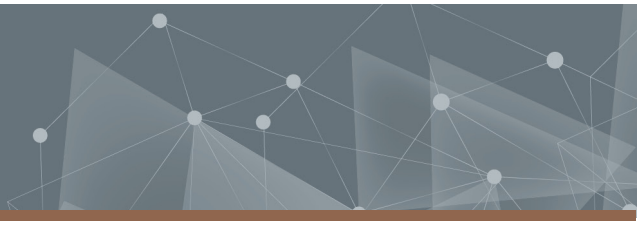
CHALMERS
UNIVERSITY OF TECHNOLOGY



Fast charging using equivalent circuit models

Optimizing lithium-ion battery fast charging using an ECM-based approach

Master's thesis in Systems, Control and Mechatronics

MÅRTEN LUNDSTRÖM
ALEXANDER RYDEVALD

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

Master's thesis 2022

Fast charging using equivalent circuit models

Optimizing lithium-ion battery fast charging using an ECM-based approach

MÅRTEN LUNDSTRÖM
ALEXANDER RYDEVALD

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Automatic Control
Chalmers University of Technology
Gothenburg, Sweden 2022

Fast charging using equivalent circuit models
Optimizing lithium-ion battery fast charging using an ECM-based approach
MÅRTEN LUNDSTRÖM
ALEXANDER RYDEVALD

© MÅRTEN LUNDSTRÖM, ALEXANDER RYDEVALD, 2022.

Supervisor: Björn Fridholm, Volvo Car Group
Examiner: Torsten Wik, Department of Electrical Engineering

Master's Thesis 2022
Department of Electrical Engineering
Division of Automatic Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2022

Fast charging using equivalent circuit models
Optimizing lithium-ion battery fast charging using an ECM-based approach
MÅRTEN LUNDSTRÖM
ALEXANDER RYDEVALD
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Fast charging is a key aspect of a battery-electric vehicle, impacting both the usefulness, competitiveness and sustainability of the vehicle. Charging speed is, among other factors, limited by degradation mechanisms such as lithium plating. By using improved models of when lithium plating may occur, charging can be done less conservatively and thereby faster. A key factor determining the risk of lithium plating is the negative electrode overpotential, which should be kept positive to avoid this.

The main tool used in the project has been simulations using the battery modelling software PyBaMM [1]. Using this and an equivalent circuit model (ECM), a model was designed with the negative electrode overpotential as output. PI control was then applied on the system to keep the overpotential positive.

Results were compared with a conventional multistage constant-current, constant-voltage (MCC-CV) algorithm. In an ideal case, the ECM-based method was able to improve charging time by 9.7% to 70% SoC compared to the MCC-CV algorithm. More aspects such as temperature and aging (including gradients on pack level) must, however, be taken into account before the algorithm can be applied in a real-life application. This will make the process of finding accurate model parameters, which was a major topic in this project, a more complex task. The results do however show the potential for improved charging time using this method.

Keywords: lithium-ion, fast charging, degradation, control, equivalent circuit, model, algorithm.

Acknowledgements

We would like to express our gratitude for the guidance, support and good collaboration with our supervisor, Björn Fridholm. Thanks for taking the time to provide us with continous support and input which have without a doubt helped us reach better results. Thanks also to Velibor Djordjic for the swift help with getting us set up, and to Volvo Car Group for providing us the excellent opportunity to do a thesis in this exciting field.

We would also like to extend a big thank you to our examiner, Prof. Torsten Wik and his team at Chalmers, including Asst. Prof. Changfu Zou and Yang Li, for the valuable input and guidance. We could not have asked for a better team to help us.

Finally, many thanks to the team behind Python Battery Mathematical Modelling (PyBaMM), especially to Valentin Sulzer who have been available to directly help out users like ourselves with big and small queries. This project would not have been possible without such a versatile software. Thank you also to Oskar Nordlander, who also used PyBaMM in his thesis, for many good discussions and the shared learning about PyBaMM.

Mårten Lundström and Alexander Rydevald, Gothenburg, June 2022

Contents

List of Figures

List of Tables

List of Acronyms

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

BEV	Battery-Electric Vehicle
BMS	Battery Management System
CC	Constant-Current
CC-CV	Constant-Current Constant-Voltage
CV	Constant-Voltage
DFN	Doyle-Fuller-Newman (model)
ECM	Equivalent Circuit Model
FPM	Full Physics Model
Li-ion	Lithium-ion
MCC-CV	Multi-stage Constant-Current Constant-Voltage
NEOP	Negative Electrode Overpotential
OCV	Open Circuit Voltage
OP	Overpotential
P2D	Pseudo Two-Dimensional (model)
PI	Proportional Integral (control)
PyBaMM	Python Battery Mathematical Modelling [1]
SEI	Solid Electrolyte Interphase
SoC	State-of-Charge
SoH	State-of-Health

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

k	Discrete time step

Variables

v	Terminal voltage
i	Current
z	State-of-Charge
T	Absolute temperature
η	Overpotential

Parameters

R_0	Ohmic battery resistance ("internal resistance")
R_1	Electrical resistance, part of ECM dynamics
C_1	Electrical capacitance, part of ECM dynamics
τ_1	Time constant, part of ECM dynamics
R_2	Electrical resistance, part of ECM dynamics
C_2	Electrical capacitance, part of ECM dynamics
τ_2	Time constant, part of ECM dynamics
c_s	Solid phase lithium concentration
r	Particle radial distance

D_s	Solid phase diffusion coefficient
σ	Electrode electronic conductivity
ϕ_s	Solid phase potential
ϕ_e	Electrolyte potential
j_f	Volumetric transfer current density
A	Electrode surface area
L	Distance between current collectors
κ	Electrolyte ionic conductivity
κ_D	Electrolyte diffusional conductivity
ϵ_s	Electrolyte volume fraction
c_e	Electrolyte lithium concentration
D_e	Electrolyte phase diffusion coefficient
t	Lithium ion transference number
F	Faraday's constant
U_{ocp}	Open circuit potential
a_s	Solid/electrolyte interface area
i_0	Exchange current
R	Universal gas constant

1

Introduction

With the transition to battery-electric cars, a number of new properties become important for customers. Driving range is an important factor, but the fast charging capability could be argued to be of even greater importance as it has the greatest impact on the time required for long-distance journeys. It is also related to the battery size, so by improving the relative (per cell) charging speed, the overall charging speed can be maintained with a smaller battery pack, allowing for cost, weight and environmental savings. The charging speed of lithium-ion batteries is limited by factors such as the charging hardware (AC to DC converter), temperature and degradation mechanisms. Some of the known degradation mechanisms can be modelled using advanced models, which may not be implementable in the battery management system (BMS) of a vehicle. This project has therefore investigated whether it is possible to use equivalent circuit models (ECMs) to estimate degradation, with focus specifically on lithium plating, in order to improve the charging performance of electric vehicles.

The project has been conducted as a Master's Thesis project at the Department of Electrical Engineering at Chalmers University of Technology, Gothenburg, Sweden. It is a collaboration with Volvo Car Group AB, where the project was primarily carried out.

1.1 Purpose

The main purpose of this project has been to find an improved model-based fast charging control system for Li-ion battery packs in automotive applications through the use of ECMs, with the aim of reducing the charging time of the pack while being aware of and avoiding to violate known causes of accelerated degradation. In order to not develop an algorithm that is too computationally complex to be feasibly implemented in a vehicle's BMS, modelling efforts focused on equivalent circuit models. The work based itself on previously conducted research within this area, and through the knowledge gained, the aim was to create and simulate a more sophisticated charging control system. The results were compared to a baseline MCC-CV charging system, similar to what is used in Volvo cars today.

1.2 Objective

The project's objective can be described and summarized with the following research questions:

- Can an ECM be used as a tool for estimating internal states such as the negative electrode overpotential (NEOP)? Specifically, is it possible to model the NEOP behavior using ECM-based models?
- What types of ECMs are suitable for this purpose in terms of computational complexity and accuracy, and how shall its parameters best be estimated?
- Is the output from the ECM accurate enough to be used for controlling current, and what method is suitable to perform this control? Does the complete system result in a time improvement compared to conventional charging algorithms?

1.3 Scope

In order to allow the project to fulfill its purpose within the time and resource constraints, a number of delimitations have been specified so that the work focus on finding a novel charging algorithm by the use of ECMs for state estimation, which ensures that known degradation causes are not violated while being feasibly implementable in a BEV. The project has aimed to show whether an improvement in charging time is possible, but will not go into depth to adapt the system to handle the impact of various external factors which would impact the system if applied to a vehicle battery pack.

A challenge of fast charging an electric vehicle is the multitude of factors that affect the process. A battery pack used in a car can often contain hundreds, or even thousands of individual battery cells, each of which may have slightly differing properties due to manufacturing, or due to varied levels of degradation, for example.

Depending on the construction of the battery pack and its heating/cooling system, this can, for example, cause temperature gradients in the pack where cells in the outer areas of the pack are cooler than cells in the middle of the pack. As the battery pack as a whole needs to be charged uniformly (all cells at approximately equal levels of SoC), a large temperature range of the individual cells will limit the temperature range in which the complete battery pack can charge at its theoretical maximum rate. Additionally, the cell resistance is known to be a function of temperature [3], which means that temperature also directly affects the electrical attributes of each individual cell. This can, in turn, aggravate aging effects on individual cells [4]. To simplify, it is here assumed that the battery pack construction and its cooling system allows for a homogeneous temperature of the entire pack. Furthermore, individual manufacturing variations between cells have not been considered.

