by s to arrive at

$$\frac{dV_{L,avg}}{dt} = K_p \frac{de}{dt} + K_i \overline{e} - R_a \frac{dI_L}{dt} \tag{2.31}$$

since multiplication with s in Laplace domain corresponds to time derivatives in time domain [35]. Implementing the transfer function digitally, moreover, requires discretization, which can be done using the forward Euler method for approximating derivatives of a quantity $x(t)$ using its samples as

$$\frac{dx(t)}{dt} \approx \frac{x_k - x_{k-1}}{T_s} \tag{2.32}$$

in which T_s is the time between consecutive samples, x_k is the current sample and x_{k-1} is the previous sample [34]. Accordingly, (2.31) becomes

$$\frac{V_{L,avg,k} - V_{L,avg,k-1}}{T_s} = K_p \frac{e_k - e_{k-1}}{T_s} + K_i \overline{e} + R_a \frac{I_{L,k} - I_{L,k-1}}{T_S}. \tag{2.33}$$

Solving (2.33) for $V_{L,avg,k}$ yields

$$V_{L,avg,k} = K_p(e_k - e_{k-1}) + K_i \overline{e}_k T_s - R_a(I_{L,k} - I_{L,k-1}) + V_{L,avg,k-1}. \tag{2.34}$$

as the final result for the transfer function. The term $K_i \overline{e}_k T_s$ is expanded as

$$K_i \overline{e}_k T_s = K_i e_k T_s + H T_s (V'_{L,avg,k} - V_{L,avg,k}). \tag{2.35}$$

using (2.25). This is the term in the transfer function mitigating integrator windup. In (2.34), the term $V_{L,avg,k-1}$ represents the accumulated error as calculated by the integrator, while the proportional gain K_p operates of the difference between the current and last sampled error term and the active damping R_a operates on the difference between the current and last sampled inductor current.

3

Methods

This chapter describes the procedure for developing the active balancing circuit. This includes the method used for the topology selection, the design and simulation procedure of the selected topology, and the physical implementation and evaluation of a constructed prototype.

3.1 Topology selection

One of the aims of this project was to select an optimal topology that would be further developed. To achieve this aim, multiple topologies needed to be compared with respect to their characteristics and performance in different areas. The method of selection used in this project was a modified version of the Pugh matrix.

A Pugh matrix is a method used to compare the properties of different options. In this project, a modified version of the Pugh matrix was used to evaluate various battery cell balancing topologies [37]. One topology is selected as the baseline, where each property is given a score of zero. Then, the properties of each alternative topology are compared to the baseline. If an alternative performs better in a certain property, it gets a score of (+1). However, if it performs worse, it gets a score of (-1). Since some properties are more important, they are given different weights to reflect their importance. The score for each property is then multiplied by its weight. Finally, the total scores for each topology are calculated by summing up the scores related to the topology. The topology with the highest score will be the one implemented.

To make the Pugh matrix more weighted, parts of a modified Pugh matrix method were added [38]. The different topologies could perform slightly better or worse than the reference, or much better or worse than it. This would not be registered when the only available scores are (+1), (0), and (-1). Similarly to the method used in [38], the evaluation scores were extended from ± 1 to ± 3 to achieve a greater level of nuance when rating the different topologies.

3.2 Simulation and design

Designing an active balancing circuit based on the selected topology required conducting computer simulations. These simulations involved two simulation software programs: MathWorks Simulink and LTspice.

Simulink was used for simulating the balancing circuit on a system level, where ideal electrical components could be combined with the control system. In Simulink, values for the electrical components central to the balancing process were determined. This included the value of the inductor responsible for charge redistribution and the filtering capacitors that served to lessen the ripple voltage experienced by the battery cells.

The determining factors for the inductor choice were the root-mean-square (RMS) current of the inductor current and the size of the current ripple around the average inductor current (the ripple is shown in Figure 2.8). The RMS current limited the inductor choice, since it made it necessary to choose an inductor with a high enough saturation current; otherwise, the inductance would decrease at a certain RMS current. The size of the ripple around the average inductor current depended on the inductance of the coil and was found by rearranging (2.2) and assuming constant inductor voltage V_L, which yielded

$$\Delta I = \frac{V_L \cdot D_i T}{L}, \tag{3.1}$$

where $D_i T$ is the time during which switch i is on, where i is either 1 or 2. In this circuit, metal-oxide-semiconductor field-effect transistors (MOSFET) were used as the switches seen in Figure 2.6. Using (3.1) and assuming a highest V_L that the inductor would experience meant that the highest ripple current could be determined.

The component values for the inductor and capacitors were determined with the specifications of the INR18650 MJ1 3500 mAh battery cell, presented in [14], in mind. These specifications were not only considered for efficient balancing but also for safe balancing, as lithium-ion batteries are highly flammable when mishandled [39]. To address this, the Simulink simulations were also designed to minimize risks such as short-circuiting and overcharging, or completely discharging the cells. Furthermore, the Simulink simulations served to design and test a control system that could maintain a stable balancing current throughout the balancing process. The setup used in the Simulink simulations can be seen in Appendix D, Figure D.1, and contained the simplified balancing circuit shown in Figure 2.6 and the control system shown in Figure 2.12.

Complementing the Simulink simulations were the LTspice simulations, which gave a more comprehensive understanding of the requirements for the hardware implementation. This involved identifying the additional components necessary to enable the full operation of the circuit, beyond those required solely for the balancing functionality. This meant implementing the circuitry needed for controlling the MOSFETs

based on PWM signals. Additionally, the LTspice simulations were conducted using the relevant component models, allowing for component selection and validation. Two types of MOSFETs were used in the circuit, p-channel MOSFET (PMOS) and n-channel MOSFET (NMOS), and their component models were retrieved from the component manufacturer [40], [41]. The circuit also included inductors and opto-couplers, and their models were included in LTspice. The LTspice simulation setup is illustrated in Appendix D, Figure D.2, and includes all the circuitry needed to run the circuit in Figure 2.6.

The circuit design was done in KiCad and used the results from the Simulink and LTspice simulations. The KiCad schematic encompassed the complete circuit layout, including all components determined necessary for a fully functional active balancing circuit and its connections. Furthermore, the KiCad schematic also included space for filtering components that could be added in case the physical measurements were noisy. A design to measure the temperature and voltage over the battery cells and current in the inductor was also included. The final addition to the designed circuit was a method of individually lowering the SoC levels of the cells for testing purposes.

3.3 Hardware implementation

The construction of the active balancing circuit prototype involved creating a printed circuit board (PCB) design that could house all the components. The PCB design was made in the KiCad PCB editor, which meant the KiCad schematic could be transferred directly to the PCB editor. Transferring the schematic to the PCB editor required the correct footprints, which defines the physical interface between the PCB and the electrical components. All the footprints' connections were automatically designated according to the schematic, however, the integrated electrical PCB connections, called traces, needed to be joined manually. When designing the PCB, special consideration was put into reducing the length and use of traces to decrease unwanted inductance. Several test points were added to the PCB, which enabled taking measurements with an oscilloscope and debugging the circuit. Assembling the PCB consisted of soldering the components to the PCB.

The control system from the Simulink simulations was realized by programming a microcontroller, in this project, an Arduino Mega 2560 Rev3. The Arduino featured enough input/output (I/O) ports and the logic necessary to operate the circuit. The PCB was designed so that the Arduino could be mounted directly onto the board, and doing so avoided any additional inductance that would have been caused by external wires.

3.4 Evaluation

The evaluation of the hardware implementation had two main parts: One involved testing the Arduino, and the other evaluated the balancing circuit itself. The Arduino testing consisted of testing the functionality of the sensors and the control system. The accuracy of the sensors was evaluated by comparing the sensors' values

to measurements conducted with an external tool and calculating the measurement uncertainty. The voltage sensors were compared to values taken using a multimeter. The current sensors measuring the balancing current through the inductor were compared to measurements taken with a current probe connected to an oscilloscope. These tests were conducted using the setup shown in Figure 3.1, which was comprised of a resistor and power supply connected to the rest of the circuit.

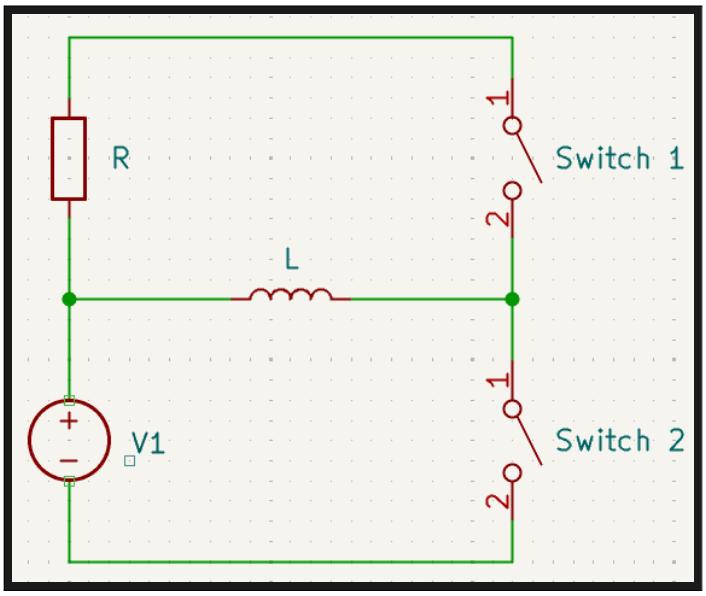


Figure 3.1: Test setup for measuring steady state waveforms.

The same setup was used to evaluate the functionality of the cell balancing circuit by measuring the MOSFET's gate to source voltages, V_{gs}, and drain to source voltages, V_{ds}, with an oscilloscope, making sure they behaved as expected based on the Arduino's PWM signals. The current and PI controller were evaluated by activating the PWM signals after connecting the power supply and resistor. This allowed for evaluation of the PI controller by measuring inductor current, I_L, rise time and error margins to the reference current, I_{ref}. This was done using both a current probe and PCB mounted current sensor connected to an oscilloscope. These tests using the setup shown in Figure 3.1 were also conducted by interchanging the power supply and resistor to test both directions of charge transfer. The voltage output from the current sensor was translated to a current using

$$I_{sensor} = 2.5 \left(V_{sensor} - \frac{V_{supply}}{2} \right) \tag{3.2}$$

where V_{sensor} is the sensor output voltage and V_{supply} is its supply voltage [42]. The control system was adapted by evaluating its ability to control the current to the desired level without causing overshoots and oscillations. The same method was used for measuring steady state current I_L, having waited for the current to reach the reference.

The setup in Figure 3.1 was also used to evaluate the efficiency of the balancing circuit by measuring the energy losses incurred when transferring charge in the circuit. The balancing efficiency was determined by measuring the power dissipated across the resistor and dividing it by the power output from the power supply as

$$\eta_{experimental} = \frac{V_R I_R}{V_{V1} I_{V1}} \tag{3.3}$$

where V_{V1} and I_{V1} are the voltage and current output from the power supply, respectively, and V_R and I_R are their resistor counterparts. Moreover, the efficiency was determined theoretically by approximating the losses as

$$P_{loss,theoretical} = R_L I_L^2 + R_{ds,on,n} I_{V1}^2 + R_{ds,on,p} I_R^2 \tag{3.4}$$

in which R_L, $R_{ds,on,n}$, and $R_{ds,on,p}$ are the resistances in the inductor, the on resistance for the NMOS and on resistance in the PMOS, respectively. Estimates of the resistances were retrieved from the respective data sheet as $R_L = 660$ mΩ and $R_{ds,on,n} = 50$ mΩ and $R_{ds,on,p} = 100$ mΩ [43], [44], [45]. For the NMOS, the I_D versus $R_{DS}(on)$ characteristics were used for $V_{gs} = 4.5$ V and $I_D \approx 1$ A. For the PMOS, the $-I_D$ versus $R_{DS}(on)$ characteristics were used, with V_{gs} = -4.5 V and $I_D \approx$ -1 A, since the PMOS's body diode was used with the test setup as in Figure 3.1, thus giving negative I_D. The theoretical efficiency was then calculated as

$$\eta_{theoretical} = \frac{V_{V1} I_{V1} - P_{loss,theoretical}}{V_{V1} I_{V1}} \tag{3.5}$$

Further evaluation of the efficiency was conducted using the LTspice simulations in the same way as in (3.3). Moreover, the causes of power loss were studied by measuring the average power dissipation in the MOSFETs and the inductor in LTspice. Having determined the power loss for these components, the difference between the total power loss and the sum of the power losses in the individual components

$$\Delta P_{loss} = P_R - \left(P_{V1} - \sum_{i=1}^{3} P_i \right) \tag{3.6}$$

was calculated, in which P_{V1} was the power from the voltage source and P_R the power dissipated in resistor R in Figure 3.1. The powers P_i are the power dissipation in the inductor, NMOS and PMOS. Accordingly, (3.6) determined if these components were the causes of power loss. Then, the resistance for each component i was estimated as

$$R_{estimated,i} = \frac{P_i}{I_i^2} \tag{3.7}$$

for the currents I_i associated with each value of P_i. The resistance values were then compared to their respective data sheet for further evaluation.

Another setup was used to verify the circuits ability to transfer charge between battery cells, and that there was no large current spikes at the start of the balancing process. This setup is shown in Figure 3.2.

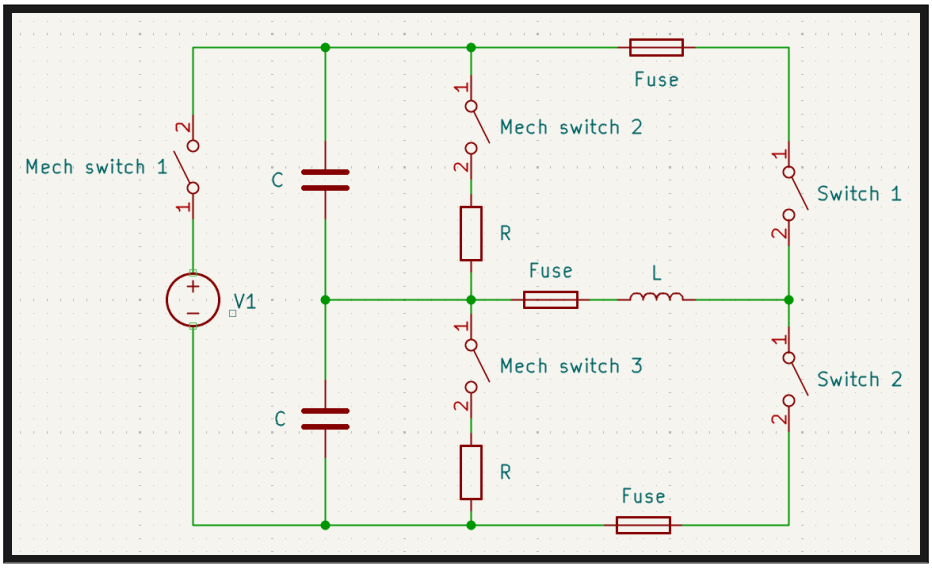


Figure 3.2: Test setup using capacitors as battery cells.

The test setup shown in Figure 3.2 mimics one with real battery cells, but the battery cells have been replaced with capacitors. Capacitors were used due to their inherent ability to store electrical charge, much like batteries, but in smaller amounts [29]. The capacitors are also not as volatile as lithium-ion batteries, which allowed safer testing to precede the tests conducted with lithium-ion batteries [39].

Figure 3.2 also featured a power supply, resistors, and fuses. The power supply, $V1$, served to charge the capacitors by closing the mechanical switch, "Mech switch 1", and the resistors could be connected to the capacitors by closing either of the mechanical switches, "Mech switch 2" or "Mech switch 3", which served to create a charge imbalance between the capacitors. The fuses were added for protection against high current spikes. No measurement data was retrieved from the capacitor tests, as they only served to test the possibility of transferring charge and the ability of the control system to avoid over currents during start up. This was done by observing the variations of capacitor voltage when operating the circuit and measuring the current with an oscilloscope.

Building on the setup in Figure 3.2, the power supply and capacitors were exchanged with battery cells to test balancing. Doing so yielded Figure 3.3.

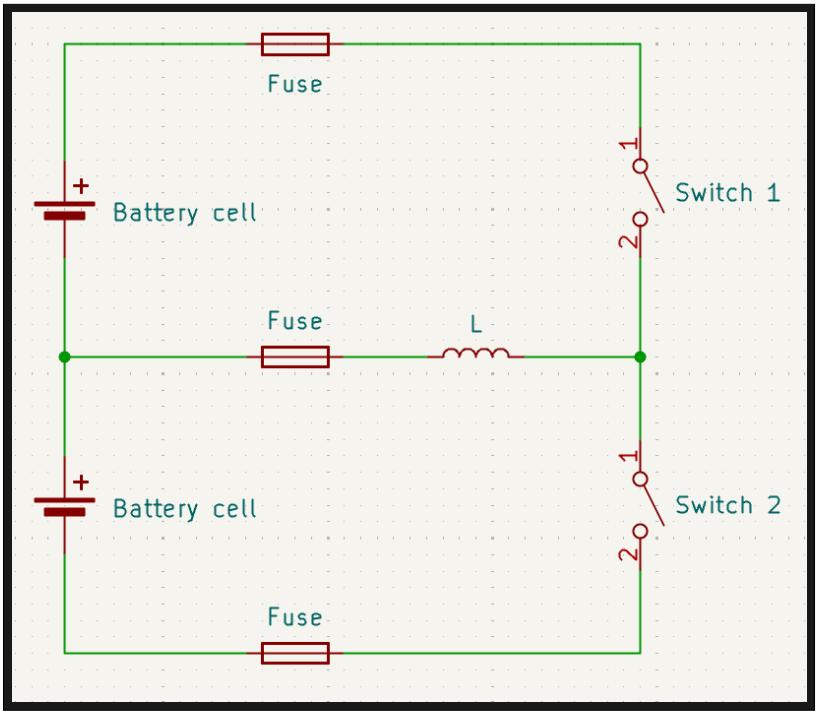


Figure 3.3: Test setup using lithium-ion battery cells.

Using battery cells, the results from testing with capacitors were substantiated by observing how battery cell voltages varied during circuit operation for several minutes. Furthermore, the long charge/discharge times made it possible to test the control system's ability to control the inductor current when using battery cells. Testing was conducted in both current directions, letting both cells discharge into the other. Due to the risk of short circuiting the cells, the testing was conducted only switching one of the MOSFETs. The battery cell voltages were measured and logged using the Arduino, and the same was true for the current, which the Arduino translated from voltage using (3.2). Furthermore, the open circuit voltage (OCV), when the battery cells were not charging or discharging, was measured before and after the tests using a multimeter. These were then translated to SoC before and after the tests using Appendix E, Figure E.1 (provided by S. Lundberg, personal communication, February 7, 2025).

The balancing speed in hours was determined as

$$t_{balancing} = 0.035 \frac{\Delta SoC_0}{I_{ref}} \tag{3.8}$$

where ΔSoC_0 is the initial SoC difference between the battery cells. The factor 0.035 converts ΔSoC_0 to decimal form and accounts for the current's relation to the maximum battery capacity of 3.5 Ah [14].

4

Topology selection and circuit design

This chapter contains the Pugh matrix reviewing the different balancing topologies and choosing the one best suited for this project. Following this is the hardware design of the necessary circuitry for its physical implementation. The circuit design was split into several distinct parts during the design process. This allowed for an easy overview of their functionality, and each of the parts is explained in detail in the sections following the topology selection. Excluding the battery cells, any components appearing without attached values were extra filter components which were added in case the circuit would be too noisy during operation. A full overview of the PCB and schematic design is shown in Appendix A in Figure A.1 and Figure A.3 respectively.

4.1 Topology selection

In Table 4.1, the different topologies have been applied to a modified Pugh matrix. The weights of each property are written in parentheses in the top row. The points are multiplied by their corresponding weight and summarized in the "Sum" column. The scores of each property is based on the advantages and disadvantages presented in Table 2.1.

Table 4.1: Modified Pugh decision matrix used on the different topologies.