It is also assumed that the thermal management not only keeps the pack temperature

homogeneous, but also constant. This is to remove rising temperature as a factor, which would normally impact the models and in turn the entire control system.

The assumption of homogeneous cells with constant temperature has the consequence that a model and algorithm developed for a single cell can easily be scaled to apply for charging an entire battery pack. Therefore, only charging on cell-level, and not pack-level, is considered.

General estimation algorithms, such as a sophisticated algorithm for estimating SoC, is not implemented or developed here. Instead, simplified algorithms will be used, based on data available from the modelling software used.

The project does not investigate what impact the use of alternative cell chemistries have on the performance of the charging algorithm. Instead, one type of cell chemistry will be used throughout the work.

Results from the developed systems will be evaluated for two cases - with a limit on the charging current of maximum 3 C (15 A), and without any limit on the charging current. The limited case aims to take into consideration that hardware limitations (maximum output of charging stations as well as current limitations of in-vehicle cabling) are present in reality, but also to investigate whether allowing an even higher charge current provide much benefit or not.

Lastly, in order to validate the performance of the proposed ECM, the selected Full Physics Model (FPM) will be considered the exact model reference from which the correct negative electrode overpotential can be derived and compared with the one estimated from the ECM.

1.4 Report overview

This thesis report begins with a background and introduction to the factors involved in the problem of battery fast charging. This includes a motivation for why improving fast charging is beneficial, as well as various aspects to consider, such as degradation and temperature. Specifics about the models and simulation parameters in use are also presented.

Once a technical foundation has been established, the various methods used are presented in Chapter 3. An overview of the development process is presented in Section 3.1, followed by more detailed descriptions of how the work has been carried out. This includes the mathematical modelling of the ECMs, determination of their parameters and the PI control that was used in the system.

Following this, the results are presented in Chapter 4. This includes each of the parameter functions that were used, as well as performance plots and numbers of the various complete systems that were created. This is followed by an analysis and discussion in Chapter 5, including aspects which need further consideration before the proposed solution is suitable for real-life implementations, as well as sustainability and ethical aspects of the work. A conclusion of the report can be

found in Chapter 6.

2

Background

With the beginning of a new decade heavily driven by society's efforts to limit the climate change, private transportation in form of battery-electric vehicles (BEVs) powered by lithium-ion (Li-ion) batteries has become popular. The prospect of eventually replacing all conventional combustion engine cars with EVs has been brought up during the United Nations Climate Change Conference (COP21) [5]. This shared opinion puts pressure on car manufacturers around the world to find electromobility solutions, which in the relatively near future could take over the private transportation sector.

In connection with the car industry shifting towards larger amounts of BEV applications, the charging performance of the batteries constantly needs improvements in order to match the competition on the market. For example, due to limited space and weight requirements for battery packs in the vehicle, high quality requirements are set on the batteries to be fast-charging while maintaining long-distance capability and a long life span. This requires the battery pack to be designed with consideration to use in different temperatures, aging and, importantly, to avoid degradation effects whenever possible [4]. The pack of batteries is also classified as the single most expensive component of the car, making up as much as 36% of the total manufacturing cost of a 72 kWh electric vehicle, according to 2022 estimates [6]. Consequently, a potentially significant way of reducing production costs is to identify improvement aspects of the current battery management system.

Volvo Cars, one of the globally most well known automotive manufacturers, has set the goal of shifting their product line-up completely to fully electric vehicles by the year 2030 [7]. In order for this transition to be successful, it is key to offer competitive products to not lose market shares on the global market. By improving the charging performance, the vehicles can charge more energy during a given time period and battery pack size. This also enables the possibility of reducing the needed battery pack capacity while maintaining competitive long-distance driving capability. By offering this, Volvo Cars can lower the production cost and weight of the vehicle, and also reduce the climate impact from the manufacturing. Therefore, an important task for the company is to develop charging algorithms that without major drawbacks can recharge BEVs in a shorter time.

2.1 Degradation mechanisms

There are a number of factors that limit the charging speed of Li-ion batteries. Apart from the power output of the charging equipment, which will not be considered as a limiting factor in this project, the battery's temperature needs to be kept within a specific range in order to not pose safety concerns, and consideration must be taken to avoid conditions where the battery cell might develop irreversible degradation effects.

Subjecting the battery cell to high temperatures changes the chemical properties of the components in the battery. For example, thermal decomposition of the electrolyte will occur at higher temperatures, a reaction which becomes significant above 85° C [4]. At a certain point, such reactions reach a point where they happen at a rate which by itself accelerates the thermal energy buildup, i.e. positive feedback occurs. This is what is commonly referred to as a thermal runaway and it is of course crucial to avoid. To do so, the battery must be kept at a safe operating temperature at all times.

There are two main types of degradation effects to be considered with Li-ion batteries: Lithium deposition and mechanical degradation. Mechanical degradation is a collective term for various physical changes to the battery cell and is generally occurring in conjunction with fast charging [4]. Examples include fractures in the electrode particles and separation between the active material and the current collector. It can occur as high lithium concentration gradients may cause volume and dimensional changes, leading to mechanical stress [8].

Mechanical degradation is however difficult to decouple from other degradation mechanisms such as various types of lithium deposition, and the level of knowledge is also lower compared to other degradation types [4], so further work is needed on this topic. The focus when discussing Li-ion degradation is therefore often on the lithium deposition-related degradation mechanisms, and this type of degradation mechanisms include solid electrolyte interphase (SEI) layer growth and lithium plating.

The SEI layer is formed on the negative electrode (anode) surface during the first cycling of the battery, and is necessary for allowing Li-ion transport while blocking electrons, preventing damage to the electrode material [9]. However, as SEI is formed, lithium in the electrolyte is made unavailable, and due to this, the capacity of the battery cell is reduced [4]. Additional SEI layer formation may occur if the battery is charged at high temperature, and therefore this is to be avoided.

During charging, metallic lithium may also start to deposit on the anode surface. In certain circumstances, the deposition may continue on the previously formed lithium deposits, forming metallic structures known as dendrites [4]. This phenomenon is known as lithium plating. Lithium plating has been shown to be largely preventable by ensuring that the negative electrode overpotential is kept above a reference value, and if the battery is charged at too high current, the overpotential will drop below this reference value [4]. The risk of the negative electrode overpotential going too

low is also aggrevated when charging at low temperatures. However, in a vehicle battery pack, the only electrical cell variables that can reasonably be measured are the cell voltage and current, as well as temperature. The negative electrode potential is not measurable using regular equipment that may be used in a vehicle's battery pack, and must therefore be estimated.

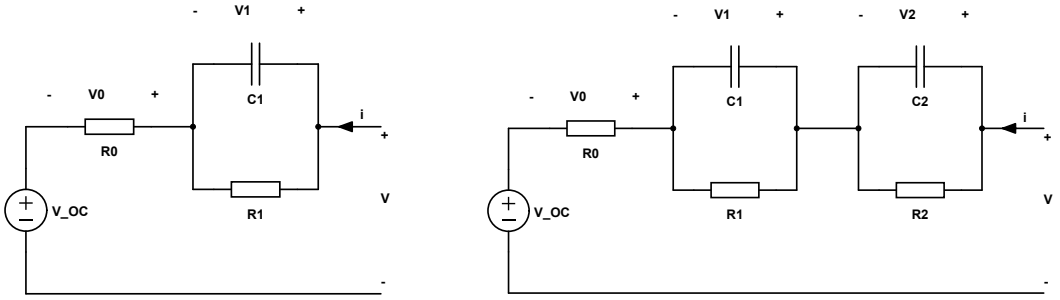
There are however methods for estimating the anode potential, for example a pseudo two-dimensional (P2D) model known as the Doyle-Fuller-Newman (DFN) model [10]. However, a problem applying this model to solve the challenge of fast charging is its computational complexity [4]. Therefore, attempts to apply the limitation of anode potential in a feasibly implementable charging algorithm must use a reduced-order model of the P2D model, or another form of simplification. One approach, which will be the main subject of this work, is to use an equivalent circuit model (ECM) as a tool for battery state estimation [11].

2.2 Battery modeling

Battery modeling can be done through two main approaches [12] - electrochemical modeling and equivalent circuit-based modeling. Electrochemical modeling is based on the ion concentrations and potentials in the electrodes and the electrolyte, as well as the reaction flux [4]. It can be used to describe side reactions in the battery with high accuracy. However, due to its high computational complexity, it is generally not a viable option for BMS applications.

The second approach, equivalent circuit-based models, have lower computational complexity, but if and how they can be used for estimation of internal states related to fast charging, and with what accuracy and robustness, is subject to ongoing research. The simplicity of ECMs comes from the electric circuit components being only resistances and capacitances that are divided into one or multiple resistor-capacitor (RC) blocks. This is described by the Thevenin model in which the ohmic battery resistance (R_0) is connected to parallel RC networks consisting of polarization resistances and capacitances to mimic the dynamic characteristics that happens during charging and discharging.

The most commonly used battery cell ECMs are of first or second order, see Figure 2.1. The order signifies the number of RC-circuits connected in series. Its parameters need to be estimated, and they, as well as the open circuit voltage in the ECM, may be dependent on factors such as the battery's state-of-charge (SoC) [12]. The parameters can be estimated offline based on voltage and current data which can be collected or experimentally retrieved from a P2D model.