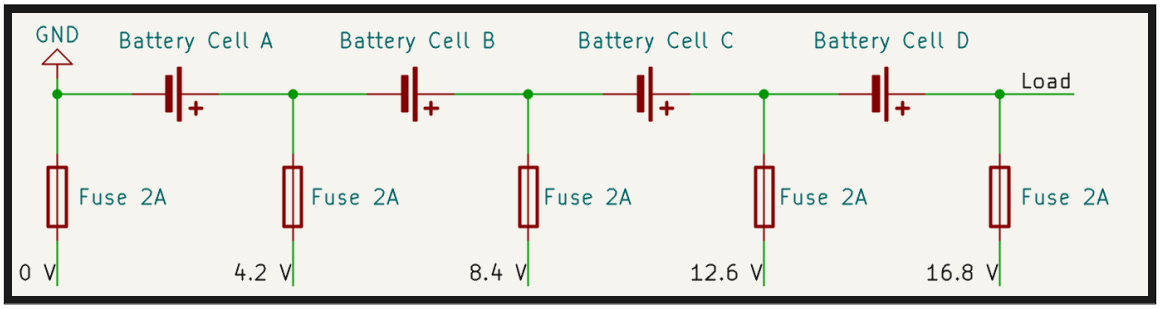
Topology	Balancing time (2)	Efficiency (2)	Complexity (2)	Size (1)	Cost (1)	Sum
Fixed resistor	0	0	0	0	0	0
Switched shunt resistor	+1	0	-1	-1	-1	-2
Multiple switched capacitor	+1	+1	-1	-1	-1	0
Single switched capacitor	+1	+1	-1	-1	-1	0
Switched transformer	+3	+2	-2	-3	-3	0
Multi-winding transformer	+3	+2	-1	-3	-3	2
Multiple switched inductor	+3	+2	-1	-2	-1	5
Single switched inductor	+3	+2	-2	-1	-1	4
Flyback converter	+2	+1	-2	-2	-2	-2
Cûk converter	+2	+2	-2	-2	-2	0

According to the modified Pugh matrix, the multiple switched inductor had the

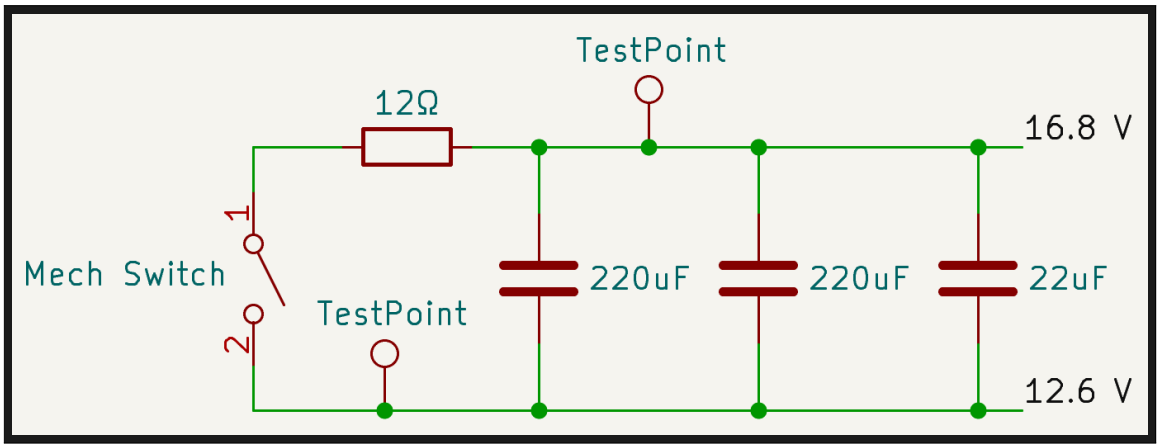
overall best performance, and was therefore the topology selected for further investigation.

4.2 Battery fixture and ripple filter

The circuit was designed for a maximum of four 4.2 V lithium-ion battery cells, giving the battery pack a max voltage of 16.8 V. The cells were not mounted on the PCB, but kept separate for safety reasons. This setup can be seen in Figure 4.1a, where the 2 A fuse between each battery cell and the circuit was a safety feature to ensure that the current did not exceed what the cells and the circuit could safely manage. The voltage labels in Figure 4.1a (and other schematics) is a feature of KiCad used to simplify schematic design. Each label of a value in a schematic is connected to the same value in every other schematic.



(a) Battery setup with safety fuses.



(b) Smoothing and filtering capacitors along with mechanical charge manipulation circuitry

Figure 4.1: Schematic of the external battery pack and anti ripple voltage capacitors

Each battery cell was connected in parallel to 12 Ω resistors and three capacitors with an equivalent capacitance of 462 μF as shown in Figure 4.1b. The figure had the input points 16.8 V and 12.6 V connected from battery cell D in Figure 4.1a. Two types of capacitors were used; the 220 μF capacitors were electrolytic capacitors that

served as smoothing capacitors that lowered the damaging ripple voltage otherwise experienced by the battery cells. The capacitance of the electrolytic capacitors was decided based on the results from the Simulink simulations. Furthermore, the 22 μF capacitor was a ceramic capacitor, which meant it had lower series impedance than the electrolytic capacitors; its purpose was therefore to reduce high frequency noise [46], [47]. The 12 Ω resistors were used to create an imbalance between the battery cells, to enable testing of the circuit's balancing properties. This imbalance was created by activating the series connected mechanical switch, which allowed the user to discharge each cell individually.

4.3 Charge distribution circuit

The inductors were connected between each pair of battery cells along with two MOSFETs, serving as switches. This configuration is shown in Figure 4.2 and was repeated for each pair of cells. In Figure 4.2, the circuit connected between the topmost battery pair is shown, and is connected to the points 16.8 V, 12.6 V and 8.4 V. The switching of the MOSFETs was controlled by PWM signals generated by the Arduino. In total, three charge distribution circuits like the one shown in Figure 4.2 were needed for a four cell circuit.

Each of these circuits was connected to the top (16.8 V in Figure 4.2) and bottom (8.4 V) points of the cell pair, in addition to the midpoint (12.6 V) between the two neighboring cells. In a circuit containing four battery cells, there would be three ways of creating cell pairs; thus, three circuits shown in Figure 4.2 would be required.

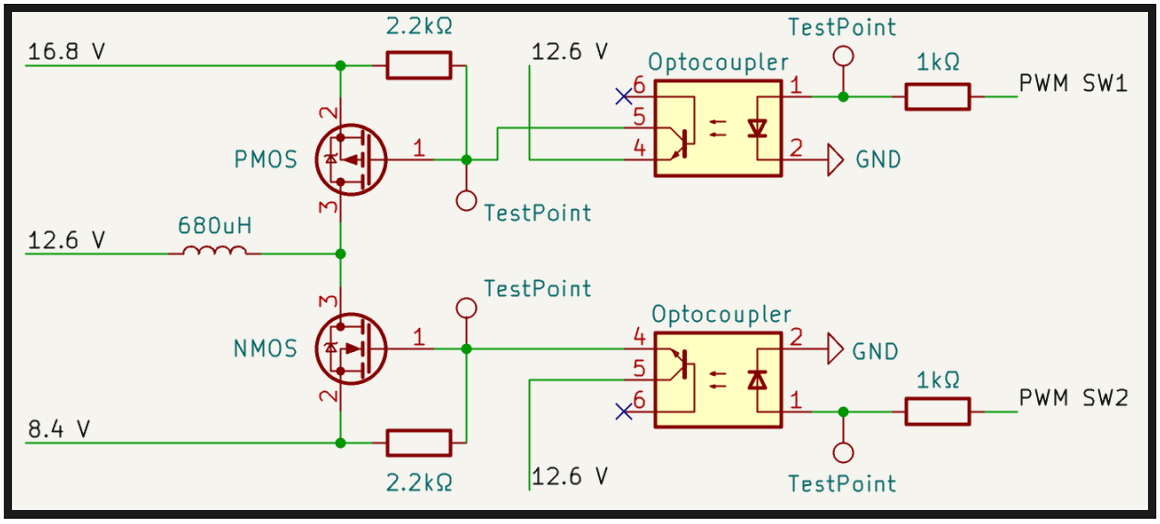


Figure 4.2: Schematic of the charge distribution circuit.

As shown in Figure 4.2 the Arduino output ports, PWM SW1, and PWM SW2 were connected in series with a 1 kΩ resistor and an optocoupler. The 1 kΩ resistor limited the current that could be pulled from the Arduino output ports, minimizing

the risk of breaking the Arduino ports. Furthermore, the output voltage of the Arduino was 5 V, meaning that the maximum current that could be pulled from the Arduino was 5 mA, given the 1 kΩ resistor. This was also was sufficient for activating the light emitting diode (LED) in the optocoupler [48].

The optocouplers in Figure 4.2 enabled control of the MOSFETs with the PWM signals from the Arduino. When PWM SW1 was in its high state, the LED in the optocoupler turned on. The emitted light would then be detected by the phototransistor on the other side of the optocoupler, which would cause it to turn on and, in turn, connect the gate (labeled 1 in Figure 4.2) of the PMOS to the 12.6 V node. The source of the PMOS (labeled 2 in Figure 4.2) was connected to 16.8 V, which meant the gate to source voltage would be -4.2 V, opening the PMOS. The NMOS worked similarly to the PMOS and was instead controlled by the PWM SW2 port. When the PWM SW2 was high, the gate of the NMOS was connected to the 12.6 V node, which meant the gate to source voltage would become 4.2 V, causing the NMOS to turn on. A 2.2 kΩ resistor was connected between gate and source of each MOSFET to allow the accumulated gate charge to discharge when the MOSFET was switched off. Making sure the gate was fully discharged before the next switching period was crucial since it ensured that the MOSFET had switched off; otherwise, there would be a risk of short-circuiting the battery cells.

When charge was being redistributed, the corresponding MOSFETs were switched according to the desired current level and direction. If, for example, the lower battery cell in Figure 4.2 was being discharged into the upper, the NMOS was switched to increase the current, and when it was off, the current flowed through the body diode of the PMOS to charge the upper battery cell. Since the current flows through the body diode, switching of the PMOS was not necessary for the upper cell to charge. However, it was switched out of phase with the NMOS, thus reducing V_{ds} of the PMOS to limit power losses.

Moreover, the selected 680 μH inductance had a saturation current of 1.3 A, which was higher than the selected balancing current of 0.68 A, meaning the inductor would not reach saturation. Furthermore, the inductance also affected the size of the ripple current around the average inductor current, as explained in Section 3.2. Using (3.1) yielded a 300 mA ripple current, assuming 50% duty cycle, $V_L = 4$ V, $L = 680$ μH and a 10 kHz switching frequency. This switching frequency gave on times D_1T and D_2T, with $T = 100$ μs, for the MOSFETs. The on times were long enough for the MOSFET gates to discharge during each switching period, while still being short enough for the ripple current to be at an acceptable level. Figure 4.3 illustrates the inductor voltage and current for three switching periods, simulated using LTspice.

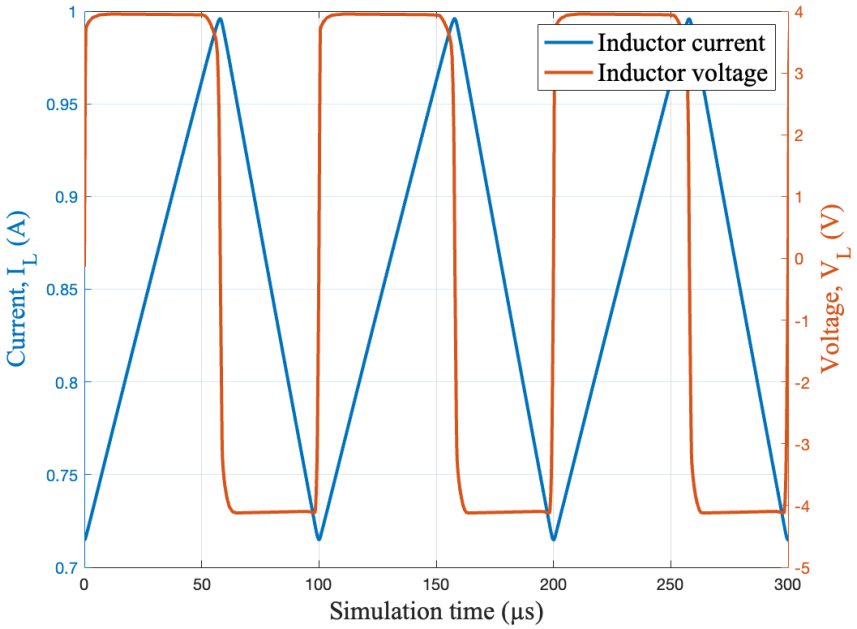


Figure 4.3: Inductor voltage V_L and inductor current I_L from LTspice simulations over three switching periods.

The inductor voltage alternated between ± 4 V and the current had a triangular shape with ripple close to 300 mA, as expected. Furthermore, the simulations showed that the circuitry driving the MOSFETs worked as intended and thus provided the desired validation of the charge distribution circuit.

4.4 Measurement circuits

To use the Arduino for voltage measurement, one input port per cell was used. The circuit for cell D is shown in Figure 4.4. The input port was rated for a maximum of 5 V, meaning battery cells B, C, and D each had voltage dividers that limited the input voltage to a level the Arduino can handle [49]. The voltage dividers in Figure 4.4 divided the potentials by four, while the voltage dividers for battery cells C and B divided by three and two. Battery cell A did not require a voltage divider since it had a voltage under 5 V.

The dotted blue line seen in Figure 4.4 was one of two possible connections that could be made to measure the voltage level of a cell. When using the dotted connection, none of the components bypassed by the blue line were mounted to the board. The Arduino was instead used to calculate battery cell voltage directly from the input on the port.

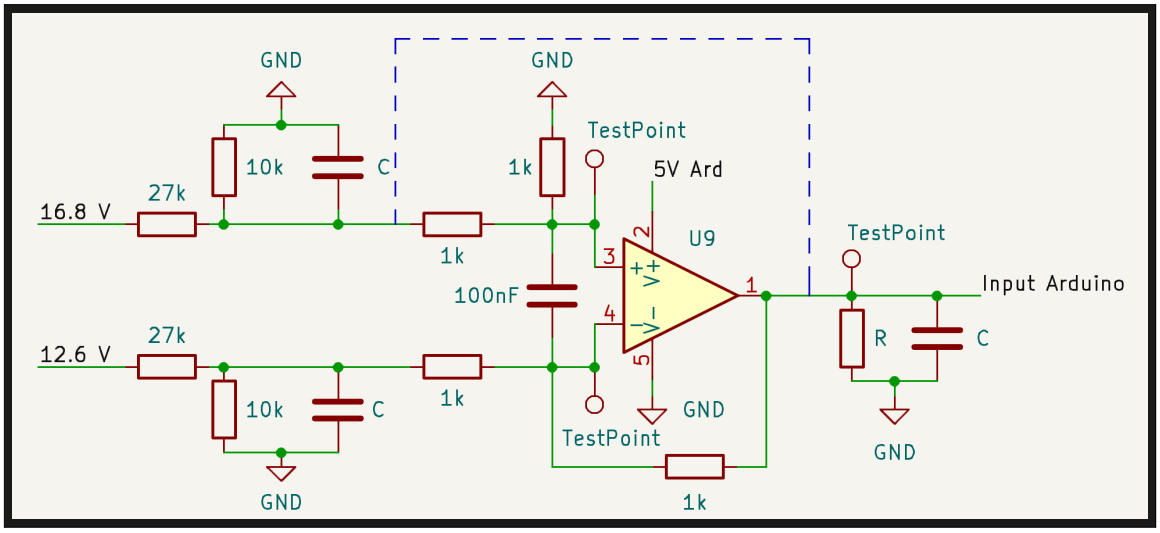
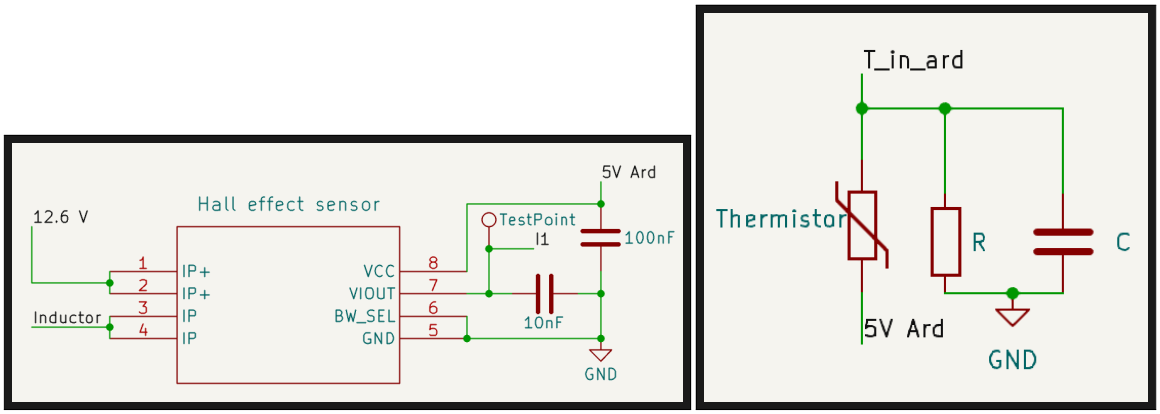


Figure 4.4: Schematic of battery voltage measurement circuit.

The second method of measuring the battery cell voltage uses the differential amplifier shown in Figure 4.4. The differential amplifiers had equal resistor values, which made the operational amplifiers (OP amps) subtract the potentials without any gain. Capacitors were placed between the inputs of the OP amps to compensate for the inductive load. From the differential amplifier, the voltage measurement was delivered to the Arduino. Furthermore, extra capacitors and resistors were added to the circuit to function as filters for the voltage dividers and OP amp outputs to reduce noise from the circuit. None of these were added to the physical board since none of the signals in the circuit were noisy enough to influence measurements.

To measure the inductor current, a Hall effect sensor was connected in series with each inductor, as seen in Figure 4.5a. Since the current flowed bidirectionally, based on the direction of charge distribution, and for the sensor not to affect the current, a Hall effect sensor capable of operating with these restrictions was chosen. The

sensor was rated for ±5 A and had a sensitivity of 400 mV/A. Ideally, the higher the sensitivity, the better the performance of the control system due to more accurate input values, which in turn would lead to a higher-performing balancing circuit. Two capacitors were also connected from the current output and power input to ground, per the manufacturer's specifications [42].



(a) Current sensor **(b)** Temperature sensor

Figure 4.5: Schematics of the current and temperature circuits.

As a safety feature, the battery temperature could be measured while the balancing circuit was in operation, using a thermistor. A thermistor is a semiconductor component whose resistance varies with temperature. The thermistor was connected to the Arduino as seen in Figure 4.5b. Should the temperature exceed the safe operating temperatures specified in the data sheet for the battery cells, the balancing circuit would immediately shut off. The capacitor and resistor are extra components which could be used for filtering in case the signal was too noisy to be used for an accurate measurement. These were not connected to the physical board.

4.5 Arduino and control system

The control system implementation included safety features, protecting against thermal runaway and over/under voltage, and a PI controller responsible for making the inductor current $I_{L,avg}$ approach the reference current I_{ref}, as described in Section 2.9. $I_{L,avg}$ was determined by the Arduino by calculating the average current over 100 samples, taken during one switching period.

The PI current controller in Figure 2.12 was implemented in the Arduino using the transfer function in (2.34). The bandwidth $\alpha_c = R_a L^{-1}$ of the modified process model in Figure 2.11 was optimized by setting $R_a = 0.2\ \Omega$. This was empirically determined to be optimal for maintaining stability while making the control system as fast as possible. Hence, given $L \approx \hat{L} = 680\ \mu$H the bandwidth was $\alpha_c \approx 294$ Hz. Moreover, the proportional gain $K_p = \alpha_c \hat{L} \approx 0.2\ \Omega$ and integral gain $K_i = \alpha_c R_a \approx 58\ \Omega\text{s}^{-1}$.

The program, which is shown in Figure 4.6, was initialized by the ”Start” block, followed by ”Open all switches” that prevented the cells from short-circuiting. Moving on, the ”Parameter measurements” gathered the SoC and cell temperature information. Then the ”Safe?” block checked if the given parameters were within operating conditions, and if any of them exceeded the safe limits, the program jumped to ”Open all switches” and was terminated by the ”Stop” block. Otherwise, the program checked if the battery cells were balanced by comparing each individual SoC with the average SoC using a predefined tolerance. If all the cells were balanced, the cycles above repeated from ”Open all switches” onward, and otherwise, a MOSFET pair was chosen in the ”Decide MOSFET pair” block according to which neighboring cells were the most unbalanced.

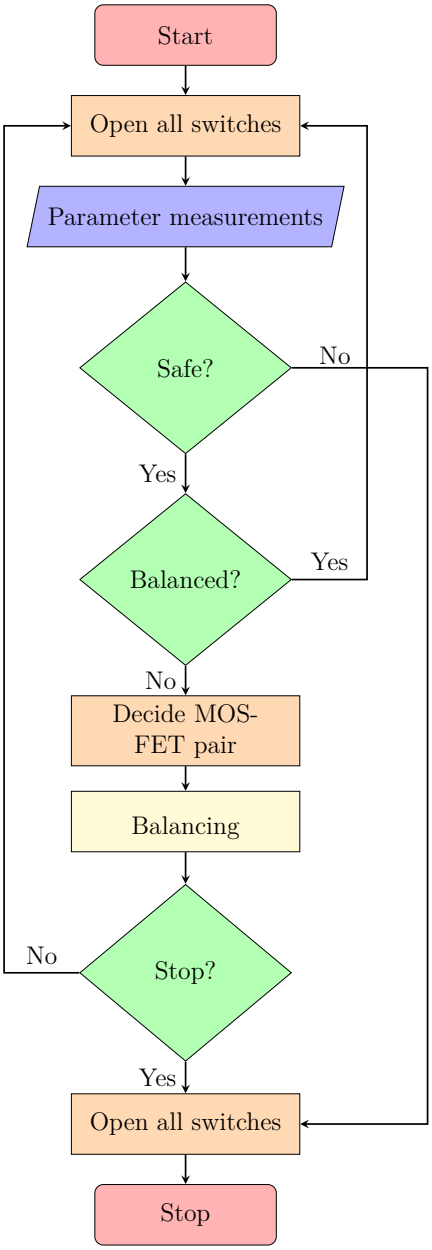


Figure 4.6: Overview of suggested algorithm for the Arduino MEGA 2560 R3

Having selected MOSFET pair in the ”Decide MOSFET pair” block, the balancing subroutine illustrated in Figure 4.7 was called. This was responsible for balancing the selected battery cells until their SoCs were within a given tolerance.

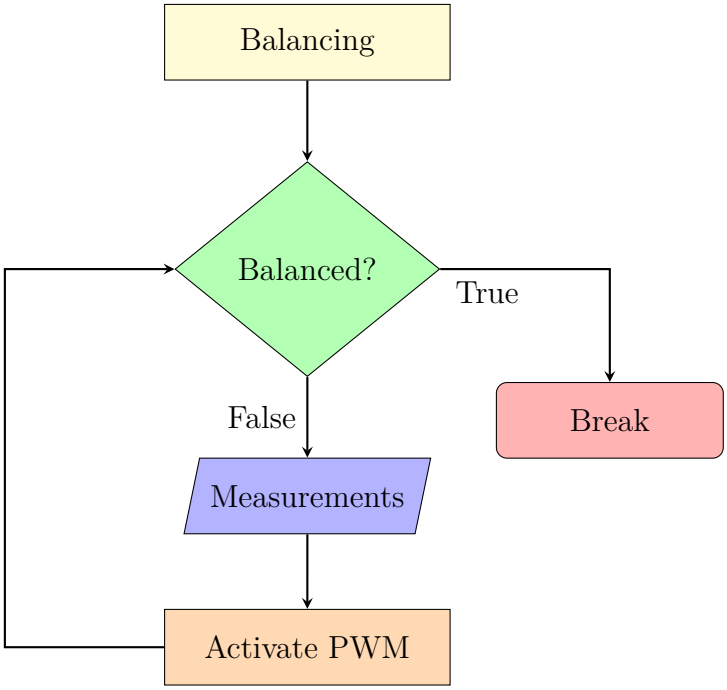


Figure 4.7: Subroutine of the algorithm used in the Arduino MEGA 2560 R3

The balancing subroutine started by checking if the two battery cells' SoC were within the tolerance. If they were balanced, the loop went to ”Break” and returned to the main program in Figure 4.6. If not balanced, new measurements were conducted from the I/O ports. This included SoC for the battery cells and inductor current. Based on these, a new duty cycle was determined by the PI controller. The PWM signals activated by ”Open all switches” run in parallel to the program and are only disrupted by the ”Open all switches” or ”Activate PWM” blocks.

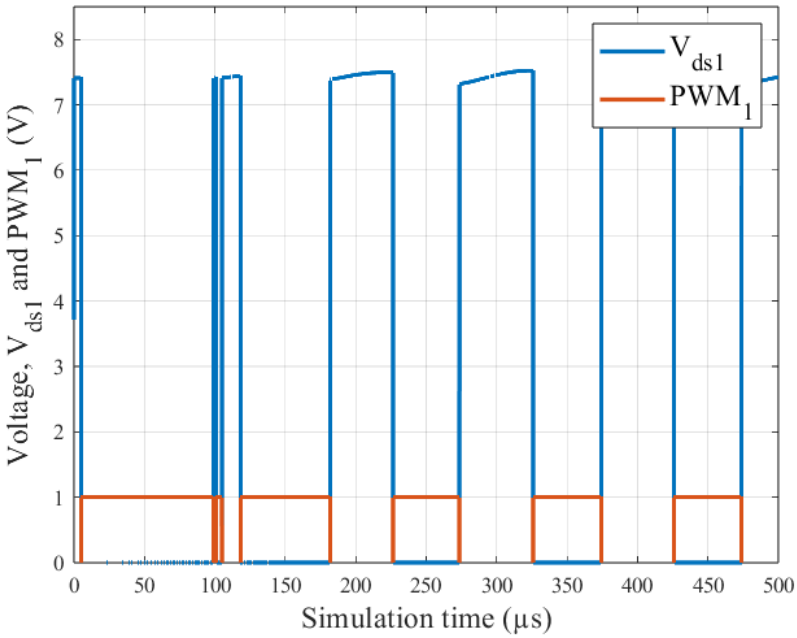
5

Simulation and prototype results

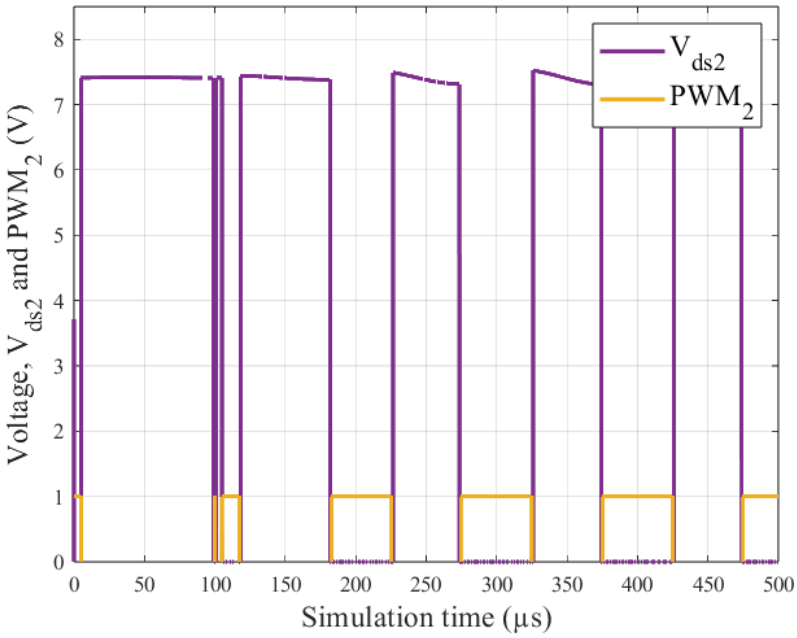
This chapter contains the simulations from Simulink and measurements from the prototype. It illustrates how the balancing circuit performed, both in a mostly ideal case with Simulink and with the prototype. Furthermore, the balancing performance was analyzed in terms of balancing speed and efficiency.

5.1 Simulink results

In this section, the Simulink simulation results are presented. All results were taken with the setup in Appendix D Figure D.1. Figures 5.1a and 5.1b show the drain to source voltage over MOSFETs one and two, along with their corresponding PWM signals.



(a)



(b)

Figure 5.1: The drain-to-source voltage across the MOSFETs and their corresponding PWM signals. (a) corresponds to switch 1 and (b) to switch 2 in Figure 2.6.

As seen in Figure 5.1a and 5.1b, the MOSFETs turned on when their corresponding PWM signals, PWM_1 and PWM_2, were in the high state, meaning that the MOSFETs drain to source voltage, V_{ds1} and V_{ds2}, fell to zero. When the PWM signals

were in their low state, the MOSFETs turned off and V_{ds} became equal to the sum of the battery voltages. Figure 5.2 shows the inductor voltage when MOSFETs one and two were switching.

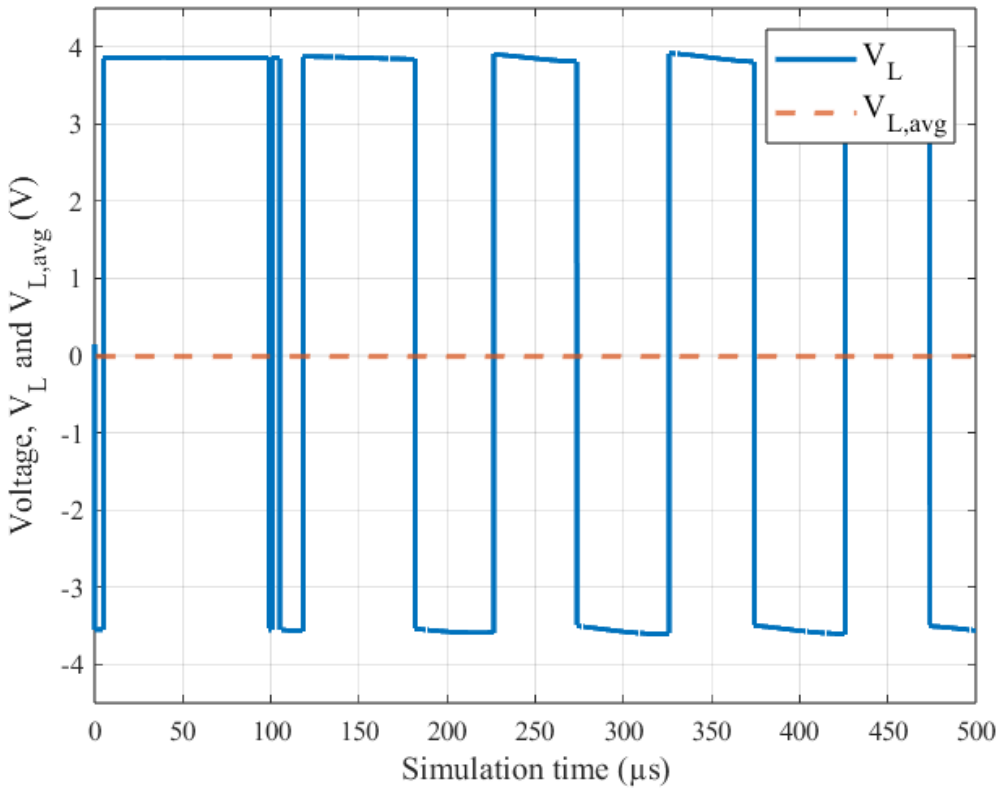
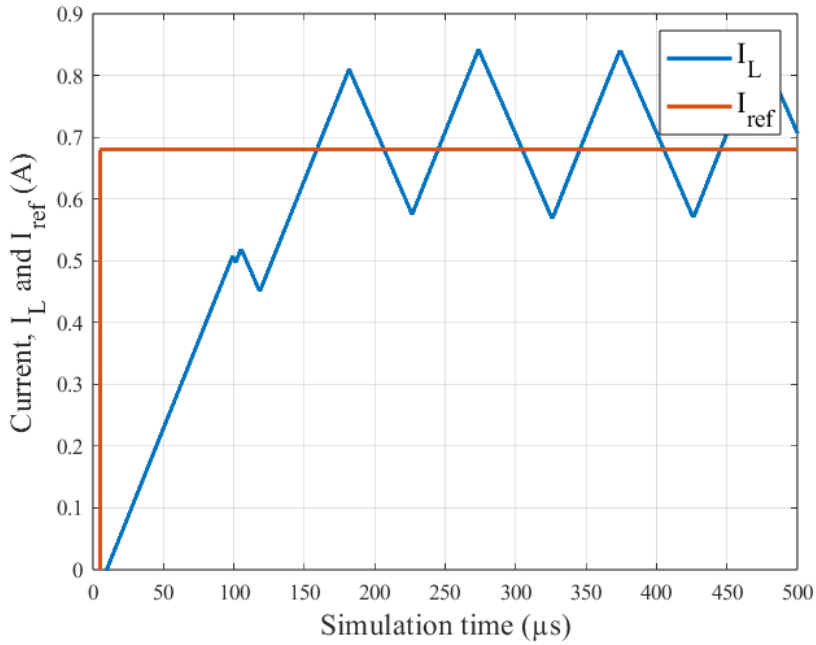
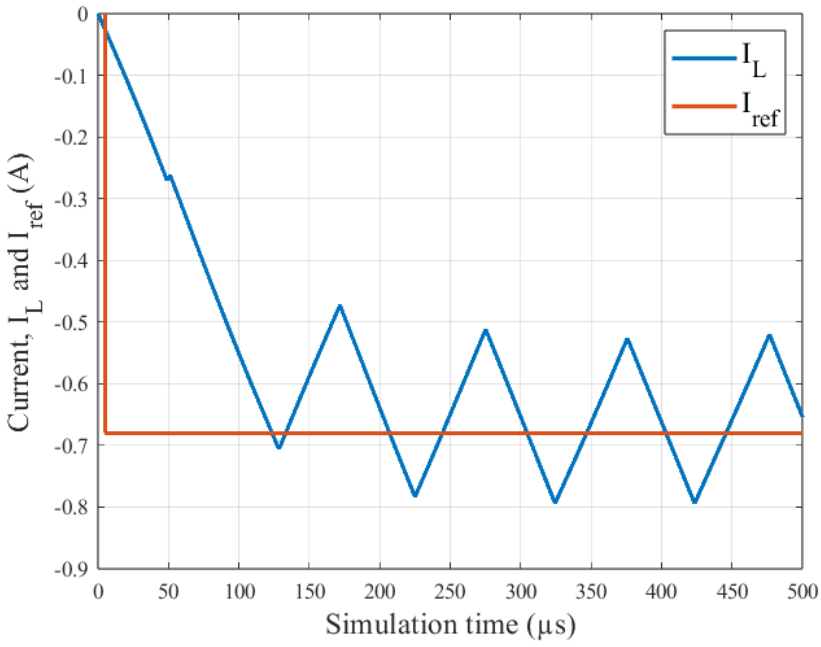


Figure 5.2: The voltage over the inductor, V_L, along with the period-time-average of the inductor voltage, $V_{L,avg}$.

As seen in Figure 5.2, the inductor voltage mirrored the voltage in Figure 5.1. When PWM_1 was high, the voltage over battery cell A was applied across the inductor. Similarly, the negative of the voltage across battery cell B was applied across the inductor when PWM_2 was high. The inductor current during switching is shown in Figure 5.3.



(a)

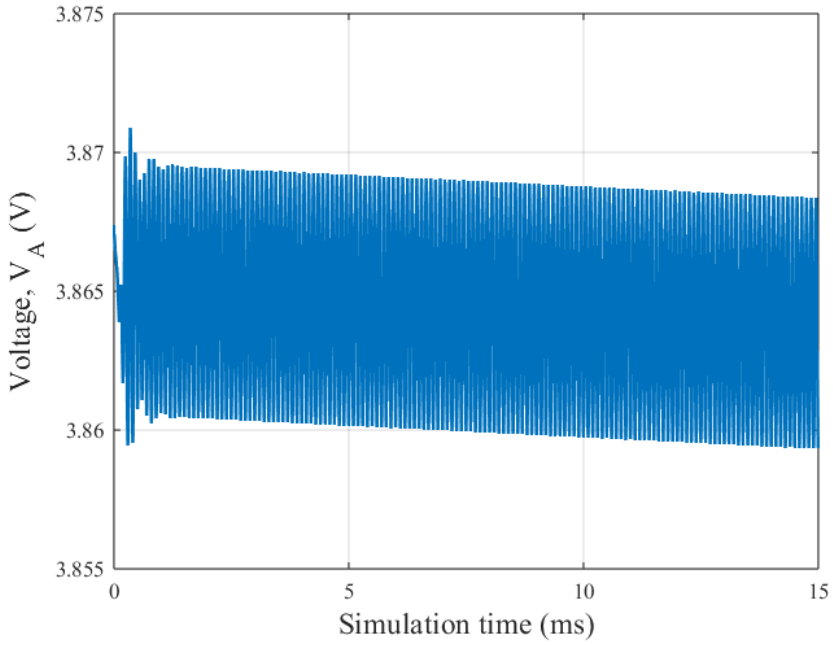


(b)

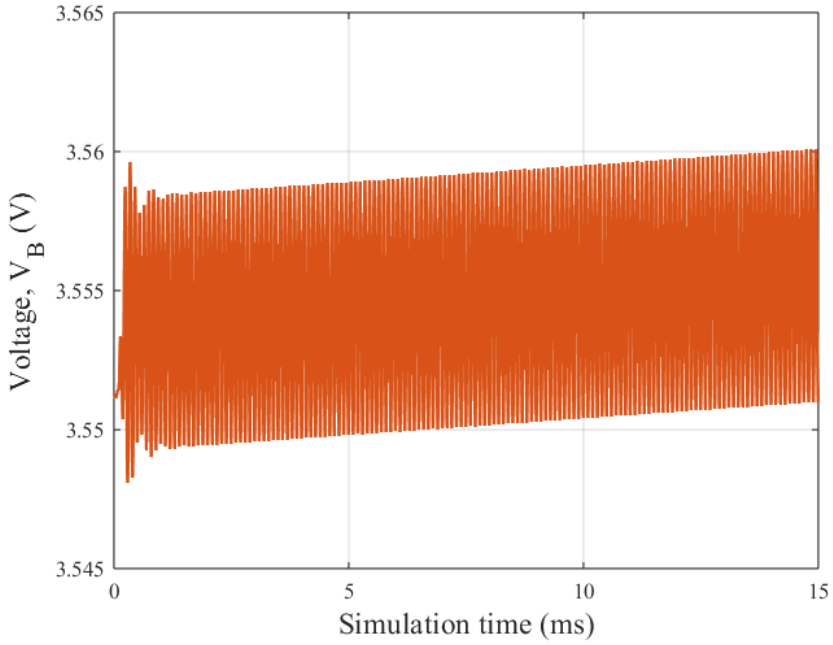
Figure 5.3: The current through the inductor settling around its reference current. The reference for (a) was 0.68 A, and -0.68 A for (b).

Figure 5.3 shows both positive and negative inductor current since the sign of I_{ref} was determined by the control system depending on the direction of charge distribution. When V_A was higher than V_B, the reference was positive as in Figure 5.3a and when V_B was higher than V_A, the reference was negative as in Figure 5.3b.

Moreover, with the current as in Figure 5.3a, the voltage across the battery cells changed as in Figure 5.4.



(a)



(b)

Figure 5.4: The voltage over battery cell A and B over a simulation time of 15 ms. In (a) the voltage over battery cell A is seen, and (b) shows the voltage over battery cell B.

As Figure 5.4 illustrates, V_A decreased and V_B increased, meaning charge from battery cell A was moved to battery cell B, since $V_A > V_B$. Similar results with reversed slopes were obtained when $V_B > V_A$.

5.2 Prototype results

This section presents the measurements from the physical prototype. Two setups were used to measure the results in this chapter: the setup shown in Figure 3.1 and the one in Figure 3.3. Figure 5.5 shows the steady-state drain to source voltage of the lowest MOSFET and its corresponding PWM signal measured using an oscilloscope with the setup in Figure 3.1.