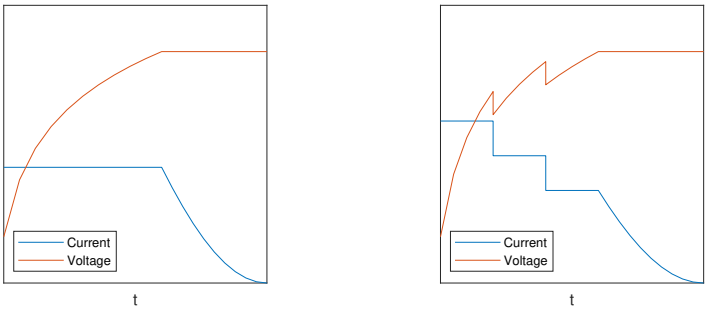
(a) 1st order ECM

(b) 2nd order ECM

Figure 2.1: Two Thevenin's equivalent circuit models of Li-ion battery

2.3 Common charging algorithms

Modern strategies for fast charging of Li-ion batteries such as CC-CV, MCC-CV and pulse charging differs from each other in charging speed but also in the amounts of degradation they cause on the batteries [4]. Constant-Current, Constant-Voltage (CC-CV), a commonly used charging pattern, consists of two phases. During the first phase, a constant, typically high current, is applied to the battery, which results in a high charging rate. However, the high current may cause accelerated degradation of the cell if applied during long time periods. Therefore, a second phase, where the voltage is instead kept constant is introduced, in order to reduce the degradation effects. The strategy of shifting between constant current (CC) and constant voltage (CV) is used by the charging algorithms CC-CV and Multi-Stage CC-CV (MCC-CV), see Figure 2.2. In MCC-CV, several CC phases are used in order to utilize a higher current during low SoC where the risk of lithium plating is lower, which is then reduced through a number of "steps" as the charging progresses and SoC increases.



(a) CC-CV algorithm

(b) MCC-CV algorithm

Figure 2.2: Two common charging algorithms for Li-ion batteries

Attempts have been made to find complementing solutions to the relatively low current CV region of CC-CV, one of them being boost charging [13], an approach used on fully discharged batteries which in addition to standard CC-CV begins by letting the battery reach its maximum allowable current. This feature enables quicker charging speed, and it can be used until SoC has reached 40% without risking deposition of lithium [14].

Another way of recharging batteries is to frequently introduce short relaxation periods in the typical CC charging profile [4], which decelerates the decreasing cathode potential during loading. This method can prevent lithium plating from happening, while charging at a higher rate than otherwise allowable in a conventional CC-CV algorithm [15].

2.4 Doyle Fuller Newman model

The DFN model, sometimes referred to as the P2D model due to it containing one-dimension for the particle and one for the electrolyte, comprises partial differential equations for charge and mass conservation in the solid (electrodes) and electrolyte (separator), and also prescribes behavior for the electrochemical reactions occurring on the interface between the solid and electrolyte [2]. It is the most common Full Physics Model (FPM) of lithium-ion batteries and captures the characteristics on cell level. Some equation parameters of the DFN model are directly obtained from the material properties. Others, however, need to be retrieved from lab tests and experimental data [16].

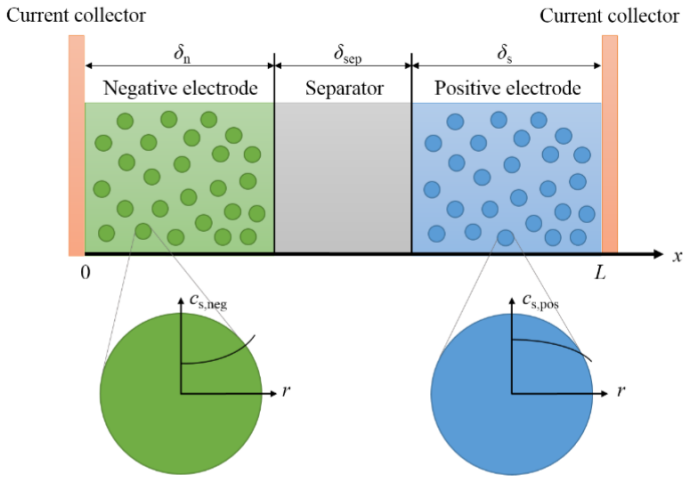


Figure 2.3: Schematic figure of a lithium-ion cell with dimensions matching the DFN model [2]

Starting with the equations describing what occurs in the solid, i.e. the particles of the electrodes corresponding to the green and blue parts of Figure 2.3, the equation describing the flux of Li-ions is given by

$$\frac{\partial c_s}{\partial t} = \frac{D_s \partial}{r^2 \partial r}(r^2 \frac{\partial c_s}{\partial r}), \tag{2.1}$$

where c_s is the solid phase lithium concentration, r the particle radial distance and D_s the solid phase diffusion coefficient.

Charge conservation in the solid phase yields

$$\frac{\partial}{\partial x}(\sigma \frac{\partial}{\partial x}\phi_s) - j_f = 0, \tag{2.2}$$

where σ is the electrode electronic conductivity, ϕ_s the solid phase potential and j_f the volumetric transfer current density.

The boundary condition of the DFN model is described by

$$-\sigma \frac{\partial}{\partial x}\phi_s(t,0) = \sigma \frac{\partial}{\partial x}\phi_s(t,L) = \frac{i(t)}{A}, \tag{2.3}$$

where A is the electrode surface area.

Within the electrolyte phase there is another set of model equations. One of them is the charge balance, given by

$$\frac{\partial}{\partial x}(\kappa \frac{\partial}{\partial x}\phi_e) + \frac{\partial}{\partial x}(\kappa_D \frac{\partial}{\partial x} \ln c_e) + j_f = 0, \tag{2.4}$$

where ϕ_e is the electrolyte potential, κ the electrolyte ionic conductivity, c_e the electrolyte lithium concentration and κ_D the electrolyte diffusional conductivity.

The dynamics of the conservation of lithium within the electrolyte phase are described as

$$\frac{\partial}{\partial t}\epsilon_e c_e = \frac{\partial}{\partial x}(D_e \frac{\partial}{\partial x} c_e) + (1 - t_+)\frac{j_f}{F}, \tag{2.5}$$

where ϵ_s is the electrolyte volume fraction, D_e the electrolyte phase diffusion coefficient, t_+ the lithium ion transference number and F denotes Faraday's constant.

For this project an important variable is the overpotential η which, as mentioned in Section 2.1, relates to lithium plating and is given by the difference between the potential in the solid phase and the electrolyte, minus the open circuit potential U_{ocp} and a term caused by the resistance in the SEI film between the solid and the electrolyte, i.e.,

$$\eta = \phi_s - \phi_e - U_{ocp} - \frac{R_{sei}}{a_s} j_f, \tag{2.6}$$

where a_s is the solid/electrolyte interface area.

The current density is a function of the overpotential and is given by the Butler Volmer relation,

$$j_f = 2a_s i_0 \sinh(\frac{F}{2RT}\eta), \tag{2.7}$$

where i_0 is the exchange current, R the universal gas constant, and T the absolute temperature.

The terminal voltage is simply the potential difference between the positive and negative electrode,

$$v(t) = \phi_s(t, L) - \phi_s(t, 0). \tag{2.8}$$

2.5 LG M50 battery cell

As explained further in Chapter 3, the charging method development will make use of the Python-based open-source battery modeling software Python Battery Mathematical Modelling (PyBaMM) [1]. Although there exist many ready to use battery models in PyBaMM, simulation of negative electrode overpotential (NEOP) is limited to only a couple of them, something that was confirmed early in the project when learning about the simulation environment and checking the available outputs for every battery model. The battery model chosen for this project is a cylindrical 21700 commercial cell (LG M50), and is based on parameter data from [16], with a set of 35 parameters required to run the DFN model, from which NEOP can be determined. The parameter settings are included with PyBaMM and can be used by choosing `Chen2020_plating` as the parameter set in the simulation. Since the DFN model is a physics-based model, it is computationally expensive to calculate the model equations and not feasible to do in a car's BMS. However, by performing simulations offline on the actual cell and observing its behavior it is possible to obtain valuable information that can be used for the ECM. The DFN model then serves as a perfect description of a battery, and the goal is to have the ECM imitate it as well as possible.

2.5.1 Parameterization of LG M50

In order to obtain the parameters with regards to the physical properties of the LG M50 cell, it is required to dismantle the cell so that each individual component can be studied and measured separately, a process that does not come without risks since the dismatling process may compromise the cell. However, in this project a full set of DFN parameters are already included in the simulation environment and performing this somewhat tricky process is therefore not needed. Parameters from

the DFN equations that are relevant here are the ones related to the electrodes and seperator and are obtained using direct measurements. Porosity and pore domain size can be determined using kerosene radiation such as X-ray [16].

In terms of the chemical and material properties, strategies involving use of spectroscopy and X-rays can be used [16].

Electrochemical parametrization can involve the use of two-electrode full cells, half cells and three-electrode configurations. It is preferable to use a combination of these when studying the electrochemical properties due to their varying effectiveness. The half cell is used for observing the negative electrode or positive electrode.

The electrolyte properties, such as the transfer, ionic conductivity and ionic diffusivity were taken from a standard electrolyte. However, the tests that were performed by the authors of the article which the model used is based on [16] showed no indications of this being an issue.

The PyBaMM simulation of the LG M50 cell uses the parameter set listed in Appendix A. The majority of the parameters in this list originate from [16] while the rest are taken from the general CRC Handbook of Chemistry and Physics.

3

Method

In the project simulations have been used as a tool for evaluating various charging algorithms. This allows for quicker and less costly results, and allows for testing various adaptions of the algorithms in order to find better results. Specifically for ECMs, their state estimation performance can also be compared to the more accurate but complex FPM to check their accuracy.

3.1 Overview of the development process

Throughout the project, the development of model-based charging methods made use of a Full Physics Model (FPM), which simulates the cell's external and internal states based on selected battery parameters and current profile over the charging session. Simulations of the FPM were done using the Python-based open-source battery modeling software PyBaMM [1].

As a first step, the data collected through running simulations were used to develop an MCC-CV algorithm. This was done by comparing when the negative electrode overpotential (η) drops below the safely allowable value (η_{ref}) to the corresponding cell voltage, which is a measurable state. The MCC-CV algorithm was thereby designed to avoid that the negative electrode overpotential drops below the threshold where there is risk of lithium plating.

In an attempt to take into account external factors and modeling and measurement errors, which would be present in a real implementation, two cases were investigated:

- $\eta_{ref} = 0$ V. This is an idealized case and keeps the negative electrode overpotential above 0 V to avoid lithium plating, but with no margin to account for errors or uncertainties that would be present in reality.
- $\eta_{ref} = 0.025$ V. This is a more realistic case, taking into account the many factors of variation and model errors which may occur in a real implementation, such as different initial SoC, measurement errors and other variations. To take this into consideration, the algorithm keeps the negative electrode overpotential positive to avoid lithium plating, with a 0.025 V margin.

As a next step, the FPM was used to retrieve data about the negative electrode overpotential, a typically unmeasureable state, and a PI controller was constructed

to control the current so that the NEOP stays in the safe range (above η_{ref}). In addition, another PI controller was designed which similarly prevents the cell voltage from surpassing a maximum voltage of 4.2 V. A rule engine, described in more detail in Section 3.8, was also designed and selects which of the 2 controllers to use base on the system states. As this solution makes use of the FPM in controlling the current, it is likely too computationally complex to be implemented in a BMS. It does however provide a reference point for the potential performance the approach can achieve, and can be compared with results using the less complex ECMs developed in the project.

Finally, the algorithm based on modelling the negative electrode overpotential using an ECM was developed. This included several steps, including finding the open-circuit cell voltage and negative electrode overpotential curves, estimating the model parameters with dependence on SoC and charge rate, fitting suitable functions to the estimated parameters, and implementing this in a Python function to generate the output, which is then fed to the PI controller that was developed earlier.

To compare the performance of the PI-controlled ECM-based algorithm to that of the MCC-CV algorithm, it was evaluated in both the marginless case ($\eta_{ref} = 0V$) and with the margin $\eta_{ref} = 0.025V$ set as the reference value.

An overview of the PI-controlled system can be found in Figure 3.1. Note that the source of the negative electrode overpotential variable can easily be switched between the FPM and the ECM. This is also the case in the python-based code which the overview is based on.

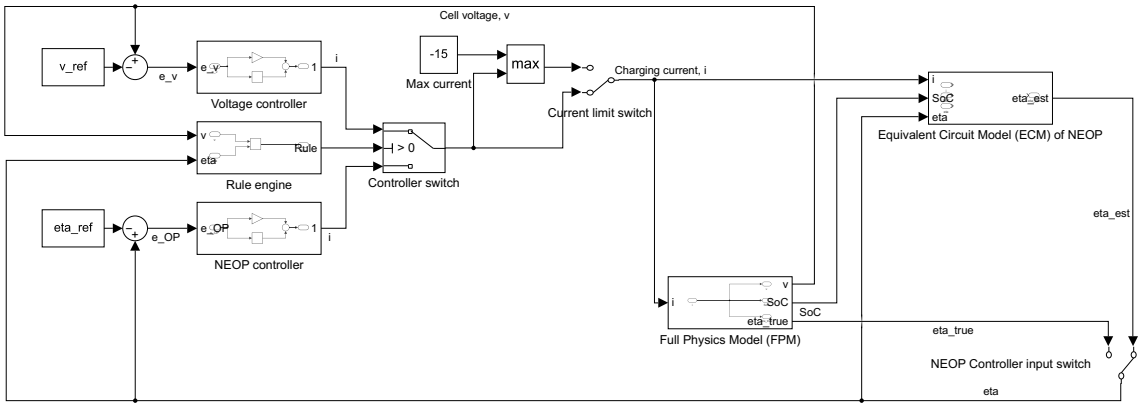


Figure 3.1: Overview of the PI-controlled system

3.2 Definition of SoC

While PyBaMM allows the user to set the initial SoC in percent, it does not provide a built-in SoC estimation algorithm. This was needed in the project since, for example, open circuit voltages and potentials, as well as ECM parameters, have a significant dependence on SoC. For this purpose, a basic SoC estimation algorithm was developed.

The estimation algorithm is based on the data PyBaMM provides of the current discharge capacity in Ah, Q (always 0 Ah at the start of the simulation), the fact that the battery being simulated has a maximum capacity of 5 Ah, and on the initial SoC set. The SoC is then calculated as follows:

$$SoC = \frac{-Q + SoC_{init}0.05}{5}100\ [\%], \tag{3.1}$$

where

$$Q = \int_0^t i(t)\,dt. \tag{3.2}$$

3.3 MCC-CV charging algorithm

The MCC-CV charging algorithm was designed in PyBaMM using its `Experiment` class. Using this, human text-like instructions are provided to the program, describing the process of the charging algorithm as follows:

```
experiment = pybamm.Experiment(
    [
        ("Charge at 3 C until "+V1_trig+" V",
        "Charge at 2 C for 50 seconds",
        "Charge at 2 C until "+V2_trig+" V",
        "Charge at 1.5 C for 50 seconds",
        "Charge at 1.5 C until "+V3_trig+" V",
        "Charge at 1 C until 4.2V",
        "Hold at 4.2 V until 100 mA"
        )
    ],
    period = "4 seconds"
)
```

The trigger voltages `V1_trig`, `V2_trig` etc, were found experimentally, first setting the experiment to charge constantly until η_{ref} (or v_{ref}) is surpassed, and find the corresponding cell voltage for when η_{ref} is reached, or set it to the maximum voltage, in case the corresponding cell voltage is higher than that. By modifying the experiment to jump to the next constant-current stage at this trigger voltage, the simulation can then be re-run to find the next trigger voltage, and so on until the maximum voltage can be reached, at which point the constant-voltage phase follows.

As can be noted in the code above, at higher charge rates, it was necessary to add manually timed instructions as well. The purpose of this is to allow the cell voltage drop caused by reducing the current to complete, since the trigger voltage could in some cases be lower than the trigger voltage of the previous stage.

For the marginless case, the CC stages 3 C, 2 C, 1.5 C and 1 C were chosen, but with the 0.025 V margin on the negative electrode overpotential. A 0.5 C CC stage was also needed.

3.4 Open circuit cell voltage/NEOP

As will be shown in Section 3.5, the mathematical expressions describing the equivalent circuit models used depend on having knowledge of what the open circuit voltage (OCV) is. The open circuit voltage (OCV) is here defined as the potential difference between the two terminals (electrodes) of the battery when fully at rest. It is a function of SoC, and was determined by the use of PyBaMM simulations.

The experiment which is used to find the OCV characteristics starts at 0% SoC with a rest period, after which it is charged in 2.5% or 5% increments, followed by a 10 minute resting period. 2.5% increments is used up to 15% SoC and after 95% SoC to give a more accurate OCV curve near the extreme values, as the OCV has a more nonlinear behavior there. This is repeated until 100% SoC has been reached. The simulation data are saved and a MATLAB script then reads the OCV at the end of each of the 10 minute resting period, to get an accurate reading of the OCV that is minimally affected by any previous load on the battery cell.

After the data of the OCV from 0% to 100% has been gathered, the data are saved to the file `ocv_data.pkl` which is later read by the Python program. 1D interpolation is used to estimate the SoC between the data points.

The experiment also saves data of the negative electrode open circuit overpotential, as this is a necessary component of the NEOP ECM. It is similarly saved and later interpolated as implemented in the following Python function, which returns the estimated OCV:

```
def SOCtoOPOCV(SoC):
    OP_OCV_data = pickle.load(open("ocv_data.pkl", "rb"))
    f = interp1d(OP_OCV_data['SoCs'], OP_OCV_data['OP_OCV'])
    return f(SoC)
```

3.5 Constructing the ECM

The simplified models which can be used for estimating cell voltage are modelled using the equivalent circuits shown in Figure 2.1. These models can also be defined by the following electrical relations and discrete-time state-space model of the RC components [2]. However, in order for the model to be accurate, the parameter values (R_0, R_1, C_1, etc.) must be estimated. The estimation of these parameters is key to achieve an accurate model, and getting good results from this estimation therefore became a substantial part of the project. The parameter estimation process is described in Section 3.6.

This section describes the development process of the various types of ECM used throughout the project. A first order ECM was first developed due to the simpler mathematical expression, but it was later found that a 2nd order ECM provided a beneficial increase in accuracy while not being too complex. To incorporate longer time-constant behavior observed which was not always included when letting all pa-

rameters of the 2nd order ECM be determined by the Least Square (LS) estimation, a 2nd order ECM with the parameters of one RC-pair fixed was developed.