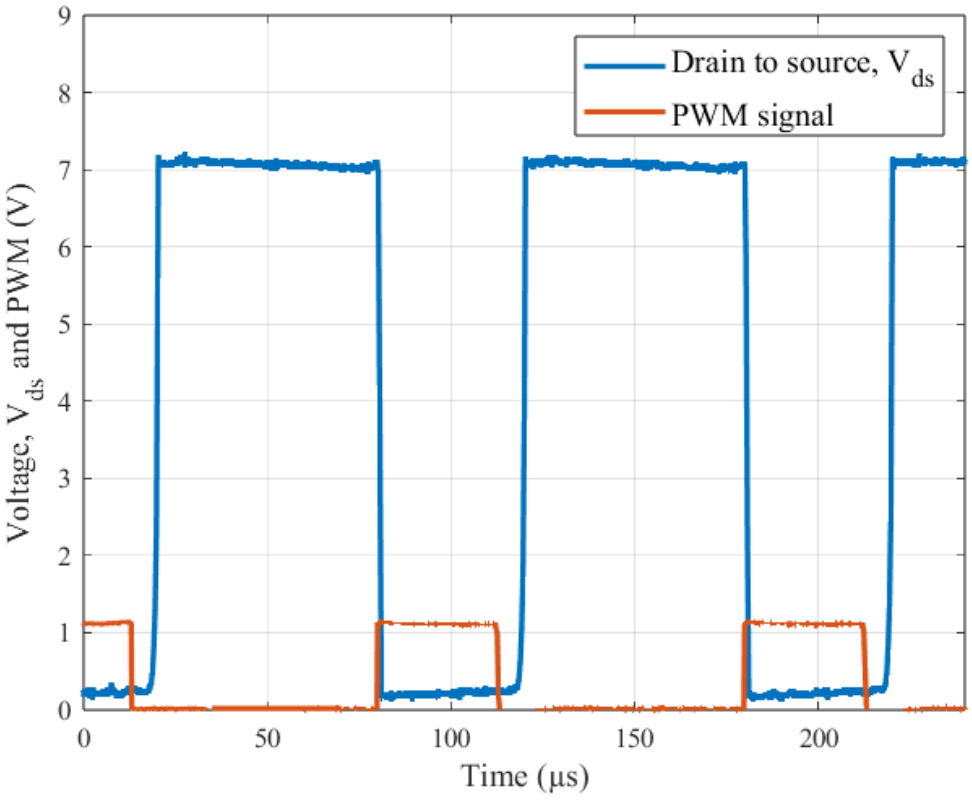


Figure 5.5: The drain-to-source voltage V_{ds} across the lowest MOSFET along with its corresponding PWM signal.

Figure 5.5 shows that the MOSFET turned on when the PWM signal was high, meaning that the drain to source voltage dropped to near zero. Similarly, the MOSFET turned off when the PWM signal was low, meaning that the drain-to-source voltage rose to the sum of the power supply and inductor voltage. Similar results were obtained for the remaining MOSFETs by activating their corresponding PWM ports. The inductor voltage during the switching is shown in Figure 5.6, and was measured using the setup in Figure 3.1 and an oscilloscope.

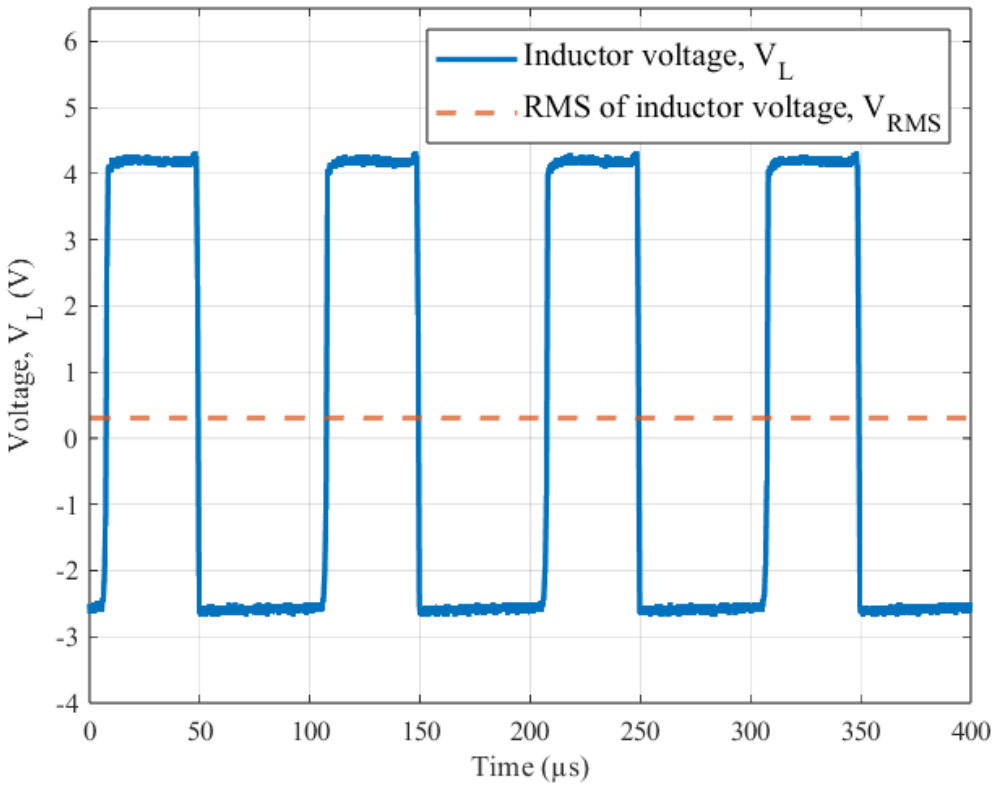


Figure 5.6: The voltage, V_L, over the inductor along with its RMS voltage, V_{RMS}.

The inductor voltage in Figure 5.6 toggled between positive and negative during the switching. Furthermore, the inductor current measured using the setup in Figure 3.1 and a current probe connected to an oscilloscope can be seen in Figure 5.7.

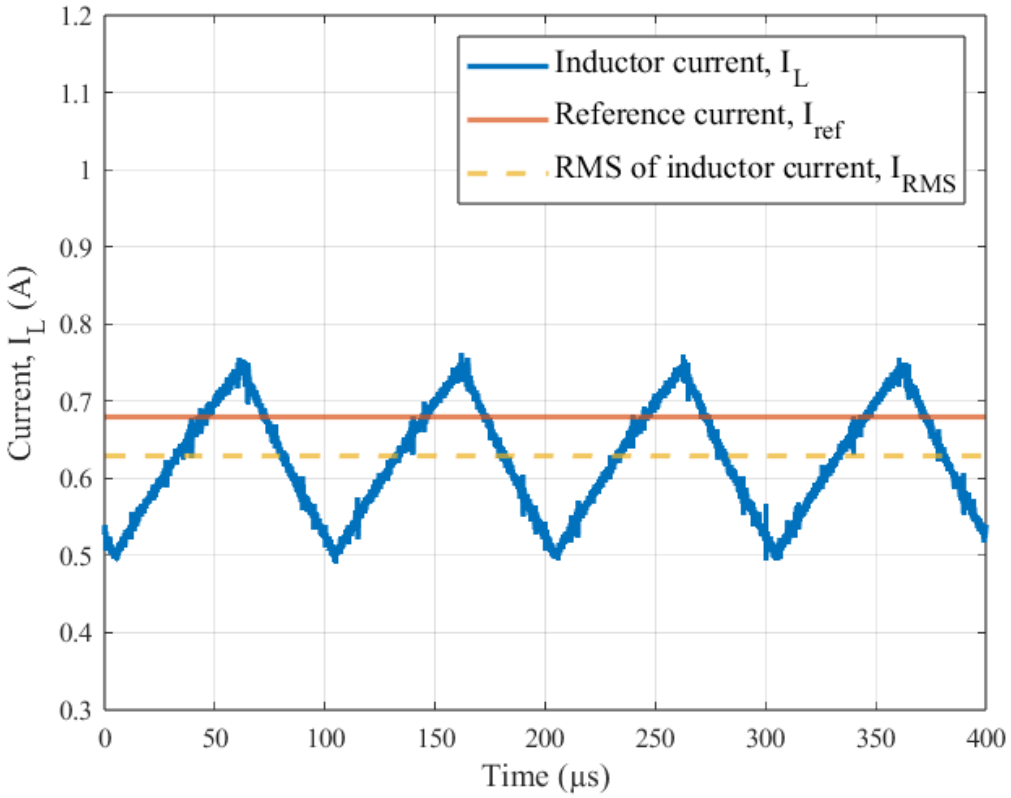


Figure 5.7: Steady state current, I_L, through the inductor measured with a current probe along with its RMS current I_{RMS}, and the reference current I_{ref}.

Another steady state current measurement conducted at the same time was done using the PCB mounted current sensor, and can be seen in Figure 5.8.

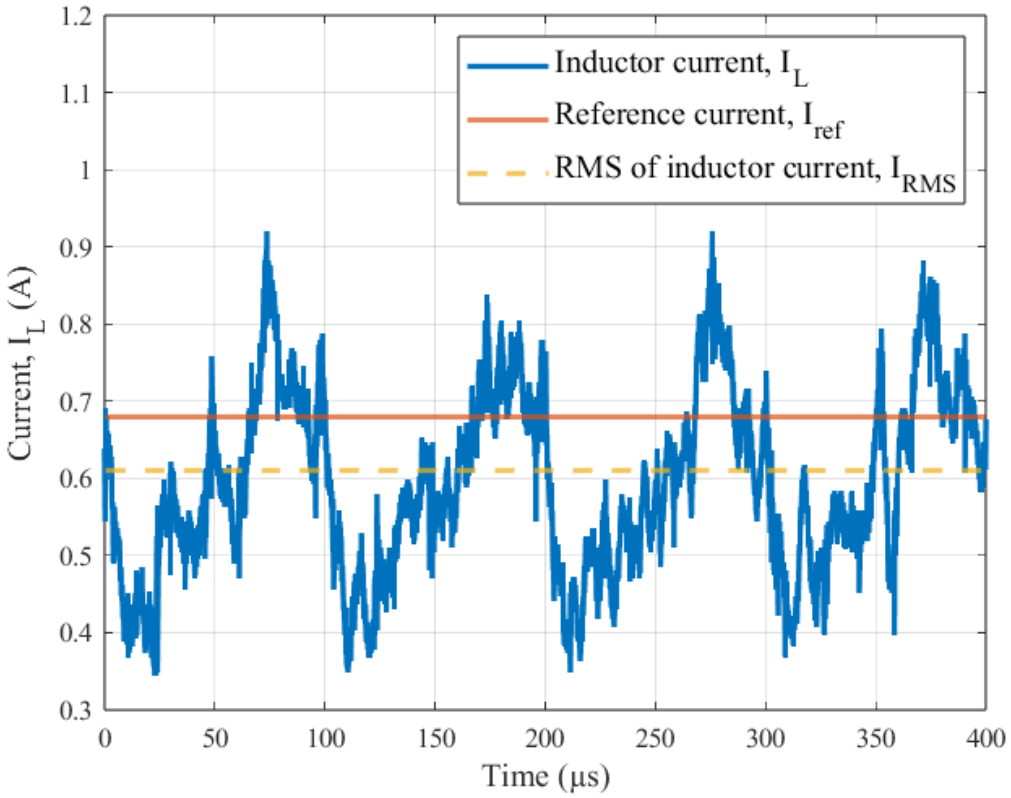
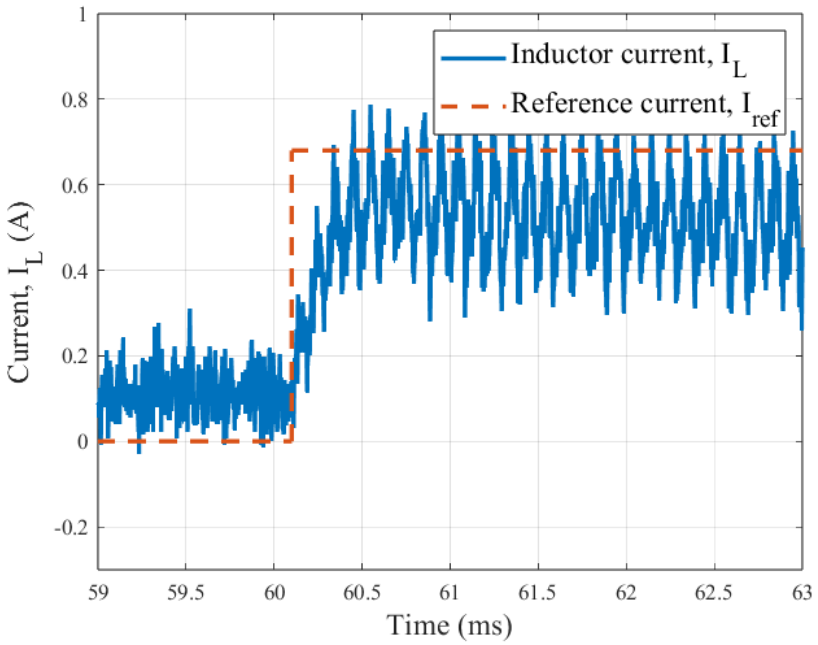


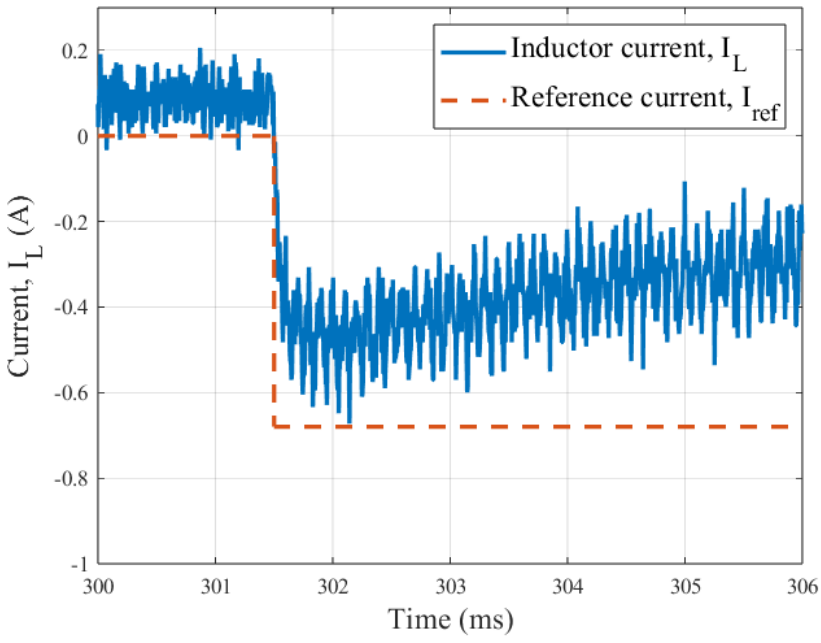
Figure 5.8: Steady state current, I_L, through the inductor measured with the current sensor along with its RMS current I_{RMS}, and the reference current I_{ref}.

The voltage output from the PCB mounted sensor was measured with an oscilloscope and translated to a current according to (3.2). The current graphs in Figures 5.7 and 5.8 differ while they have similar RMS currents.

Further current measurements were conducted to illustrate the step response of the PI controller, having the current start from zero and then increase to the reference current I_{ref}. These graphs are shown in Figure 5.9, and were obtained using the setup shown in Figure 3.1.



(a)



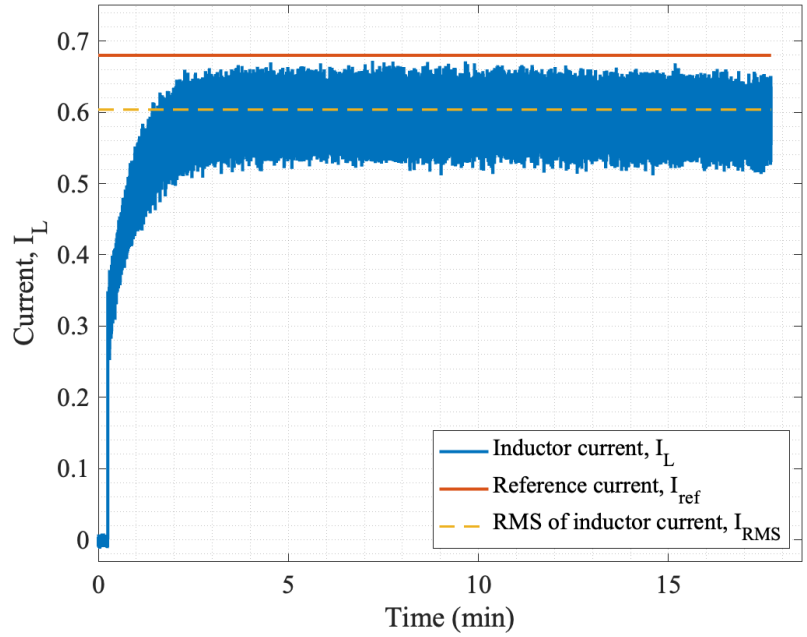
(b)

Figure 5.9: The current through the inductor settling around the reference current. The reference was 0.68 A in (a), and -0.68 A in (b).

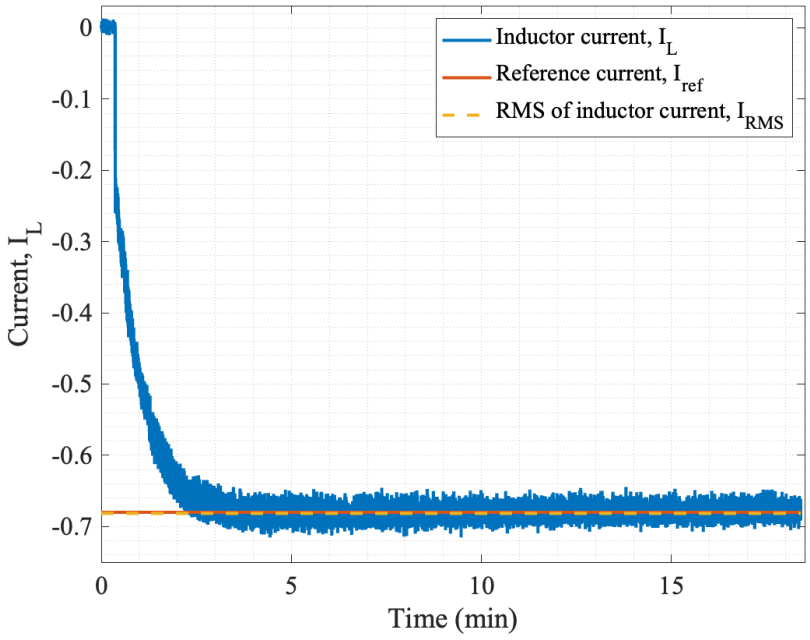
The step response shown in Figure 5.9 illustrates how the current rises towards I_{ref} for both positive and negative reference currents, showing that both directions of charge transfer could be facilitated by the cell balancing circuit and PI controller.

While the current, as shown in Figures 5.7 and 5.8, settles near the reference current, Figure 5.9 illustrates that it takes some time for the PI controller to achieve this with its chosen design, while causing slight overshoots in the process. This is further illustrated in Figure 5.10.

The results in Figures 5.10 and 5.11 were logged from the Arduino and thus are from the Arduino's voltage measurements. These tests were conducted using battery cells, as in Figure 3.3. The tests were conducted when only switching one of the MOSFETs, thus limiting efficiency while still providing the desired results. Figure 5.10 shows the inductor current.



(a)



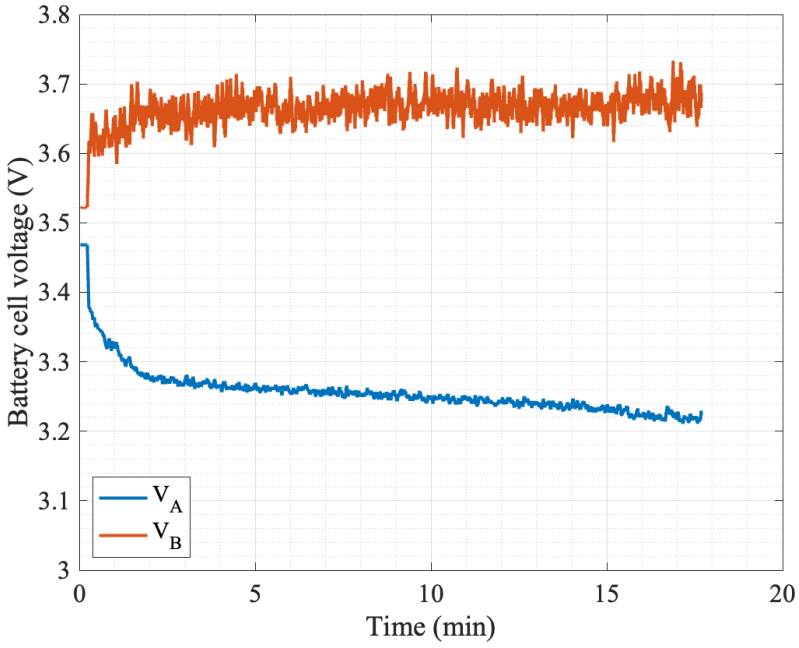
(b)

Figure 5.10: The current through the inductor. In (a) for positive I_{ref} and in (b) for negative I_{ref}.