As the ECMs were based on previous work on modelling cell voltage, the mathematical construction and parameter estimation of the ECMs were first done on the cell voltage ECM and is hence more thoroughly described. The adjustments that were made later, once the method of modelling the negative electrode overpotential with an ECM was chosen, is described later. The results in Chapter 4 are exclusively based on the 2nd order ECM with fixed parameters of one RC-pair, modelling the NEOP.

3.5.1 1st order equivalent circuit

Based on the electric circuit in Figure 2.1, and assuming piece-wise constant current, the following discrete-time model equations can be derived:

$$v_1(k+1) = e^{-\frac{\Delta t}{\tau}} v_1(k) + R_1(1 - e^{-\frac{\Delta t}{\tau}})i(k) \tag{3.3}$$

$$v(k) = v_{OC}(z(k)) + v_1(k) + R_0 i(k), \tag{3.4}$$

where $\tau = R_1 C_1$ and $z(k)$ is the SoC.

Substituting $a = e^{-\frac{\Delta t}{\tau}}$, $\Delta v = v - v_{OC}$ and rearranging the equations gives the following expression:

$$\Delta v(k+1) - R_0 i(k+1) = a(\Delta v(k) - R_0 i(k)) + R_1(1-a)i(k) \Rightarrow \tag{3.5}$$

$$\Rightarrow \Delta v(k+1) = a\Delta v(k) + R_0 i(k+1) - aR_0 i(k) + R_1(1-a)i(k) \Rightarrow \tag{3.6}$$

$$\Rightarrow \Delta v(k+1) = \underbrace{\begin{bmatrix}\Delta v(k) & i(k+1) & i(k)\end{bmatrix}}_{\varphi\tau(k)} \underbrace{\begin{bmatrix} a \\ R_0 \\ R_1(1-a) - aR_0 \end{bmatrix}}_{\theta} \tag{3.7}$$

As Δv and i are both easily available from BMS measurements, or in this case simulations, θ can be found by applying least squares to the regression (3.7).

The parameter estimation is then done by using a simulated test cycle suitable for parameter estimation, solving the LS problem by the use of MATLAB's "\" operator to get θ, and then from the components of θ solve for the parameters as shown below:

$$a = \theta_1 \tag{3.8}$$
$$R_0 = \theta_2 \tag{3.9}$$
$$R_1 = \frac{\theta_3 + aR_0}{1 - a} \tag{3.10}$$
$$C_1 = -\frac{\Delta t}{\ln(a)R_1} \tag{3.11}$$

3.5.2 2nd order equivalent circuit

The second-order equivalent circuit model is based on similar equations as the first-order model, with the addition of the voltage drop over the second RC-component, v_2, i.e.,

$$\begin{bmatrix} v_1(k+1) \\ v_2(k+1) \end{bmatrix} = \begin{bmatrix} e^{-\frac{\Delta t}{\tau_1}} & 0 \\ 0 & e^{-\frac{\Delta t}{\tau_2}} \end{bmatrix} \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} + \begin{bmatrix} R_1(1 - e^{-\frac{\Delta t}{\tau_1}}) \\ R_2(1 - e^{-\frac{\Delta t}{\tau_2}}) \end{bmatrix} i(k) \tag{3.12}$$
$$v(k) = v_{OC}(z(k)) + v_1(k) + v_2(k) + R_0 i(k). \tag{3.13}$$

Substituting $a_1 = e^{-\frac{\Delta t}{\tau_1}}$, $b_1 = R_1(1 - a_1)$, $a_2 = e^{-\frac{\Delta t}{\tau_2}}$, $b_2 = R_2(1 - a_2)$, $\Delta v = v - v_{OC}$ and rearranging the equations gives the following expression:

$$\Delta v(k+1) = a_1 \Delta v(k) + (a_2 - a_1)v_2(k) + R_0 i(k+1) + (b_1 + b_2 - R_0 a_1)i(k) \Rightarrow \tag{3.14}$$

$$\Rightarrow v_2(k) = \frac{1}{a_2 - a_1}\Delta v(k+1) - \frac{a_1}{a_2 - a_1}\Delta v(k) - \frac{R_0}{a_2 - a_1} i(k+1) - \\ - \frac{b_1 + b_2 - R_0 a_1}{a_2 - a_1} i(k) \tag{3.15}$$

Solving Equation (3.13) for v_1 and inserting the expression for v_2 from Equation (3.15) results in the following expression:

$$v_1(k) = \Delta v(k) - v_2(k) - R_0 i(k) \tag{3.16}$$
$$= -\frac{1}{a_2 - a_1} v_t(k+1) + (1 + \frac{a_1}{a_2 - a_1})v_t(k) + \frac{R_0}{a_2 - a_1} i(k+1) + \\ + (\frac{b_1 + b_2 - R_0 a_1}{a_2 - a_1} - R_0 i(k)) \tag{3.17}$$

To make the equations more compact, the coefficients applied to the variables are substituted as follows:

$$\begin{aligned}
\alpha_1 &= -\frac{1}{a_2 - a_1} \\
\alpha_2 &= 1 + \frac{a_1}{a_2 - a_1} \\
\alpha_3 &= \frac{R_0}{a_2 - a_1} \\
\alpha_4 &= \frac{b_1 + b_2 - R_0 a_1}{a_2 - a_1} - R_0
\end{aligned} \tag{3.18}$$

This gives the following expression for v_1:

$$v_1(k) = \alpha_1 \Delta v(k+1) + \alpha_2 \Delta v(k) + \alpha_3 i(k+1) + \alpha_4 i(k), \tag{3.19}$$

which can also simply be time-shifted one sample:

$$v_1(k+1) = \alpha_1 \Delta v(k+2) + \alpha_2 \Delta v(k+1) + \alpha_3 i(k+2) + \alpha_4 i(k+1). \tag{3.20}$$

Inserting this into the first row of the state-space system (3.12) and rearranging gives:

$$\begin{aligned}
&\alpha_1 \Delta v(k+2) + \alpha_2 \Delta v(k+1) + \alpha_3 i(k+2) + \alpha_4 i(k+1) = \\
&= a_1(\alpha_1 \Delta v(k+1) + \alpha_2 \Delta v(k) + \alpha_3 i(k+1) + \alpha_4 i(k)) + b_1 i(k)
\end{aligned} \tag{3.21}$$

$$\begin{aligned}
\Delta v(k+2) = &\frac{a_1\alpha_1 - \alpha_2}{\alpha_1} \Delta v(k+1) + \frac{a_1\alpha_2}{\alpha_1} \Delta v(k) + \frac{-\alpha_3}{\alpha_1} i(k+2) + \\
&+ \frac{a_1\alpha_3 - \alpha_4)}{\alpha_1} i(k+1) + \frac{a_1 * \alpha_4 + b_1}{\alpha_1} i(k) \Rightarrow
\end{aligned} \tag{3.22}$$

$$\Rightarrow \Delta v(k+2) = \begin{bmatrix} \Delta v(k+1) & \Delta v(k) & i(k+2) & i(k+1) & i(k) \end{bmatrix} \underbrace{\begin{bmatrix} \frac{a_1\alpha_1 - \alpha_2}{\alpha_1} \\ \frac{a_1\alpha_2}{\alpha_1} \\ -\frac{\alpha_3}{\alpha_1} \\ \frac{a_1\alpha_3 - \alpha_4)}{\alpha_1} \\ \frac{a_1\alpha_4 + b_1}{\alpha_1} \end{bmatrix}}_{\theta} \tag{3.23}$$

Similarly to the 1st order equivalent circuit, this gives a least squares problem which can be solved using MATLAB's "\" operator to get θ. After inserting the previously substituted α-variables, this leaves a system of five equations which must now be solved in order to get the variables R_0, a_1, b_1, a_2 and b_2.

$$
\begin{aligned}
\theta_1 &= \frac{a_1 \frac{-1}{a_2-a_1} - (1 + \frac{a_1}{a_2-a_1})}{\frac{-1}{a_2-a_1}} \\
\theta_2 &= a_1 \frac{1 + \frac{a_1}{a_2-a_1}}{\frac{-1}{a_2-a_1}} \\
\theta_3 &= -\frac{\frac{R_0}{a_2-a_1}}{\frac{-1}{a_2-a_1}} \\
\theta_4 &= \frac{a_1 \frac{R_0}{a_2-a_1} - \frac{b_1+b_2-R_0 a_1}{a_2-a_1} + R_0}{\frac{-1}{a_2-a_1}} \\
\theta_5 &= \frac{a_1 (\frac{(b_1+b_2-R_0 a_1}{a_2-a_1} - R_0) + b_1}{\frac{-1}{a_2-a_1}}
\end{aligned}
\tag{3.24}
$$

This equation system can be solved with the MATLAB command `vpasolve`, yeilding two solutions which is expected, as the two RC-circuits are interchangeable.

3.5.3 2nd order equivalent circuit with fixed parameters of one RC-pair

In some cases it may be desirable to test the parameter estimation with one of the RC-pairs set manually. The ECM is then mathematically constructed similarly to in Section 3.5.2. However, since one RC-pair is now known, v_2 is also known. The equation system is then solved by simply defining

$$\Delta v(k) = v(k) - v_2(k) - v_{oc}(k) \tag{3.25}$$

3.6 ECM parameter estimation

This section describes the parameter estimation process for the different types of ECM that were used in the project. Furthermore an explanation on how the dependencies on SoC and current were added to the ECM is given.