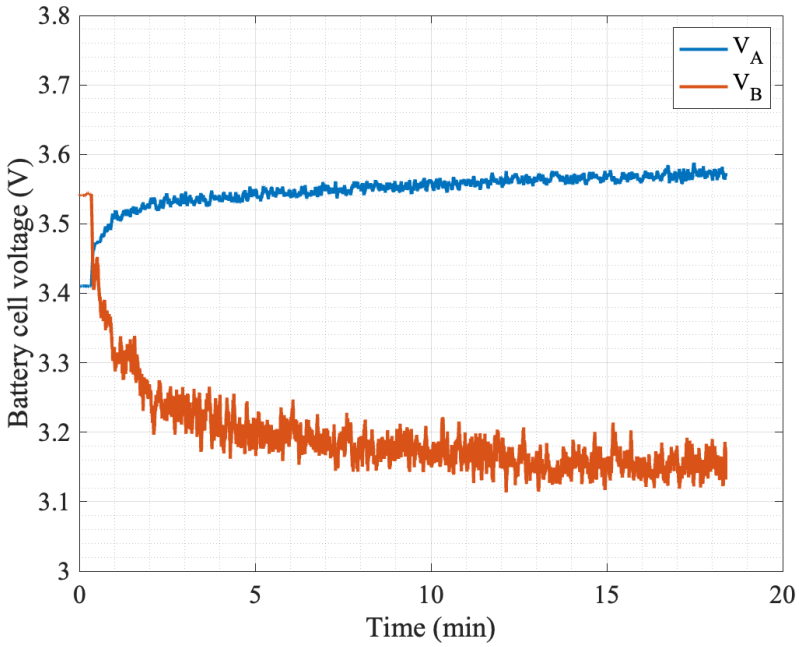
Figure 5.10a, discharging battery cell A into battery cell B, is with positive reference current, while Figure 5.10b, discharging battery cell B into battery cell A, is with negative inductor current. As for the battery cell voltage measurements from the same test, the current measurements were conducted with the Arduino. Here, the

Arduino measured the output voltage from the current sensor and translated it to a current using (3.2). The RMS current for the negative direction is close to its reference, however, the positive direction is about 0.08 A under its reference. Moreover, Figure 5.10 illustrates the response time of the PI controller, while Figure 5.9 is for such a short time that the current does not have time to settle.

Figure 5.11 illustrates battery cell voltages V_A and V_B when charge was moved between two battery cells. The first test, in Figure 5.11a, discharged battery cell A into battery cell B, while the second test, in Figure 5.11b, did the opposite.



(a)



(b)

Figure 5.11: The voltage over battery cell A and B. In (a) for positive I_{ref} and in (b) for negative I_{ref}.

Figure 5.11a shows how charge was redistributed using positive reference current, I_{ref}, and battery cell A charging battery cell B, while Figure 5.11b shows the opposite. The graphs show the average value of the voltages over 30 samples, to reduce the impact of noise induced by the Arduino's voltage measurements. The start of the

graphs are the respective OCV while the rest of them show the measured battery cell voltage during charging and discharging.

Further results from the test with battery cells include SoC estimation from the battery cells' OCV before and after charging. These are presented in Table 5.1. The "Before test 1" row corresponds to the battery cells voltage and charge level before any balancing. The "After test 1 / Before test 2" row shows the same levels after running the balancing circuit to move charge from Cell A to B. Finally "After test 2" shows the same levels after running the balancing circuit to move charge from Cell B to A.

Table 5.1: Open circuit voltages and states of charge for the battery cells.

	V_A (V)	SoC_A (%)	V_B (V)	SoC_B (%)
Before test 1	3.438	17.1	3.491	22.4
After test 1 / Before test 2	3.380	13.4	3.511	24.1
After test 2	3.368	12.7	3.455	18.6

From Table 5.1 it is clear that the cell balancing circuit can move charge with positive reference current, since SoC_A decreased during the test, while SoC_B increased. For the second test, using negative reference current, SoC_B decreased as expected while SoC_A also had a decrease. Combining the observations in Table 5.1 with the results in Figures 5.10 and 5.11, it is evident that the circuit can move charge between neighboring cells.

5.3 Battery cell balancing performance

Two key metrics of the cell balancing circuit's performance are the efficiency and the balancing speed. The tests and simulations yielding the results in this section are from using the setup in Figure 3.1. Figure 5.12 shows how the efficiency varies depending on the reference current. The figure includes both experimental, theoretical and simulated results. The theoretical calculations were conducted using (3.4) and (3.5).

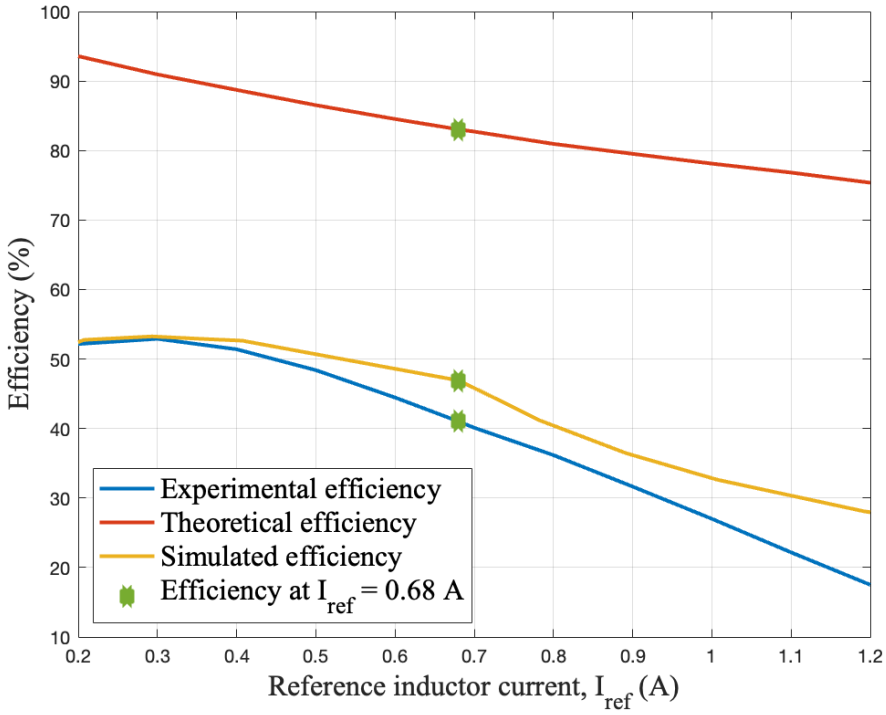


Figure 5.12: Theoretical, simulated and experimental cell balancing efficiency.

The desired balancing current of 0.68 A is marked. The efficiencies were evaluated according to Section 3.4. Theoretically, it yielded an efficiency of 83.0%, while the efficiency was 41.0% and 46.9% for the experimental and simulated counterparts, respectively. Figure 5.13 shows the simulated losses in the NMOS, PMOS and inductor for different currents. Note that the horizontal axis is component currents while I_{ref} is the inductor current, meaning that the sum of the average MOSFET currents equals the average inductor current.

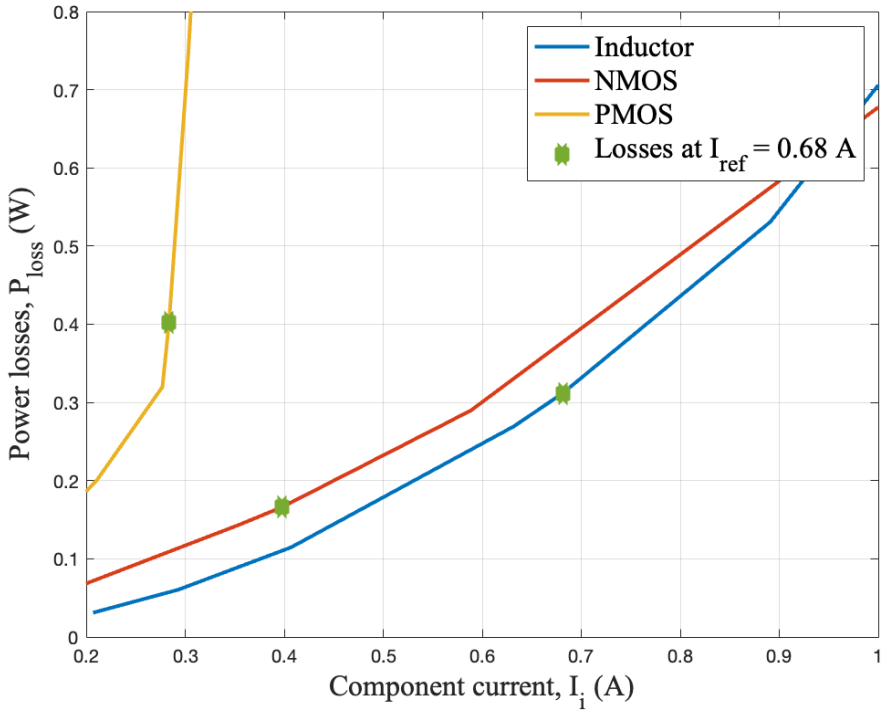


Figure 5.13: Losses for inductor, PMOS and NMOS for different component currents, based on LTspice simulations.

From Figure 5.13, it is evident that at the reference average inductor current $I_{ref} = 0.68$ A, the largest cause of losses was the PMOS at 0.4 W, while the NMOS caused the smallest losses at 0.17 W and the losses in the inductor were 0.31 W. The losses were evaluated using (3.6), resulting in -0.085 W in difference between the sum of the losses for the inductor, NMOS and PMOS, and the losses by taking the difference between power in and power out. Hence, these three components are the causes of power loss in the LTspice simulations. Using (3.7), the resistance values for the three components was estimated based on the losses in Figure 5.13 and the results are presented in Figure 5.14.

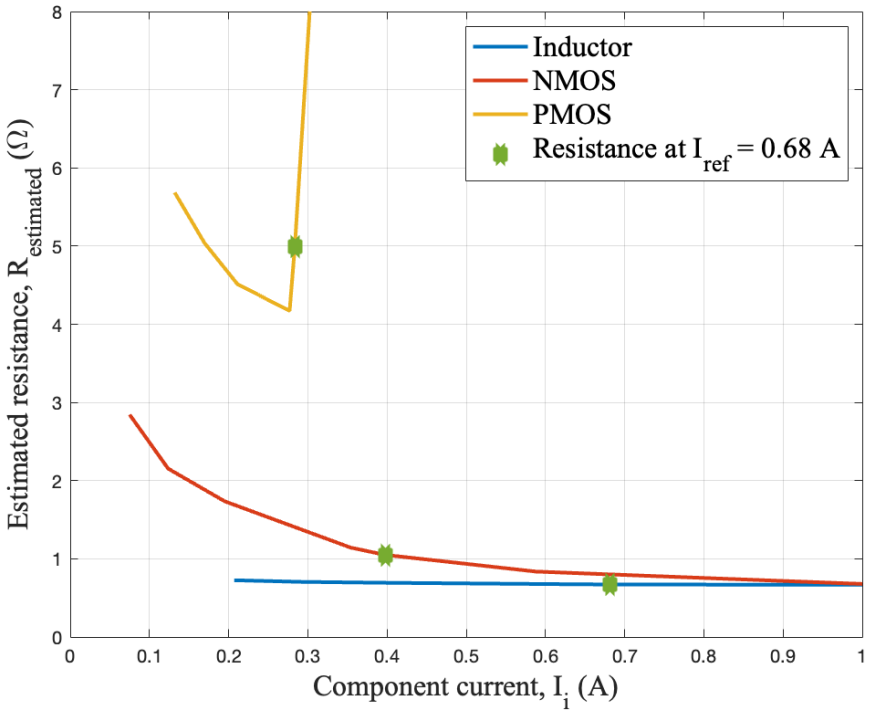


Figure 5.14: Estimated resistance values for different currents for inductor, PMOS and NMOS based on LTspice simulations.

As Figure 5.14 illustrates, the resistance for the inductor was constant while the resistances for the MOSFETs differed for different currents. The simulations were done according to Figure 3.1, using the NMOS to increase the inductor current (thus driving current in the NMOS's forward direction by turning it on) and the body diode of the PMOS to drive current through the resistor. The NMOS had a decreasing resistance for increasing currents, while the PMOS had a sharp resistance increase between 0.2 and 0.3 A. The estimated resistances at the reference inductor current $I_{ref} = 0.68$ A were 0.67 Ω for the inductor, 1.1 Ω for the NMOS, and 5.0 Ω for the PMOS.

Since balancing time decreases with increasing current while efficiency decreases with increasing current, it is interesting to study the relationship between efficiency and balancing time. This is done in Figure 5.15.

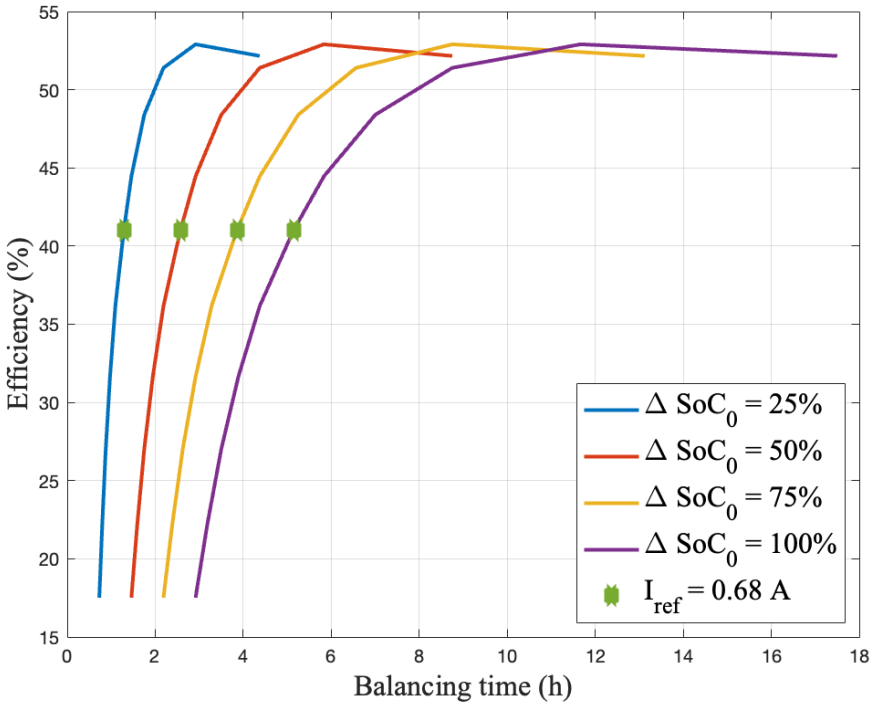


Figure 5.15: Balancing time for different efficiencies, determined for different ΔSoC_0.

The efficiency variation in Figure 5.15 was caused by varying reference current. The balancing time for two neighboring cells was calculated using (3.8). This was done using the reference currents associated with the measurement data of efficiency used in Figure 5.12. ΔSoC_0 is initial difference in SoC between the two battery cells. For each 25%, or 875 Ah, of initial imbalance in SoC between the cells, it takes 1 h 18 minutes to balance them at $I_{ref} = 0.68$ A. As expected, the efficiency is higher for longer balancing times.

6

Discussion

This chapter presents a discussion of the results, beginning with the discussion concerning the topology selection. This mostly focused on why the criteria were weighted as they were, and the implications of choosing the selected topology. This is followed by a discussion of the design choices, their implications, and possible avenues of improvement. The results of the Arduino programming and what improvements can be made are then discussed. A comparison is made between the ideal, simulated, and measured voltage and current waveforms. The cell balancing performance is then discussed further. Finally, the ethical aspects and implications of the project are discussed.

6.1 Topology selection

The Pugh matrix is adaptable due to the weight function, and the most important criteria for this project were given a weight of 2 in Table 4.1. These criteria were: efficiency, balancing time, and complexity. One of the primary motivations for developing an active balancing circuit was to address inefficiencies in passive balancing circuits, where energy is dissipated as heat during the balancing process instead of being utilized. Efficiency was therefore an important criterion, as designing an active balancing circuit with similar efficiency to a passive one would negate the benefits of the active solution.

Balancing time was another important criterion, as passive balancing speed is limited by heat dissipation. Active balancing circuits, by contrast, can redistribute the same charge with less heat generation, lowering balancing time. Achieving low balancing time is therefore of interest, since it highlights the advantages of active balancing and strengthens the case for transitioning from passive balancing.

Another important criterion was complexity, since more complex topologies often require more components. Having more components in a circuit increases the risk of hardware failures. Furthermore, complex topologies also need more complex control algorithms and more precise timing, increasing the risk of software errors. Active balancing topologies will undeniably be more complex than passive ones, but minimizing the complexity is still desirable for active balancing topologies to become a viable alternative to passive topologies.

Cost and size were weighted with 1 due to the fact that optimizing in these areas is more relevant in later development stages. This project rather focused on creating a proof of concept, which meant cost and size were not given an increased weight.

While the multiple switched inductor topology was deemed the most suitable for this project, the single switched inductor was not far behind. These two topologies performed similarly in terms of both balancing time and efficiency, two of the most important evaluation criteria. However, the single switched inductor topology is more complex, which may bring the previously mentioned problems with complex topologies. Even so, the single switched inductor could be a topology worth considering in future studies, to get a more nuanced analysis of the performance of different topologies.

Moreover, the topology selection could be made more thorough by simulating all the topologies and comparing the results. A complete evaluation of the topologies would require constructing prototypes of every topology and comparing their performance. Simulating and constructing all the active balancing topologies in Chapter 2 was outside of this project's scope. The Pugh matrix was therefore deemed a useful tool to rank the topologies since it relied on previous scientific work to examine the different topologies.

6.2 Design and construction

The tests performed on the circuit confirmed the functionality of all the parts presented in Chapter 4. However, several issues arose during the construction and testing of the prototype. These issues included out-of-stock/unavailable components, wrong and/or missing connections, interference between the current sensors and the magnetic field from the inductors, and limitations in the Arduino. All the problems with the PCB, such as component footprints, placement, and connections, were fixed physically, using wires and a breadboard. A revised design of the PCB and schematic design, and its component list can be seen in Appendix A, Figure A.2, Figure A.3, and Table A.1 respectively. However, no replacement boards were ordered since the modified original board was functional.

Two methods were considered for the voltage measurements. The first method used voltage dividers directly connected to the Arduino input ports, and the second used differential OP amps. The originally intended OP amps were unavailable, so the simpler, less accurate voltage divider method was used. The connection between the voltage divider and Arduino is illustrated by the dotted blue line in Figure 4.4. During testing, the measurement error from the voltage dividers was less than 2% (see Appendix C, Table C.1), which was deemed adequate for this application. In a real integrated active balancing circuit, using the differential amplifier would have been preferable due to the Arduino's 10-bit resolution limitation in the input ports. Since the battery cell voltages were in the range 2.5 V to 4.2 V, only 1.7 V out of the Arduino's 5 V input range was used. The calculation of cell voltages using the differential amplifiers was analog, meaning that the error caused by the Arduino's 10-bit resolution and the limited voltage range, was only induced on the difference

between two potentials rather than on the two potentials themselves. Hence, each cell voltage measurement introduced the error only once, unlike the voltage divider method, which induced it twice.

Furthermore, it was discovered that the optocouplers driving the PMOSs were incorrectly connected. One of the issues was that the wrong voltage levels were fed to the MOSFET gates, thus preventing the MOSFETs from turning on. Another issue was the orientation of the internal phototransistor in the optocouplers, preventing the PMOSs from charging and discharging with their desired current direction. These issues were corrected using wires (see Appendix B, Figure B.1), after which the PWM controlled MOSFETs worked as originally intended.

Another problem was that the current sensor measurements were distorted by the magnetic field induced by the inductors. The PCB was designed such that the inductors were placed right above the sensors, and since the current sensors utilize the Hall effect for their measurements, the sensor values were inaccurate. This was solved by removing the inductors from the board and instead connecting them using a breadboard (see Appendix B, Figure B.2), consequently making the current sensor give correct values. However, by placing the inductors away from the MOSFETs and capacitors, additional inductance might have been induced by the wires, thus affecting the size of the inductor current ripple.

Furthermore, the mechanical switches (see Appendix A, components S1-S4 in Figure A.1) had the wrong pin connection, so consequently, the built-in passive balancing did not work. This was deemed unnecessary to fix for the PCB used during testing, since the SoC levels could be altered externally beforehand. New mechanical switches were instead added to the new PCB design. However, these switches would not be necessary in a real application since they only serve testing purposes.

Future improvements of the circuit could involve removing the Arduino and replacing it with an integrated chip. This would make the PCB smaller since it would no longer need to allocate space for the Arduino. The Arduino has many functionalities not utilized in the balancing circuit, meaning an integrated chip only containing the necessary functions would reduce the complexity of the circuit. Furthermore, selecting better fitting components could increase efficiency and lower balancing time, however, this is further discussed in Section 6.5.

6.3 Control system implementation

To perform balancing, the algorithm presented as a flowchart in Figure 4.6 would be used. The algorithm accounts for balancing and safety features not encompassed by the PI controller. However, for this project, all tests were conducted under supervision, which meant that safety features that would allow the circuit to work unsupervised were not tested. These safety features were under/over voltage and temperature controls. Furthermore, the PI controller also served as a current limiter, allowing for safe testing. The algorithm in Figure 4.6 was implemented into the Arduino before the hardware was fully finished, which limited the ability to test the

full algorithm. The functionality of the algorithm was instead tested using other test setups that did not fully mimic the operation of the finished hardware. When the algorithm was tested on the finished hardware, it did not work as intended. At this point, it was decided to simplify the algorithm to only focus on moving charge between cells. The simpler solution still showed that balancing was possible, and with further programming, the algorithm could be implemented to balance battery cells.

Moreover, the presented balancing algorithm contained issues pertaining to its method of gathering current data. The algorithm in Figure 4.6 gathered all data except for the current at the start of the switching period. Instead, the current was measured using a mean value, meaning several measurements were taken during the switching period, which made the program slower. Two solutions to the current measurements were found: rolling mean and measuring mean. A rolling mean is similar to what was done before, but instead of gathering all the data at once, only one data point was gathered. This data point could then be added to a list with previous values, and the oldest data point would be discarded. Doing so would not slow down the program as much. A measuring mean is a carefully selected point in time when the current is at its mean value. This method would work since the current's shape is known. However, due to the noisy measurements from the Hall effect current sensors as seen in Figure 5.8 compared to the current probe in Figure 5.7, the rolling mean method was used.

6.4 Comparison of prototype and simulations

Comparing Figure 5.1a to Figure 5.5 made it clear that the switching in the physical prototype worked as intended. This is substantiated by the fact that the MOSFETs switched on when the PWM signals were high and switched off when the PWM signals were low. The physical measurement in Figure 5.5 was slightly noisier than its Simulink counterpart in Figure 5.1a. However, this was not relevant to the circuit's functionality, as the noise amplitude was many times smaller than the amplitude of the PWM signal.

Furthermore, comparing the ideal inductor voltage shown in Figure 2.7 to the simulated and measured ones in Figure 5.2 and 5.6, respectively, shows that the inductor voltage behaved mostly as expected. The voltage across the inductor toggled between positive and negative for both the simulated and physical measurements, just like in Figure 2.7. In the ideal case, the average inductor voltage should be zero assuming $|V_A \cdot D_1 T| = |V_B \cdot D_2 T|$, which held true for the simulated voltage. This was, however, not the case for the measured inductor voltage in Figure 5.6, which had an average voltage of about 0.3 V. The cause of this deviation was the resistive losses in the circuit (see Figure 5.14), causing voltage drops. Thus, the PI controller compensates by keeping the average inductor voltage at a level where the average inductor current does not decrease, see (2.8).

Studying the inductor current in Figure 5.7 further shows that it possessed similar, but not identical, characteristics as the currents shown in the ideal case and the

Simulink simulation, Figure 2.8 and 5.3 respectively. As can be seen in Figure 5.3, the simulated inductor current rises to its reference and ripples evenly around it. However, this was not the case for the measured current in Figures 5.7 and 5.8, which illustrate the current measured with a current probe and with the PCB-mounted current sensor, respectively. Both of these currents settled below the reference. One cause for this was that the voltage that the current sensor outputs was not read accurately by the Arduino. For Figure 5.7, the current probe might have given an additional offset, since it was observed not to indicate zero current when the current was zero.

Both Figure 5.7 and 5.8 show that the PI controller implemented in Simulink could control the current to a desired level. Figure 5.9a and 5.9b further illustrate the step response of the current controller for about 4 ms. Due to the chosen low bandwidth $\alpha_c = 294$ Hz of the control system, it took several minutes for the current to reach the reference, and this is depicted in Figure 5.10. For Figure 5.9, however, it appears that the current reached the reference in about 0.5 ms. However, this was a current spike, and it is discernible how the current started decreasing from the spike. The current spike, however, was similarly sized as the reference current and therefore deemed acceptable. The current then slowly increased towards the reference, and this is the slow rise seen in Figure 5.10. The current spike in Figure 5.9 is not present in Figure 5.10, and the likely cause of this is that the Arduino prints the average current over 100 samples every 90th ms and thus misses the current spike. There was a significant difference between the rise times as simulated in Simulink and as measured on the prototype. The slow rise time was caused by the low bandwidth, giving a low proportional gain K_p, chosen to avoid oscillatory behavior and unacceptable current spikes. Accordingly, the integration was responsible for slowly making the current reach the reference.

The testing conducted with battery cells, the results of which are presented in Figure 5.11 and Table 5.1, confirms that the cell balancing circuit can discharge one battery cell into another. Furthermore, Figure 5.10 shows that this can be done in a safe manner with sufficient control over the inductor current, facilitated by the PI controller. While no balancing algorithm has been developed, the result suffices to show that balancing could have been done, had the algorithm been developed.

For Figure 5.10b, the current rises from zero to its reference as intended. However, for Figure 5.10a, the current starts from zero but does not rise to the desired reference. It instead settles at around 0.60 A. This inconsistency might be due to several factors, but since the current still manages to settle at a value, the control system still seems to be working. Since the reference current is chosen manually in the code, the input might be wrong, causing it to be 0.60 A instead of the 0.68 A as intended.

The voltages in Figure 5.11 cannot be directly translated into SoC, since they are voltages read during charging and discharging of the battery cells. However, the slope of the lines indicates that charge is being redistributed, and this is substantiated by voltage measurements conducted before and after the charging/discharging process, having let the battery cells settle. These results are in Table 5.1. From the

first test, the SoC of one cell decreased, while it increased for the other, according to expectations. For the second test, the correct SoC decreased while the other also had a slight decrease. For this test, only the PMOS was switched, resulting in the negative inductor current. The PMOS's on resistance $R_{DS(on)}$ is high and impossible to read exactly from its data sheet for V_{gs} values in the range 2 V to 4 V, where the test was conducted [45]. Thus, losses are the likely cause of the SoC for both cells decreasing. Despite the inefficiency, the results suffice to determine that the circuit functions according to expectations and can balance battery cells bidirectionally. The efficiency would have increased slightly had both MOSFETs been switched, however this would not have prevented SoC_A in Table 5.1 since the PMOS was switched in this case.

6.5 Battery cell balancing performance

Two key metrics for evaluating the circuit are the balancing time and the balancing efficiency. The balancing time is a result of the chosen reference current used in the PI controller, and is hence limited by component choices and the maximum ratings for the battery cells. The efficiency, on the other hand, was determined in three ways: theoretically, experimentally, and by simulations, and the results are presented in Figure 5.12.

The first observation to make from Figure 5.12 is that the theoretical efficiency, calculated by estimating the losses in the switches and the inductor as described in Section 3.4, with good margin exceeds the efficiency as determined experimentally and by simulations. The second observation to make is that the experimental and simulated efficiencies were similar, while the experimental efficiency had a greater rate of decrease for larger currents than its simulated counterpart. A probable explanation for this is that the resistive losses in the inductor cause heat which further increases the resistance and hence the losses in the inductor. This effect is more evident the larger the current, both since large currents generate more heat and since the testing was conducted starting from the smallest current level and increasing it. Thus, the circuit had been running for longer when measuring the larger currents, having generated more heat. This explanation is strengthened since the efficiency was observed to increase when the inductor was replaced with a new and colder one.

The similarities between the experimental and simulated efficiencies provide an argument for using the LTspice simulation of the circuit in Figure 3.1 to further analyze the cause of losses. Resistance values for the components were estimated using their losses, and are presented in Figure 5.14. The inductor resistance was constant for all currents and in line with its data sheet [43]. The NMOS resistance values did not match its counterparts in the data sheet (I_D vs. $R_{DS}(on)$ characteristics), since the resistance should increase for increasing current and also is notably higher for the estimated values [44]. For the PMOS, the shape of the resistance graph resembles the graph in the data sheet, however the values are more than an order of magnitude larger for the estimated values than in the data sheet (-I_D vs. $R_{DS}(on)$ characteris-

tics) [45].

The resistance estimations for the NMOS and PMOS in Figure 5.14 appear implausible since they are much higher than what the data sheets suggest. Since they are estimated from the power losses, a probable explanation is that power losses in the MOSFETs are not only caused by the on resistance. Further losses were likely caused by simultaneous V_{ds} voltage and drain current I_{ds} with the instantaneous size $V_{ds} \cdot I_{ds}$. This can occur when the MOSFETs are turning on or off, meaning that V_{ds} is increasing or decreasing, but the MOSFET still conducts a small current I_{ds}. The theoretical efficiency, using the relevant characteristics from the data sheets, is double the measured efficiency and as per the discussion above, the only cause of this appears to be the MOSFETs.

Since there, as Figure 5.15 clearly illustrates, is a compromise between efficiency and balancing time and both properties are desirable, more careful component choices would have been needed had the circuit been revised. For the inductor, one with fewer windings could have been chosen. This, however, would have decreased inductance and thus increased the current ripple. An increased current ripple might require a decreased average current to limit the current peaks, which would decrease the balancing speed. Another solution would be to increase the switching frequency and thus limit the MOSFET on time. However, the switching frequency here was chosen since it allowed for the MOSFET gates to discharge properly each switching period. For the MOSFETs, component choice would have to be done with switching frequency in mind, making sure they turn on and off quickly to limit switching losses and, if so needed, account for a higher switching frequency. Furthermore, MOSFET choice needs to take V_{gs} values into account since (as the tests with battery cells illustrate) the PMOS's on resistance is too high to enable efficient balancing.

The results and discussion above highlight the challenges of designing low power electronic converters: a substantial portion of the converted power can easily be lost. The design must be approached with a broad perspective, considering factors such as switching frequency, inductance, switches, and the circuitry driving the switches, in order to achieve the desired balancing time and efficiency. Nevertheless, the prototype and theoretical analysis demonstrate the viability of this type of battery cell balancing circuit.

6.6 Ethical discussion

Throughout the project, the topic of safety was prioritized during construction and testing, serving as a key ethical foundation for the project. The control system served to help determine and control that the circuit operated within the predetermined safety parameters. Furthermore, the testing procedure was also executed in a manner that minimized risks to personnel and equipment. These safety measures included: Using power supplies to limit the current in the circuit and using the test rig in Figure 3.2 before the second test rig in Figure 3.3. Using the capacitor test rig meant that the circuit's functionality could be tested without risking overloading the more volatile lithium-ion batteries. As a final fail-safe, the use of fuses meant

that the two test rigs using capacitors and batteries would be limited to currents that the circuit was capable of handling. Access to the lab used for constructing and testing the prototype required passing a safety course and having a guided tour. These measures ensured that all group members possessed the knowledge necessary to safely work in the lab.

With environmental effects in mind, the added complexity of an active balancing circuit compared to the passive alternative will likely include more semiconductor components. The negative environmental impacts of these semiconductor components could be used as an argument not to transition to active balancing. However, the accelerated degradation of the batteries that arises with passive balancing will increase the need for replacing the cells, further fueling lithium mining. Therefore, research into active balancing circuits that use few semiconductor components while also achieving a low DoD could help reduce environmental pressures associated with both lithium mining and semiconductor manufacturing.

7

Conclusion

The Pugh matrix analysis, comparing multiple topologies, showed that the multiple switched inductor topology performed best overall when evaluating balancing time, efficiency, complexity, size, and cost. The multiple switched inductor prototype designed, constructed, and evaluated in this project was capable of balancing battery cells. The circuit and its PI controller facilitated control over the balancing current in accordance with the set reference current, without external intervention. With the chosen balancing current of $I_{ref} = 0.68$ A, the balancing efficiency was concluded to be 41% and this was verified by simulations. The theoretical efficiency, however, was concluded to be roughly 83%. At this current, it would take 1 h 18 minutes to balance 875 mAh (25% of the total battery cell capacity) of imbalance between two battery cells. Since efficiency decreased with increasing current, while increasing current increased balancing speed, there is a compromise between efficiency and balancing speed. Both high balancing speed and high efficiency are desirable. Accordingly, careful component selection is vital, and this point is substantiated by the high theoretical efficiency.

Bibliography

[1] S. M. Shafi Ul Alam and M. H. Rahman, "Use of green energy instead of ips to lessen energy crisis in bangladesh", in *2nd International Conference on Green Energy and Technology*, 2014, pp. 83–86. DOI: 10.1109/ICGET.2014.6966668.

[2] C. Guu-Chetruca and D. Braga, "Energy crises energy transition driving force", in *2023 International Conference on Electromechanical and Energy Systems (SIELMEN)*, 2023, pp. 1–6. DOI: 10.1109/SIELMEN59038.2023.10290742.

[3] Z. Shi, W. Wang, Y. Huang, P. Li, and L. Dong, "Simultaneous optimization of renewable energy and energy storage capacity with the hierarchical control", *CSEE Journal of Power and Energy Systems*, vol. 8, no. 1, pp. 95–104, 2022. DOI: 10.17775/CSEEJPES.2019.01470.

[4] IEA. "Electrification". (Jan. 2025), [Online]. Available: https://www.iea.org/energy-system/electricity/electrification (visited on 02/01/2025).

[5] I. C. T. Bowen and P. Denholm. "Grid-scale battery storage: Frequently asked questions". (Sep. 2010), [Online]. Available: https://www.nrel.gov/docs/fy19osti/74426.pdf (visited on 04/28/2025).

[6] L. Zheng, J. Zhu, and G. Wang, "A comparative study of battery balancing strategies for different battery operation processes", in *2016 IEEE Transportation Electrification Conference and Expo (ITEC)*, 2016, pp. 1–5. DOI: 10.1109/ITEC.2016.7520204.

[7] D. Y. Reddy, B. Routh, A. Patra, and S. Mukhopadhyay, "Gaussian process regression based state of health estimation of lithium-ion batteries using indirect battery health indicators", in *2021 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2021, pp. 1–7. DOI: 10.1109/ICPHM51084.2021.9486519.

[8] P. Ramesh, J. Sahoo, A. Patra, and D. Kastha, "A voltage-based charge balancing algorithm to mitigate the charge recovery effect of li-ion batteries", in *2024 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, 2024, pp. 1–6. DOI: 10.1109/PEDES61459.2024.10960994.

[9] M. Hassini, E. Redondo-Iglesias, and P. Venet, "Lithiumion battery data: From production to prediction", *Batteries*, vol. 9, no. 7, 2023, ISSN: 2313-0105. DOI: 10.3390/batteries9070385. [Online]. Available: https://www.mdpi.com/2313-0105/9/7/385.

[10] Amin, K. Ismail, A. Nugroho, and S. Kaleg, "Passive balancing battery management system using mosfet internal resistance as balancing resistor", in *2017 International Conference on Sustainable Energy Engineering and Application (ICSEEA)*, Jakarta, Indonesia, 2017, pp. 151–155. DOI: 10.1109/ICSEEA.2017.8267701. (visited on 01/25/2025).

[11] B. Jiang, "Active cell balancing algorithms in lithium-ion battery", Chalmers Department of Electrical Engineering, 2020.

[12] M. Räber, D. H. A. Heinzelmann, and D. O. Abdeslam, "A novel non-isolated active charge balancing architecture for lithium-ion batteries", in *2018 IEEE 27th International Symposium on Industrial Electronics (ISIE)*, Cairns, QLD, Australia, 2018, pp. 31–36. DOI: 10.1109/ISIE.2018.8433669. (visited on 01/25/2025).

[13] N. Khan *et al.*, "A critical review of battery cell balancing techniques, optimal design, converter topologies, and performance evaluation for optimizing storage system in electric vehicles", *Energy Reports*, vol. 11, pp. 4999–5032, Jun. 2024. DOI: 10.1016/j.egyr.2024.04.041.

[14] *Product specification rechargeable lithium ion battery*, INR18650 MJ1, Rev. 1, LG Chem, Aug. 2014.

[15] I. Baccouche and N. E. Ben Amara, "A comprehensive overview of ai based methods for soc estimation of li-ion batteries in ev", in *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2024, pp. 1885–1890. DOI: 10.1109/CoDIT62066.2024.10708464.

[16] M. Daowd, N. Omar, P. Van Den Bossche, and J. Van Mierlo, "Passive and active battery balancing comparison based on matlab simulation", in *2011 IEEE Vehicle Power and Propulsion Conference*, 2011, pp. 1–7. DOI: 10.1109/VPPC.2011.6043010. (visited on 01/26/2025).

[17] W. Zhu, X. Zhou, M. Cao, Y. Wang, and T. Zhang, "The cloud-end collaboration battery management system with accurate state-of-charge estimation for large-scale lithium-ion battery system", in *2022 8th International Conference on Big Data and Information Analytics (BigDIA)*, 2022, pp. 199–204. DOI: 10.1109/BigDIA56350.2022.9874144.

[18] R. Chakraborty, B. Srinivas, and A. Chakraborty, "Impact of lithium mining on climate change in the atacama desert, south america", in *202what 2 3rd URSI Atlantic and Asia Pacific Radio Science Meeting (AT-AP-RASC)*, Gran Canaria, Spain, 2022, pp. 1–4. DOI: 10.23919/AT-AP-RASC54737.2022.9814388. (visited on 02/02/2025).

[19] S. Ahmad. "The lithium triangle: Where chile, argentina, and bolivia meet", Harvard International Review. (Jan. 15, 2020), [Online]. Available: https://hir.harvard.edu/lithium-triangle/ (visited on 02/03/2025).

[20] G. Yüksek and A. Alkaya, "Effect of the depth of discharge and c-rate on battery degradation and cycle life", in *2023 14th International Conference on Electrical and Electronics Engineering (ELECO)*, 2023, pp. 1–5. DOI: 10.1109/ELECO60389.2023.10415967. (visited on 02/05/2025).

[21] K. Frost and I. Hua, "A spatially explicit assessment of water use by the global semiconductor industry", in *2017 IEEE Conference on Technologies for Sustainability (SusTech)*, Phoenix, AZ, USA, 2017, pp. 1–5. DOI: 10.1109/SusTech.2017.8333525. (visited on 02/02/2025).

[22] G. Plett, *Battery Management Systems, Volume II: Equivalent-Circuit Methods*. Artech House, 2015, pp. 237–263.

[23] M. A. F. Bossche Alex Van den, *Battery Management Systems and Inductive Balancing*. Institution of Engineering and Technology (The IET), 2021, ISBN: 978-1-83953-357-0. [Online]. Available: https://app.knovel.com/hotlink/toc/id:kpBMSIB003/battery-management-systems/battery-management-systems (visited on 02/24/2025).

[24] N. Kutkut and D. Divan, "Dynamic equalization techniques for series battery stacks", in *Proceedings of Intelec'96 - International Telecommunications Energy Conference*, 1996, pp. 514–521. DOI: 10.1109/INTLEC.1996.573384.

[25] C.-H. Kim, H.-S. Park, C.-E. Kim, G.-W. Moon, and J.-H. Lee, "Individual charge equalization converter with parallel primary winding of transformer for series connected lithium-ion battery strings in an hev", *Journal of power electronics*, vol. 9, May 2009. [Online]. Available: https://www.researchgate.net/publication/237110282_Individual_Charge_Equalization_Converter_with_Parallel_Primary_Winding_of_Transformer_for_Series_Connected_Lithium-Ion_Battery_Strings_in_an_HEV (visited on 05/13/2025).

[26] J. Yan, Z. Cheng, G. Xu, H. Qian, and Y. Xu, "Fuzzy control for battery equalization based on state of charge", in *2010 IEEE 72nd Vehicular Technology Conference - Fall*, 2010, pp. 1–7. DOI: 10.1109/VETECF.2010.5594380.

[27] R. Petrocelli, "One-quadrant switched-mode power converters", *CERN*, pp. 131–132, May 2015. DOI: https://doi.org/10.5170%2FCERN-2015-003.

[28] M. Tyrpekl, M. Zavrel, and V. Kindl, "Comparison of active and passive battery balancing", in *2022 20th International Conference on Mechatronics - Mechatronika (ME)*, 2022, pp. 1–6. DOI: 10.1109/ME54704.2022.9983462.

[29] C. K. Alexander and M. N. Sadiku, *Fundamentals of Electric Circuits*. McGraw Hill, 2021, ISBN: 978-0-07-338057-5.

[30] Y. Ye, J. Jiang, E. Zhao, P. Li, Z. Li, and X. Hui, "An improved balancing strategy for inductor-based balancing circuit", in *2023 IEEE International Conference on Power Science and Technology (ICPST)*, 2023, pp. 387–391. DOI: 10.1109/ICPST56889.2023.10165622.