3.6.1 Cell voltage ECM

The first set of ECM-parameters was calculated based on the equations of a 1st order ECM as previously described in Section 3.5.1, and the required inputs to calculate the parameters R_0, R_1 and C_1 from the ECM are pre-measured voltage and current data. The parameter estimation was repeated at different SoC, as well as with different charge currents, due to the dependencies on these variables, which will be described further in Section 3.6.3 and 3.6.4. The complete fast-charging algorithm developed, based on the existing `pybamm.Experiment()` function, is described by Algorithm 1.

Algorithm 1 Create current protocol

SoC_list = [0, 5, 10, ..., 100]
C_list = [0.1, 0.5, 1, 2, 3, 4]

for $C\_value \in C\_list$ **do**
  for $SoC \in SoC\_list$ **do**
    [Charge at C_valueC for 60 seconds
    Rest for 10 minutes
    Discharge at $\frac{C\_value}{2}C$
    Rest for 10 minutes]
  end for
end for

By simulating the LG M50 battery cell using the current protocol from Algorithm 1 and a sampling period of one second, the corresponding voltage protocol was obtained as an output from the simulation.

As shown in Equation (3.25), another input is $\Delta v(k)$ which is the difference between the cell voltage and the OCV. The OCV was calculated, similar to the Algorithm 1 and as described earlier in Section 3.4, by charging to one of the SoC points (0%, 5%,..., 100%), resting for 10 minutes and repeating the same for the next SoC point. The resulting OCV data points were then interpolated, creating a continuous OCV curve on the interval 0-100% SoC. Both current and voltage data as well as OCV and $\Delta v(k)$ for the LG M50 cell are displayed in Figure 3.2.

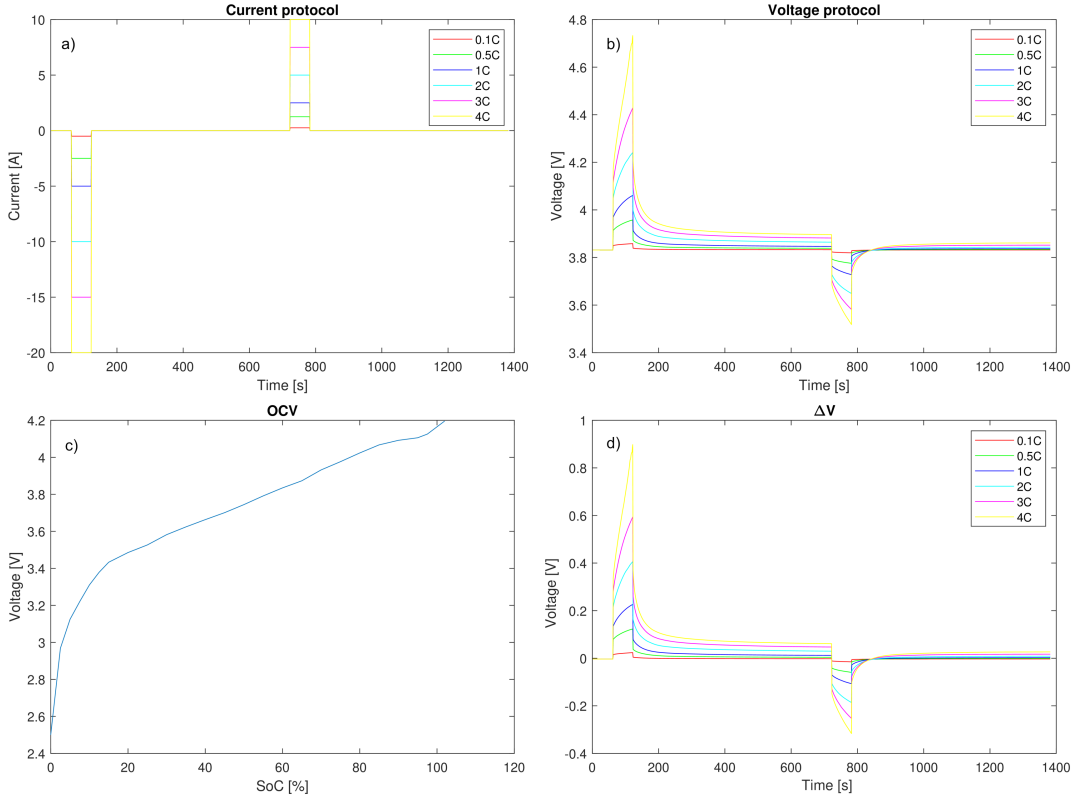


Figure 3.2: Inputs required to calculate ECM parameters. Plots a), b) and d) shows examples at 60% SoC

Using these inputs and following the equations of the 1st order ECM, the parameters describing the internal dynamics of the battery were calculated.

After having estimated the ECM-parameters discretely for intervals of 5% SoC, a polynomial fitting of the estimation points were done in order to make the parameters continuous over the interval 0-100% SoC. A more detailed description of the polynomial adaptions is given in Section 3.6.3.

In the next case a 2nd order ECM was instead used, changing the properties of the previous ECM due to the additional RC-block. This required the parameter estimation to be redone following the relatively more complex although similar procedure described in Section 3.5.2.

Initially, the polynomial adaptation of the parameters was done with regards to keeping the deviation between the polynomials and the estimation points low. This became difficult with the 2nd order ECM, as the parameter estimation gave a very high variance for the R_1 and C_1 pair, as well as for the R_2 and C_2 pair, over the SoC-scale. Therefore, it was attempted to through visual analysis set the 2nd pair's parameters to fixed values over the SoC-scale, and perform parameter estimation on the remaining parameters (R_0, R_1 and C_1). How this was done is described in Section 3.5.3. This resulted in better polynomials with reduced variance, which can be seen in Section 4.1.

3.6.2 Negative electrode overpotential ECM

While analyzing the estimated voltage, based on both 1st and 2nd order ECM parameters, and comparing it to simulations using PyBaMM's FPM description, similarities in shape between the estimated voltage and PyBaMM's calculated NEOP was found, see Figure 3.3. This lead to the idea of recalculating the same ECM-parameters but based on NEOP data instead, in order to find an ECM-based model which outputs an estimation of NEOP.

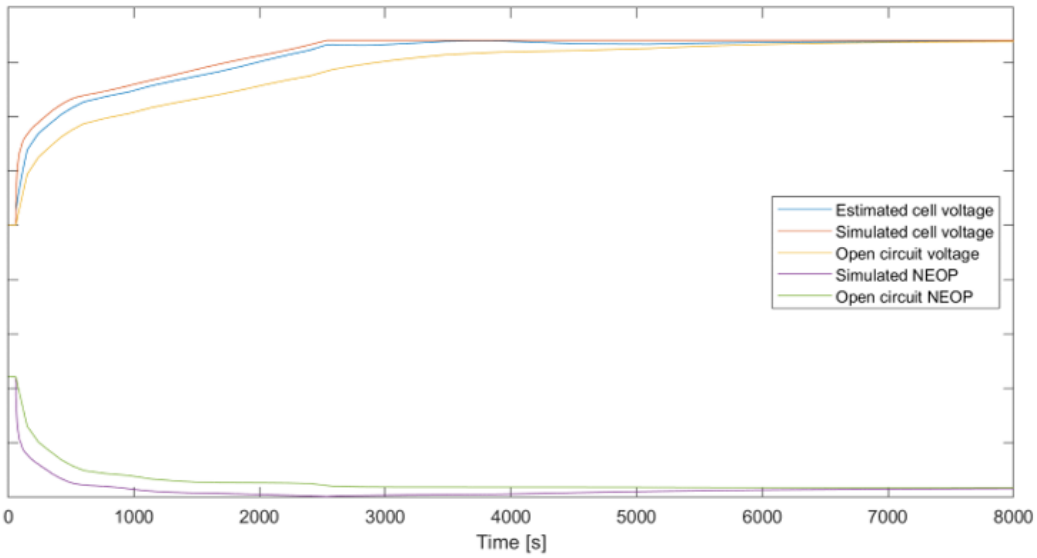


Figure 3.3: Comparison of ECM-based cell voltage and PyBaMM's NEOP during a charging cycle

In order to do this, the equations of the 1st and 2nd order ECM had to be slightly adjusted, yielding the following two equations for the overpotential:

$$\Delta OP(k+1)_{1st} = \begin{bmatrix} \Delta OP(k) & i(k+1) & i(k) \end{bmatrix} \underbrace{\begin{bmatrix} a \\ R_0 \\ R_1(1-a) - a * R_0 \end{bmatrix}}_{\theta} \tag{3.26}$$

$$\Delta OP(k+2)_{2nd} = \begin{bmatrix} \Delta OP(k+1) & \Delta OP(k) & i(k+2) & i(k+1) & i(k) \end{bmatrix} \underbrace{\begin{bmatrix} \frac{a_1\alpha_1 - \alpha_2}{\alpha_1} \\ \frac{a_1\alpha_2}{\alpha_1} \\ \frac{-\alpha_3}{\alpha_1} \\ \frac{a_1\alpha_3 - \alpha_4)}{\alpha_1} \\ \frac{a_1 * \alpha_4 + b_1}{\alpha_1} \end{bmatrix}}_{\theta} \tag{3.27}$$

By making these adjustments, the ECM-parameters could then be calculated in the same way as before. However, it is important to understand that this particular relation between the parameters (R_1, C_1, R_2 and C_2) and the NEOP is just an hypothetical attempt of utilizing the logic behind a voltage ECM for an ECM based on NEOP data. Similarly to the voltage ECM case, a polynomial fit was then done for each parameter of the NEOP-ECM, also here keeping R_2 and C_2 constant, once again achieving continuous parameter values in the interval 0-100% SoC. The values of R_2 and C_2 were chosen experimentally by performing simulations and finding approximate values which yeilded a well-performing ECM estimation, in terms of similarity to the output of the FPM.