[31] A. Fernandez and D. Dang, "Chapter 9 - 1s and 0s revisited: The digital stream", in *Getting Started with the MSP430 Launchpad*, A. Fernandez and D. Dang, Eds., Oxford: Newnes, 2013, pp. 127–142, ISBN: 978-0-12-411588-0. DOI: https://doi.org/10.1016/B978-0-12-411588-0.00009-1.

[32] T. Wilmshurst, "Chapter 9 - taking timing further", in *Designing Embedded Systems with PIC Microcontrollers (Second Edition)*, T. Wilmshurst, Ed., Second Edition, Boston: Newnes, 2010, pp. 257–294, ISBN: 978-1-85617-750-4. DOI: https://doi.org/10.1016/B978-1-85617-750-4.10012-5.

[33] Fluke. "What is duty cycle?" (), [Online]. Available: https://www.fluke.com/en/learn/blog/electrical/what-is-duty-cycle (visited on 05/08/2025).

[34] L. Harnefos, *Control of Variable-Speed Drives*. Mälardalen University, 2002.

[35] M. LibreTexts. "6.2: Transforms of derivatives and odes". (), [Online]. Available: https://math.libretexts.org/Bookshelves/Differential_Equations/Differential_Equations_for_Engineers_(Lebl)/6%3A_The_Laplace_Transform/6.2%3A_Transforms_of_derivatives_and_ODEs (visited on 05/10/2025).

[36] A. Aurousseau, V. Vuillerme, and J.-J. Bezian, "Control systems for direct steam generation in linear concentrating solar power plants a review", *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 611–630, 2016, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2015.11.083. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032115013507.

[37] R. Vongvit, "The triz integrated pugh model approach for design problem solving", in *2019 5th International Conference on Information Management (ICIM)*, 2019, pp. 308–311. DOI: 10.1109/INFOMAN.2019.8714697.

[38] S. S. Sreejith and M. Mathirajan, "A modified pugh matrix method for continuous performance evaluation of employees", *South Asian Journal of Human Resources Management*, pp. 149–163, Mar. 2022. DOI: https://doi.org/10.1177/23220937221081946.

[39] Australian Department of Fire and Emergency. "Lithium-ion batteries". (2025), [Online]. Available: https://www.dfes.wa.gov.au/hazard-information/fire-in-the-home/lithium-ion-batteries (visited on 02/09/2025).

[40] Infineon. "Irlml9301". (May 2010), [Online]. Available: https://www.infineon.com/cms/en/product/power/mosfet/p-channel/irlml9301/ (visited on 03/29/2025).

[41] Infineon. "Irlml6346". (Oct. 2010), [Online]. Available: https://www.infineon.com/cms/en/product/power/mosfet/n-channel/irlml6346/ (visited on 03/29/2025).

[42] Allegro, *Acs723 high accuracy, galvanically isolated current sensor ic with small footprint soic8 package*, Document No. 1.508.853.5000, Worcester, Massachusetts 01615-0036 U.S.A., 2014.

[43] *Specification for release*, 7447709681, Rev. 2, Würth Elektronik, Aug. 2011. [Online]. Available: https://docs.rs-online.com/c09c/0900766b80febd2b.pdf (visited on 05/08/2025).

[44] *Irlml6346trpbf*, International Rectifier, Sep. 2012. [Online]. Available: https://media.distrelec.com/Web/Downloads/_t/ds/IRLML6346TRPBF_eng_tds.pdf (visited on 05/08/2025).

[45] *Irlml9301trpbf*, International Rectifier, Sep. 2012. [Online]. Available: https://media.distrelec.com/Web/Downloads/_t/ds/IRLML9301TRPBF_eng_tds.pdf (visited on 05/08/2025).

[46] *Aluminum electrolytic capacitors*, Nichicon, n.d. [Online]. Available: `https://docs.rs-online.com/8f49/0900766b80ebdadd.pdf` (visited on 05/11/2025).

[47] *Multilayer ceramic chip capacitors*, TDK, 2021. [Online]. Available: `https://docs.rs-online.com/298a/A700000008889547.pdf` (visited on 05/11/2025).

[48] *Optocoupler, phototransistor output, with base connection*, Vishay, n.d. [Online]. Available: `https://docs.rs-online.com/57c7/A700000009004816.pdf` (visited on 05/11/2025).

[49] *Arduino mega2560*, Arduino, n.d. [Online]. Available: `https://media.distrelec.com/Web/Downloads/_t/ds/mega2560_eng_tds.pdf` (visited on 05/12/2025).

Appendix A

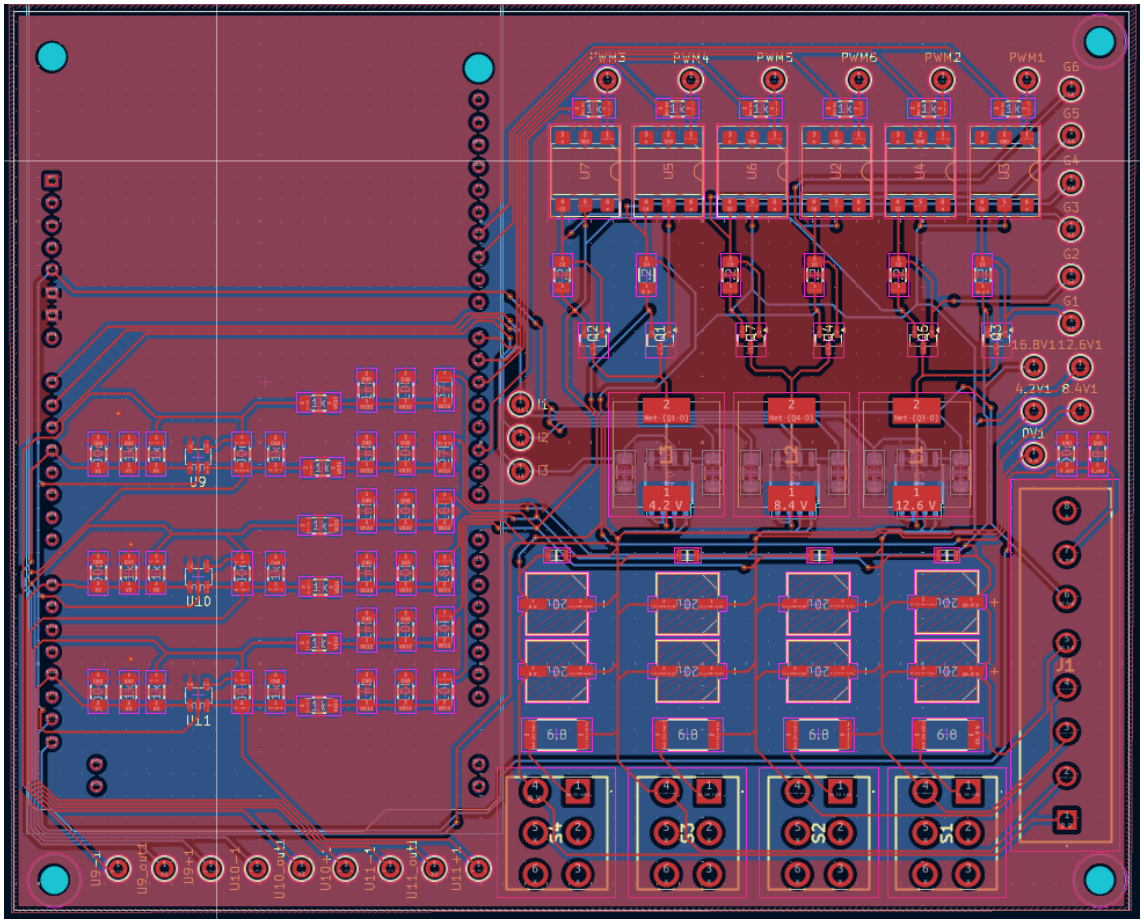


Figure A.1: KiCad PCB design used during prototype measurements and evaluation.

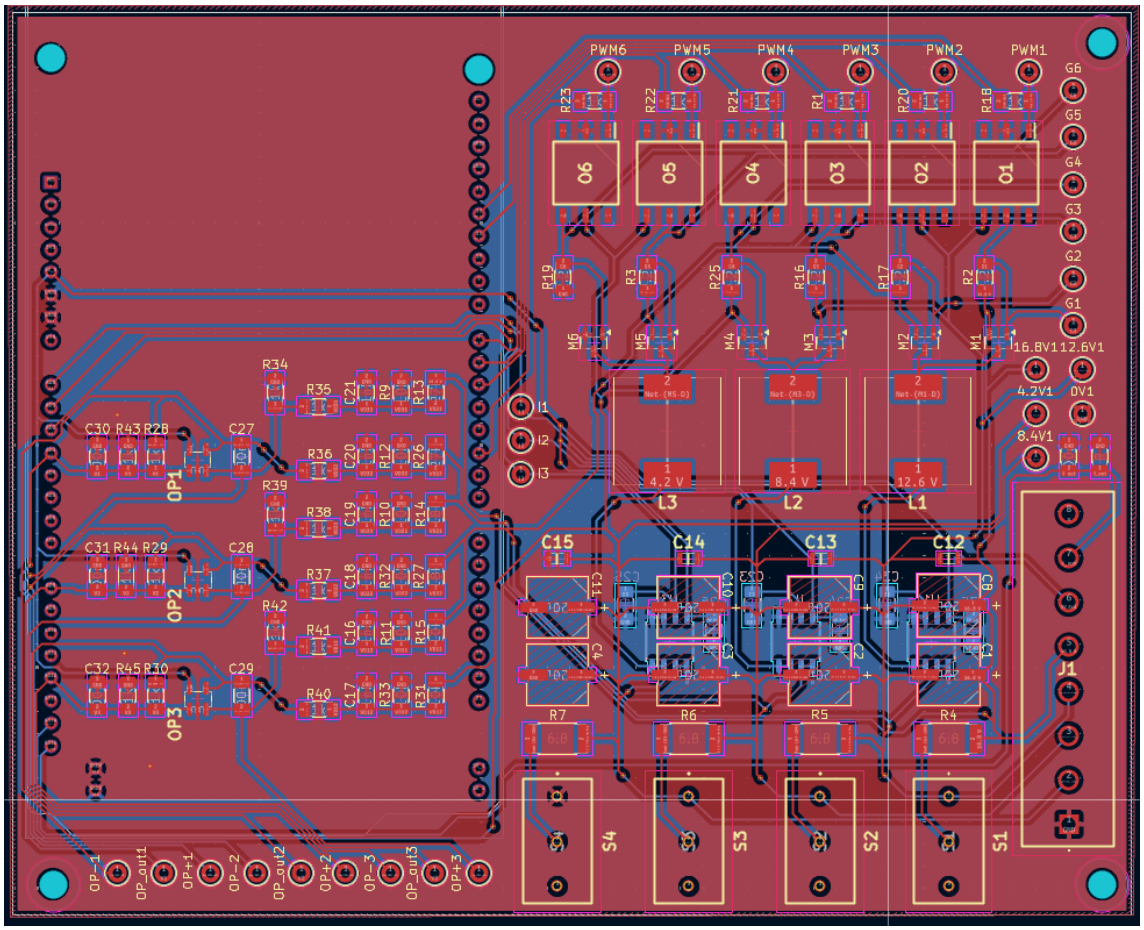


Figure A.2: Improved KiCad PCB design adapted from the errors from the first design.

Table A.1: Components to PCB

Articale number	Type	Quantity
A000067	Microcontroller	1
RS PRO 144-4314	Terminal Block	1
AD8031ARTZ-REEL7	OP amp	3
IRLML6346TRPBF	NMOS	3
IRLML9301TRPBF	PMOS	3
NTCLG100E2103JB	Thermistor	1
CRCW12062K20FKEA	Resistor 2.2 k	6
CRCW120610K0FKEA	Resistor 10 k	8
CRCW12061K00FKEA	Resistor 1 k	22
CRCW120627K0FKEA	Resistor 27 k	2
352212RJT	Resistor 12 k	4
CRCW120618K0FKEA	Resistor 18 k	2
7447709681	Inductor 680 uH	3
4N25-X009	Opto coupler	6
MSS3112D	Mechanical switch	4
ACS723LLCTR-05AB-T	Current sensor	3
KEY-5002	Test Loop	28
885342208021	Capacitor 10nF	9
CGA4J1X7T0J226M125AC	Capacitor 22uF	4
UCD0J221MCL1GS	Capacitor 220uF	8
885342208004	Capacitor 100nF	6

Table A.2: Components to capacitor and battery rig

Articale number	Type	Quantity
0251010.MAT1L	Fuse 10 A	1
Keystone 1042	Battery holder	4
37012000000	Fuse 2 A	10
RE521-LF	Stripboard	1
B41231A4689M060	Capacitor 6800uF	4
RND 210-00582	Slide switch through hole	5

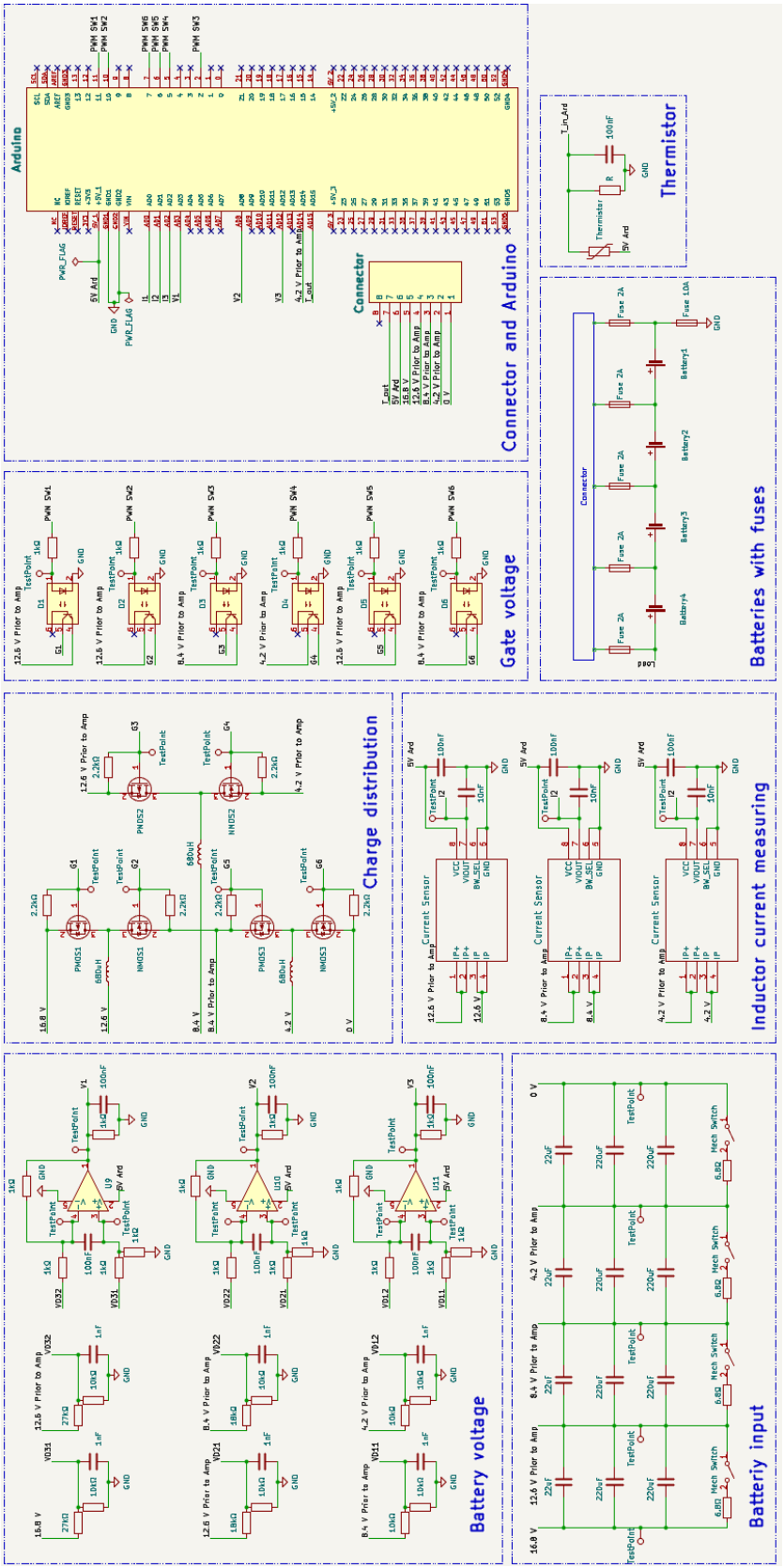


Figure A.3: Full overview of the final KiCad schematic design.