The parameters, based on NEOP data, could then be used to give an estimate of NEOP over SoC. Validation of the estimated ECM-based NEOP, in terms of tracking accuracy and charging time, was done in PyBaMM, comparing simulation between ECM-based NEOP towards PyBaMM's FPM-based NEOP.

3.6.3 SoC-dependence of parameters

The model parameters and variables are not constant, but depend on a number of parameters. For the purpose of reducing complexity, factors such as temperature and aging are not taken into account. However there is a dependence on state-of-charge as well as on the current.

For R_0, R_1 and C_1, the SoC-dependence could often be modeled with a 4th-order polynomial function, which is the case in the 2nd order equivalent circuit with fixed parameters of one RC-pair. This was done using the MATLAB function `polyfit`. Initially it was planned to use the entire SoC-scale from 0 to 100% with 5% increments. However due to difficulties generating the correct parameter estimates at the extreme points, 0% and 100%, these were excluded.

For R_1 in the first order model, an exponential function of the form $f(z) = p_1 + p_2^{(p_3 z + p_4)}$ was found to give a good fit, as the resistance was very high at low SoCs but then fairly stable during the remainder of the SoC-scale. The fit was done using the MATLAB function `fitnlm`. However there were significant fluctuations in the estimated values over time, making an accurate fit difficult.

C_1 in the first order model had a rough tendency of having lower capacitance at low SoC and gradually increasing with SoC. Therefore, it was modelled with a logarithmic function of the form $f(z) = p_1 \ln(z + p_2)$ using the MATLAB function `fitnlm`.

The second-order ECM had highly varying parameter estimations for different SoCs, making it difficult to find a clearly suitable function type to fit to the estimations. For R_1 and C_2, the estimations followed similar patterns to R_1 and C_1 from the first order ECM respectively, and the same type of functions were therefore used.

For R_2 and C_2 of the second order, linear functions were chosen in the end, due to the high variance.

3.6.4 Charge current dependence of parameters

The parameters also show a significant dependence on the selected charge current. This dependence was however shown to be more linear than the SoC-dependence. The current dependence was implemented by repeating the process of estimating parameters and fitting suitable functions over the SoC-scale, at six different charge currents: 4 C, 3 C, 2 C, 1 C, 0.5 C and 0.1 C. To do this required adjusting the upper voltage cut-off in the parameter set used in PyBaMM, as it would otherwise be hit at high charge current and high SoC. The fitted parameter-functions were sampled over the SoC-scale with 1% increments, after which a 2D interpolation over SoC and charge current could be made for each of the five parameters.

Due to previous findings that the 2nd order ECM with fixed parameters of one RC-pair gave the best results, the current dependence was only implemented for this ECM. As can be seen in Figure 4.5 and 4.6 however, the estimations for the R_2 and C_2 parameters did change when changing the charge current.

3.7 Closed loop control in PyBaMM

As was found out during the course of the project, PyBaMM was not created with battery control theory applications as its primary purpose. It is possible to provide the software with a current function, but this should be predefined when solving the simulation. However, after investigating different possibilities of how the current can be controlled, a way was found to perform changes to the input current after each simulation step, effectively enabling the sought-after closed-loop control.

In order to initialize the simulation, it must however first be solved as a regular simulation, using `Simulation.solve()` with the arguments `t_eval = np.array([0, 1])` and `inputs = {"Current function [A]" : -0.01}`. This means the simulation will be solved for 1 second with a very low charge current.

After being initialized, it is now possible to begin stepping forward the simulation. This was done using a `while`-loop. First, data from the simulation up until this point is retrieved using `Simulation.solution`. This is saved to the variable `sol`. After using this data to either directly get the negative electrode overpotential from the PyBaMM simulation, or by using the above described ECM-based estimation method, the resulting NEOP value is then used together with the latest voltage reading from the simulation to determine which variable should be controlled. This is determined by a rule engine, which will be described in Section 3.8. The chosen PI controller then outputs its selected current, which is then used as input when performing the next simulation step.

The last action performed in each iteration of the `while`-loop is stepping forward the simulation. This is done using `Simulation.step()` with the arguments `dt = 4` (the simulation timestep in seconds), `save = True` (enabling storing all previous timesteps), `starting_solution = sol` (basing the next simulation step on the solution retrieved in the same `while`-loop iteration) and finally `inputs = {"Current`

`function [A]" : I}`, which sets the current for the next simulation step to the output of the selected PI controller. Simulations using a different sample frequency were also performed. However, after analysis it was concluded that a sample time of 4 seconds was a good middle ground between having sufficient data point simulation accuracy and a reasonable simulation time.

The simulation steps are then repeated as a `while`-loop until its running criteria is no longer met. The running criteria used in the project was `SoC < 95`. After the `while`-loop has finished, the simulation data is saved in a `.mat` file to be used for plotting and analysis in MATLAB.

3.7.1 PyBaMM settings and parameters used

The model used throughout the project is `pybamm.lithium_ion.DFN` with the options listed in Table 3.1.

Table 3.1: DFN model options

Option	Value
`"lithium_plating"`	`"irreversible"`
`"thermal"`	`"isothermal"`
`"operating_mode"`	`"current"`

The parameter set used for all simulations in the project is `Chen2020_plating` (described further in Section 2.5, with the modifications listed in Table 3.2.

Table 3.2: Modifications to `Chen2020_plating` parameter set

Model parameter	Value
`Initial temperature [K]`	293.15
`Ambient temperature [K]`	293.15
`Reference temperature [K]`	293.15
`Upper voltage cut-off [V]`	4.5
`Current function [A]`	`"[input]"`

The simulations are also solved with `initial_soc=0` as an argument.

3.8 Rule engine for control switching

Aside from avoiding too low NEOP when charging li-ion batteries, it is also important to avoid overcharging. This is typically done by transitioning over to a constant-voltage phase once the cell voltage reaches a set voltage threshold, normally 4.2 V for li-ion batteries. An explanation and illustration of charging algorithms using this method, such as CC-CV and MCC-CV, can be found in Section 2.3.

A similar method was also implemented in the proposed algorithm, by adding a separate PI-controller for voltage. This controller should only be active when the

voltage is close to the set maximum cell voltage, in other cases, using the NEOP controller is preferred. To switch between the two, the following logic is evaluated for each new simulation step:

Algorithm 2 Rule engine for control switching

if $OP\_value(k) < (OP\_ref - 0.005)$ **then**
  control_new = "OP"
else if $V\_value(k) > (V\_ref + 0.01)$ **then**
  control_new = "V"
else
  if $OP\_value(k-1) > OP\_ref$ & $OP\_value(k) < OP\_ref$ **then**
    control_new = "OP"
  else if $V\_value(k-1) < V\_ref$ & $V\_value(k) > V\_ref$ **then**
    control_new = "V"
  end if
end if

where k denotes the most recent value of V/OP.

During testing of the rule engine, it was found necessary to add some margin to prevent the rule engine from switching controller during small fluctuations that will be present when either of the controllers are controlling the voltage/NEOP. The margin was set to 0.005 V for the NEOP and 0.01 V for the cell voltage.

The variable `control_new` determines which controller is in use, how the controllers themselves are designed as well as the switching is described below.

3.9 PI controller

As can be seen in Figure 3.1, two PI controllers are used. However, despite controlling two different state variables (voltage and NEOP), they are very similar. Based on the control variable set by the rule engine, the calculation of the integral state differs depending on whether this changed since the last iteration, by comparing the current control variable (`control_new`) with the previous control variable (`control_old`), as shown below:

```
if control_new == "V":
    if control_new == control_old:
        q_new = V_Ki*max(min(V_err,0.1),-0.1) + q_old
    else:
        q_new = I[-1] - V_Kp*V_err
    I = max(Imax, V_Kp*V_err + q_new)
elif control_new == "OP":
    if control_new == control_old:
        q_new = OP_Ki*max(min(OP_err,0.01),-0.2) + q_old
    else:
```

$$\text{q\_new} = \text{I}[-1] - \text{OP\_Kp} * \text{OP\_err}$$
$$\text{I} = \max(\text{Imax}, \text{OP\_Kp} * \text{OP\_err} + \text{q\_new})$$

Algorithm 3 Rule engine for control switching

```
if control_new == "V" then
  if control_new == control_old then
    q_new = V_Ki*max(min(V_err,0.1),-0.1) + q_old
  else
    q_new = I[k] - V_Kp * V_err
  end ifI = max(Imax, V_Kp*V_err + q_new)
else if control_new == "OP" then
end if
```

To limit the integral wind-up and consequent overshoot that occurs especially during the initial phase of charging, the integral state is constrained by bounding the accumulation.

The proportional and integral gains were experimentally tuned and were for both controllers set to 10 and 8, respectively.

4

Results

This section of the report presents the results obtained from the previously described methodology involving simulations done in Matlab and PyBaMM. This includes the ECM-parameters and the performance of the charging algorithms for the initial MCC-CV-algorithm, the NEOP based charging from PyBaMM and the algorithm regulated based on the estimated NEOP from the ECM-parameters.

4.1 Estimated parameters

The resulting SoC dependent parameter-polynomials for the final ECM, 2nd order based on NEOP, are displayed in Figure 4.1. The figure illustrate the estimations points for every 5% SoC, the polynomial fit and how these change with current. The accuracy of the polynomials were validated by calculating the mean square error (MSE) between the estimation points and the polynomials, see Table 4.1.

This is followed by the 2D-interpolation of each of the parameters of the ECM in Figure 4.2 - 4.6.

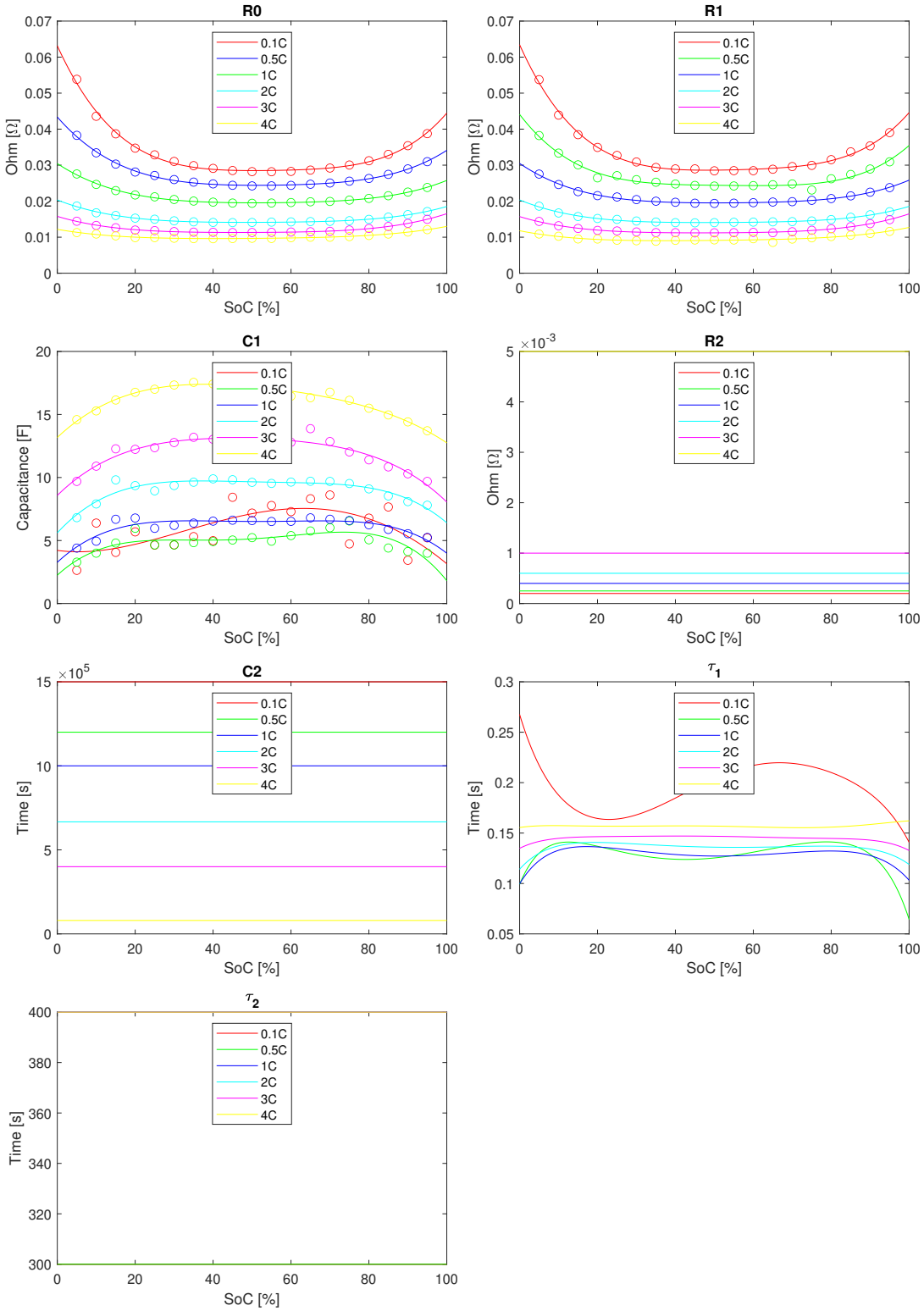


Figure 4.1: Polynomial-adapted 2nd order NEOP-ECM parameters

Table 4.1: MSE of ECM-polynomial fitting to data points, related to Figure 4.1

.

	0.1 C	0.5 C	1 C	2 C	3 C	4 C
R_0 [Ω]	2.96E-6	9.42E-7	3.76E-7	1.88E-7	2.39E-7	1.87E-7
R_1 [Ω]	2.84E-6	1.85E-6	4.10E-7	2.01E-7	2.48E-7	2.01E-7
C_1 [F]	1.6992	0.4206	0.1749	0.2134	0.3188	0.2134

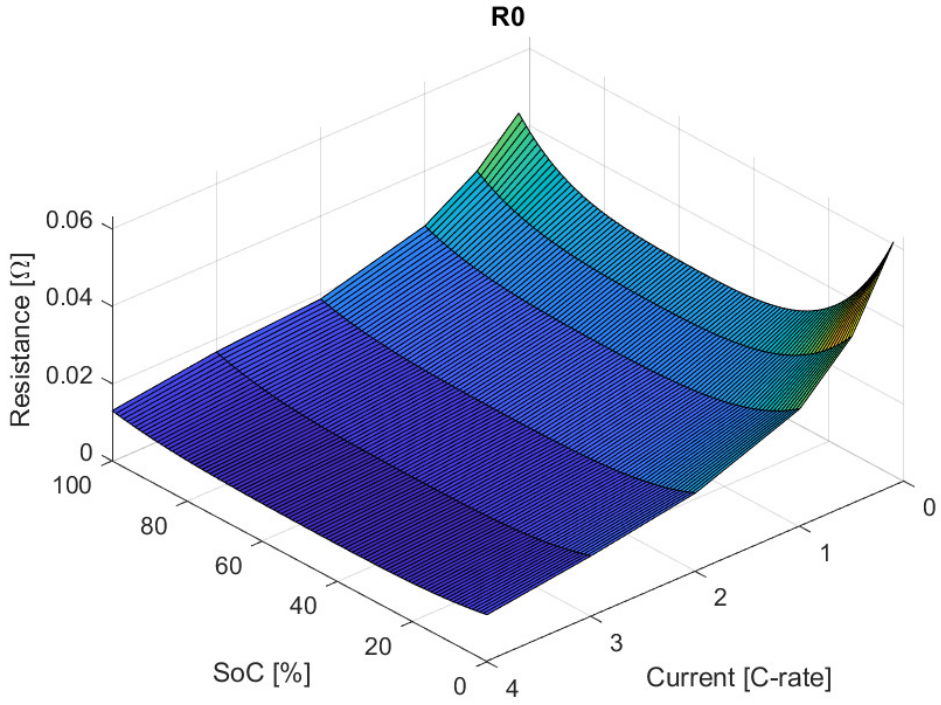


Figure 4.2: Interpolated R_0 (2nd order NEOP-ECM)

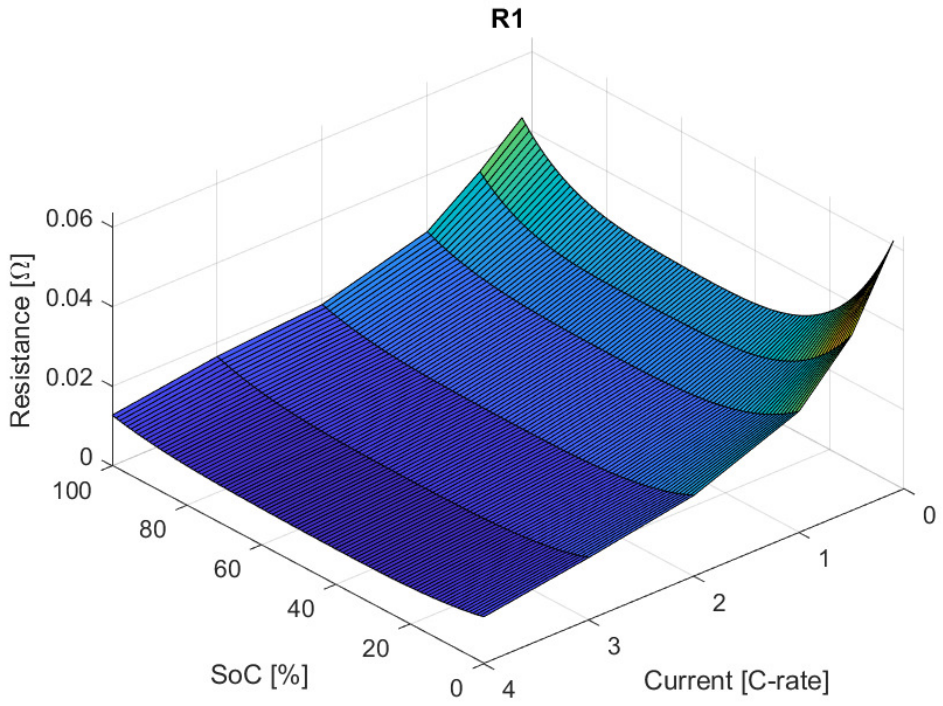


Figure 4.3: Interpolated R_1 (2nd order NEOP-ECM)