Appendix B

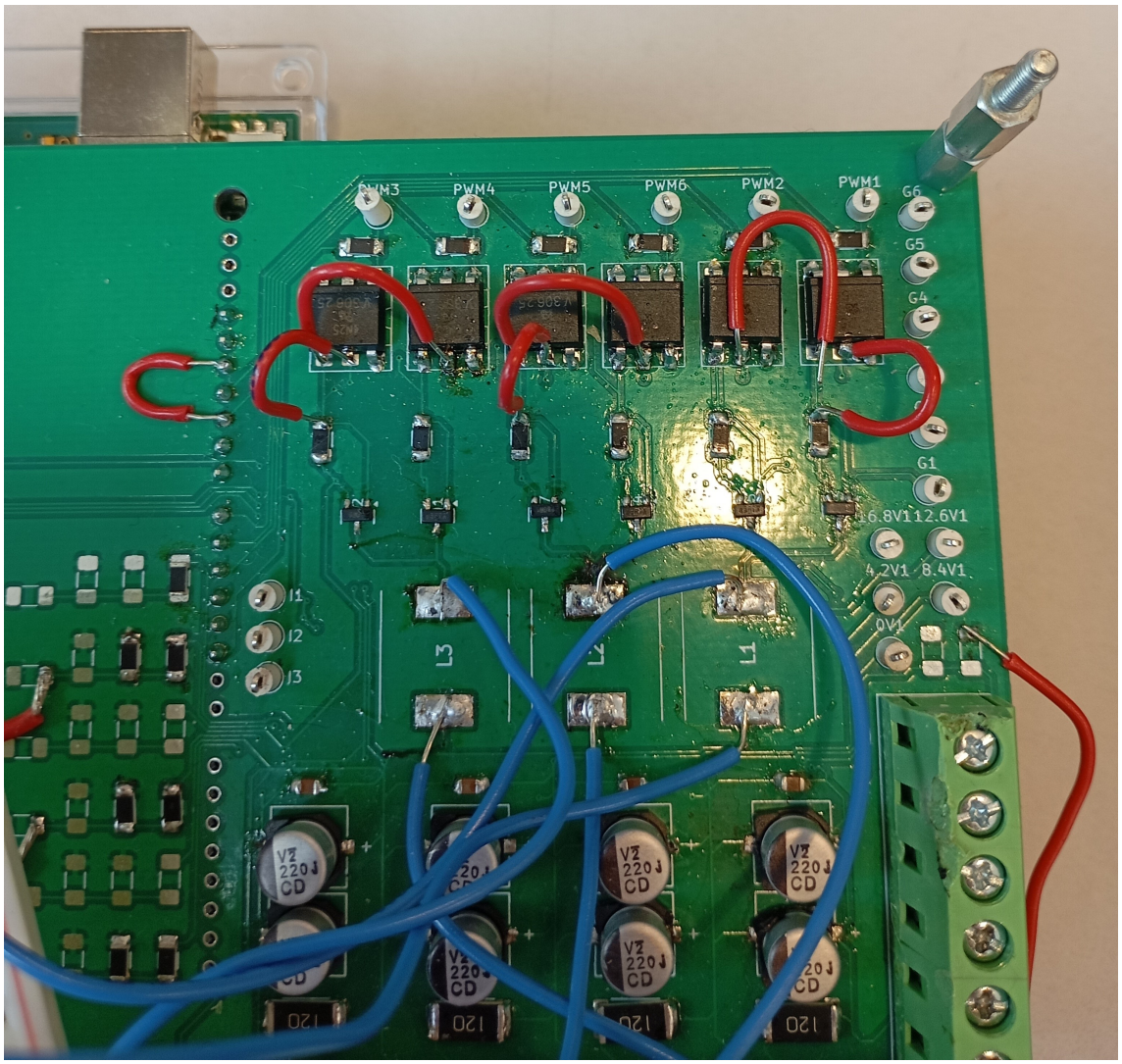


Figure B.1: The optocoupler connections fixed with wires

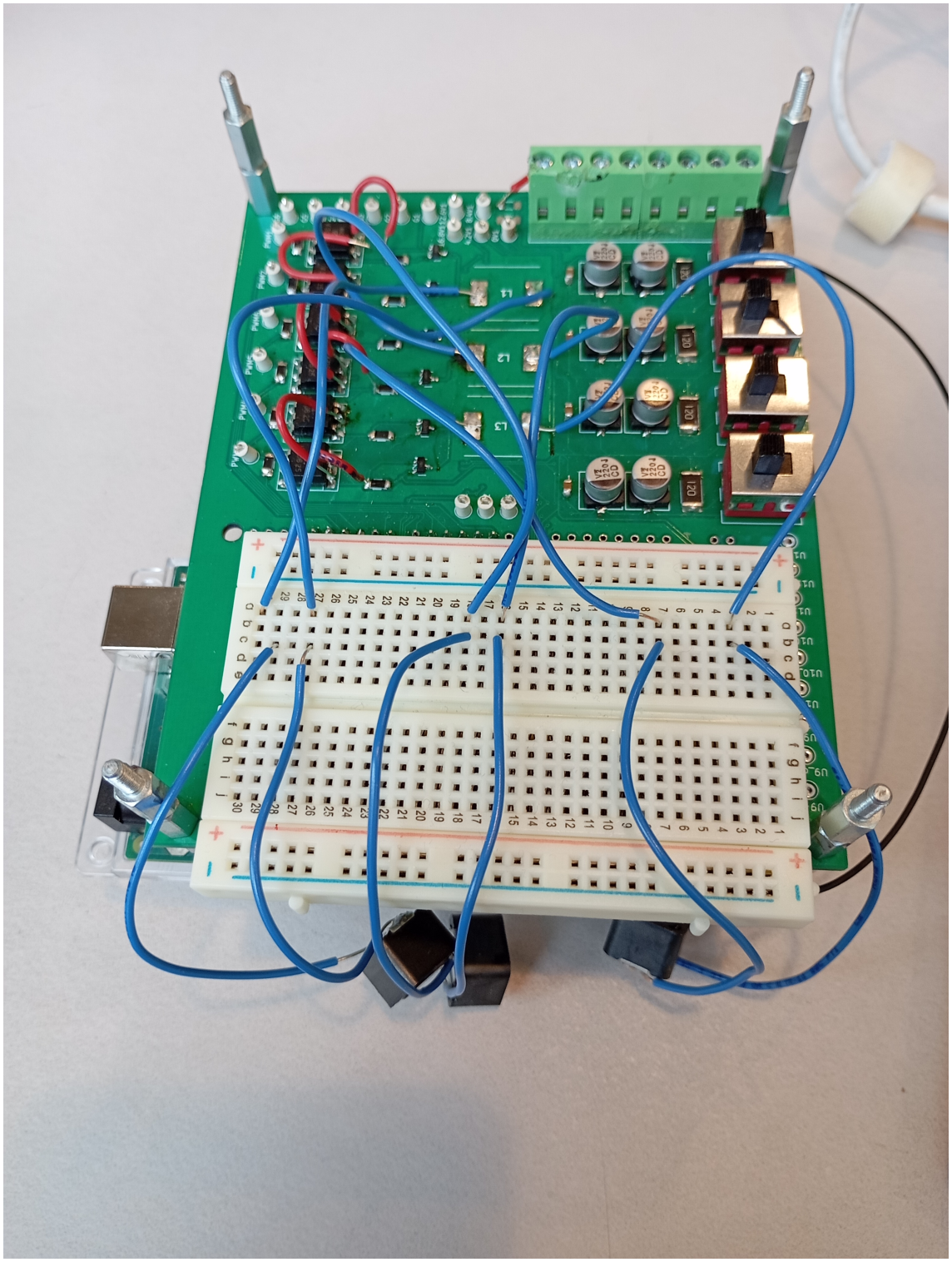


Figure B.2: The breadboard used to connect the inductors and PCB

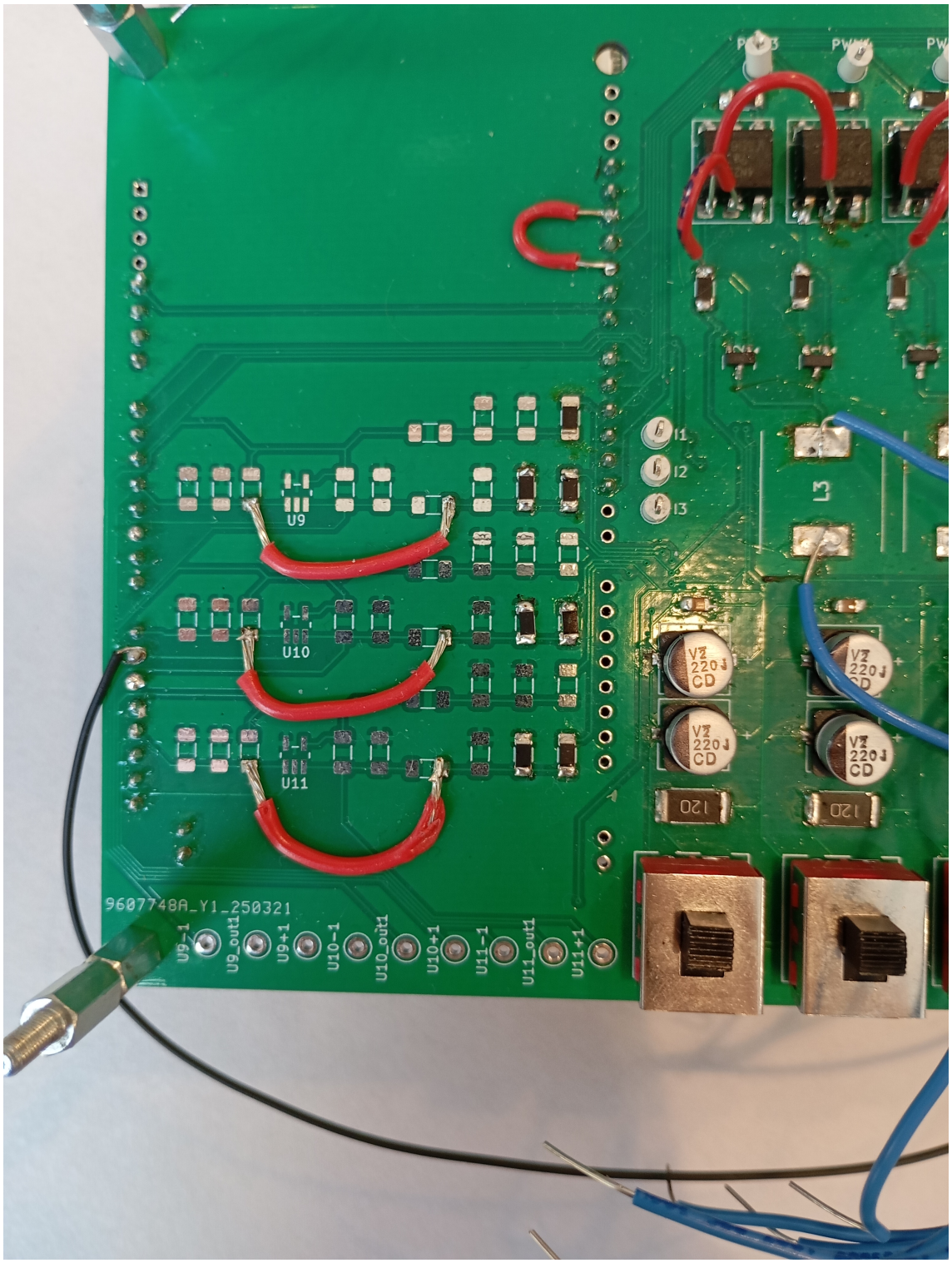


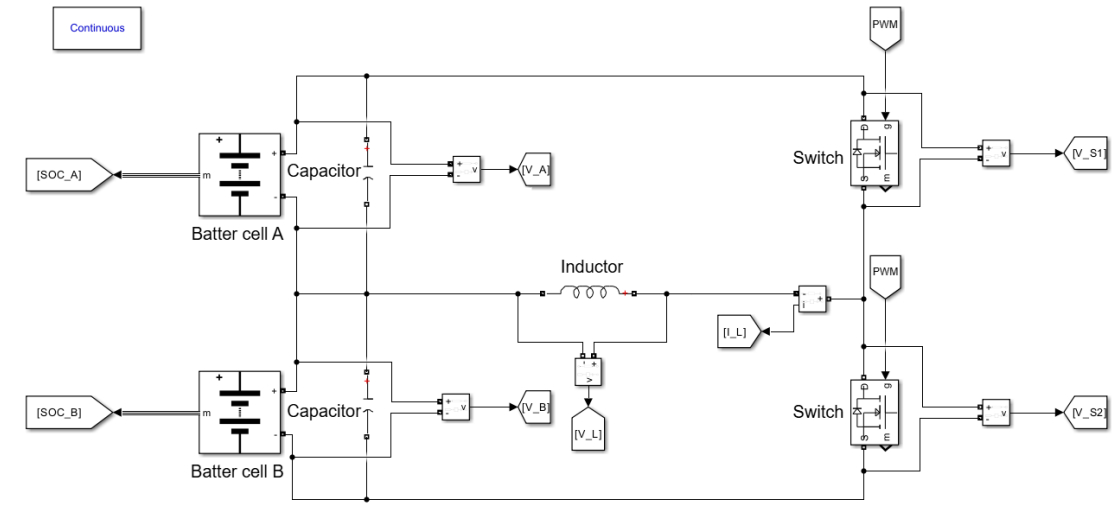
Figure B.3: The voltage dividers connections to the Arduino

Appendix C

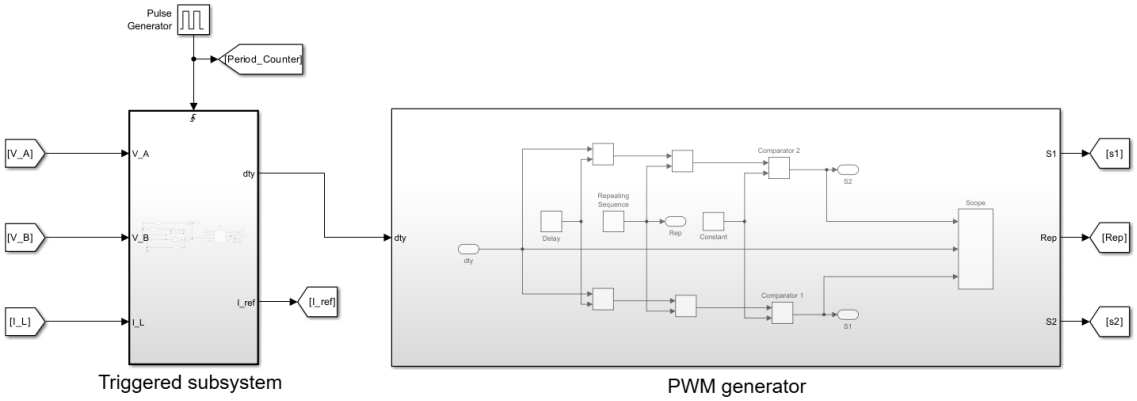
Table C.1: Result from voltage measurement tests

	Multimeter (V)	Arduino (V)	Error (%)
V1	2.50	2.53	1.16
	3.00	3.04	1.27
	3.49	3.55	1.55
	3.99	4.07	1.68
	4.20	4.27	1.74
V2	5.00	5.06	1.22
	6.00	6.09	1.45
	7.00	7.11	1.56
	8.00	8.13	1.71
	8.40	8.54	1.75
V3	7.52	7.61	1.20
	9.02	9.14	1.41
	10.77	10.94	1.55
	12.76	12.97	1.68
	13.56	13.80	1.76
V4	10.01	10.13	1.18
	12.01	12.19	1.53
	14.00	14.21	1.50
	16.01	16.28	1.67
	16.81	17.09	1.66

Appendix D

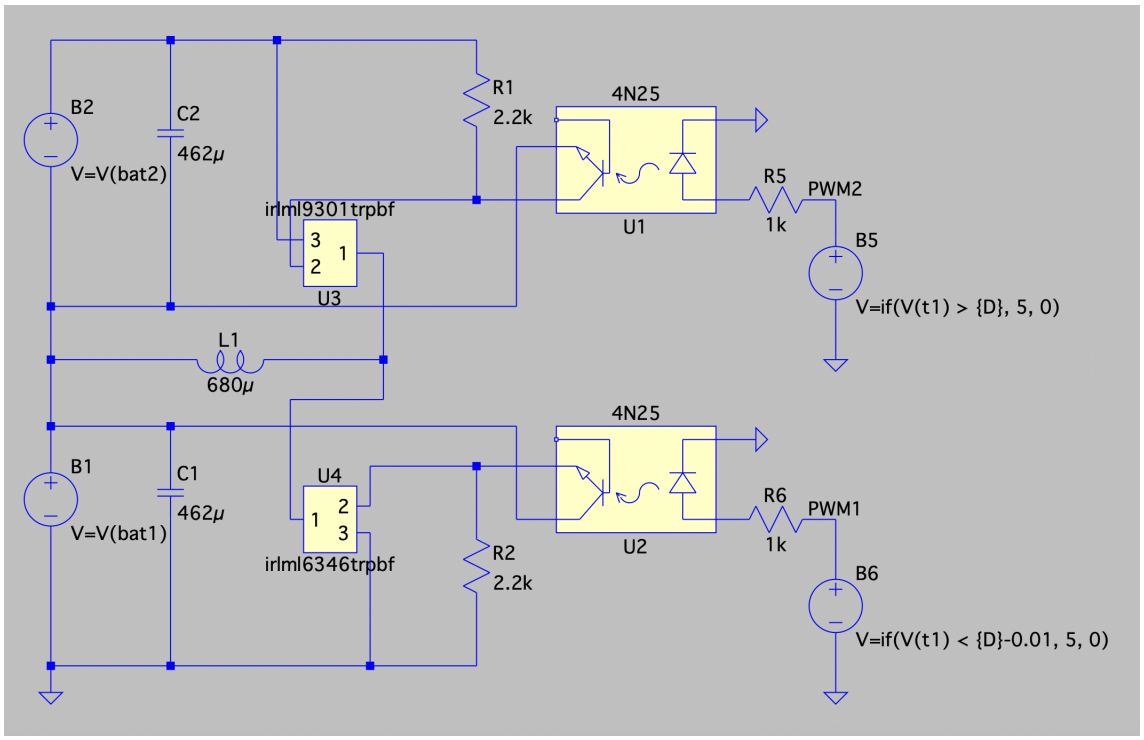


(a)

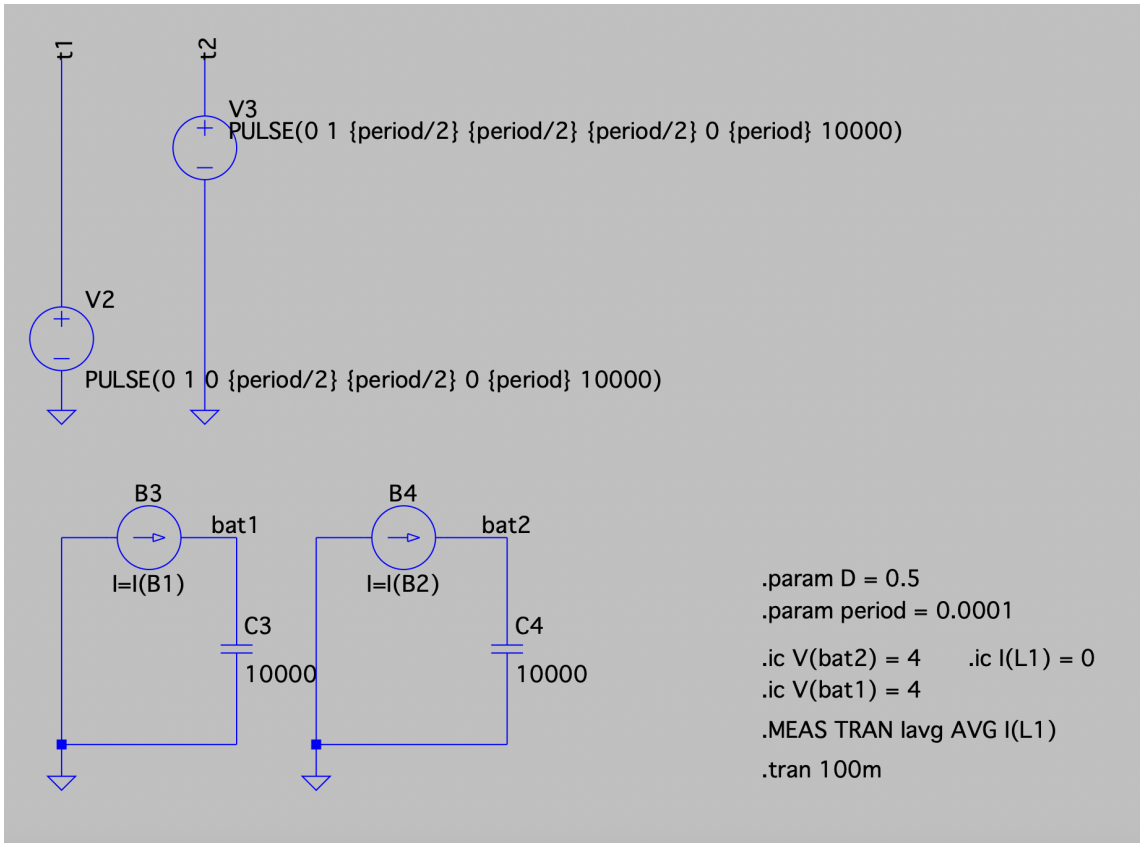


(b)

Figure D.1: Setup used in Simulink simulations. Sub-figure (a) displays the circuit, the triggered subsystem in (b) houses the control system from Figure 2.12. The PWM generator houses the necessary logic for generating PWM signals.



(a)



(b)

Figure D.2: Setup used for LTspice simulations, with charge distribution circuit in (a) and pulse generation and simple battery cell models in (b).

Appendix E

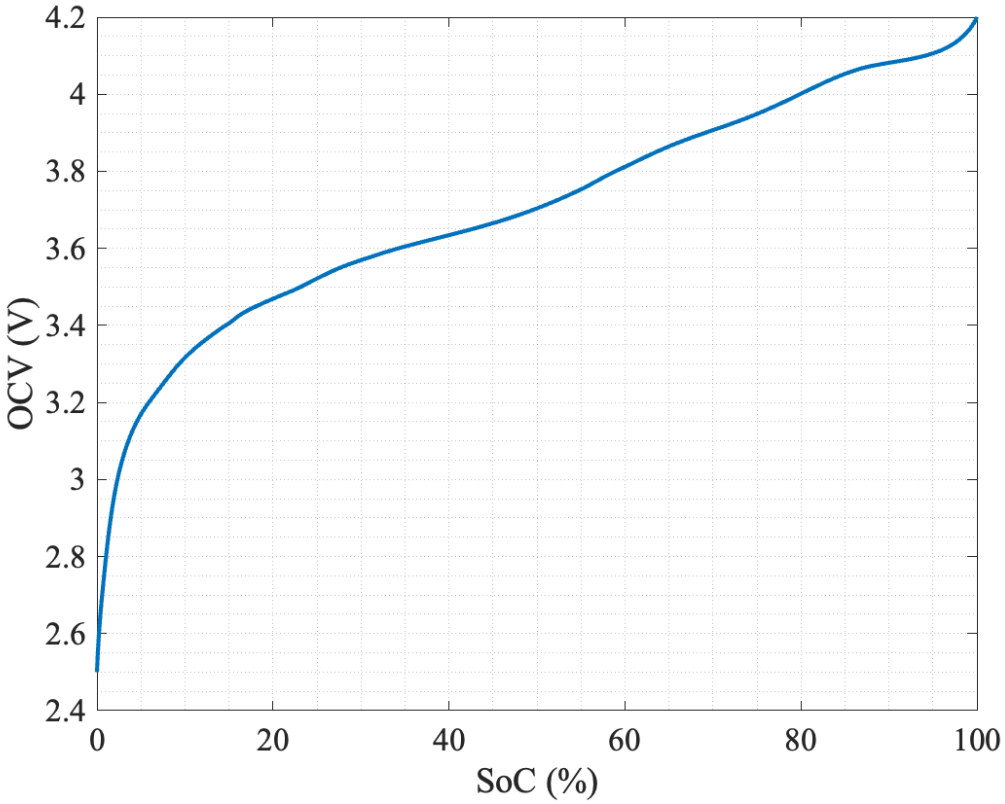


Figure E.1: Measured SoC to OCV relation for the INR18650 MJ1 3500 mAh battery cells (provided by S. Lundberg, personal communication, February 7, 2025).

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY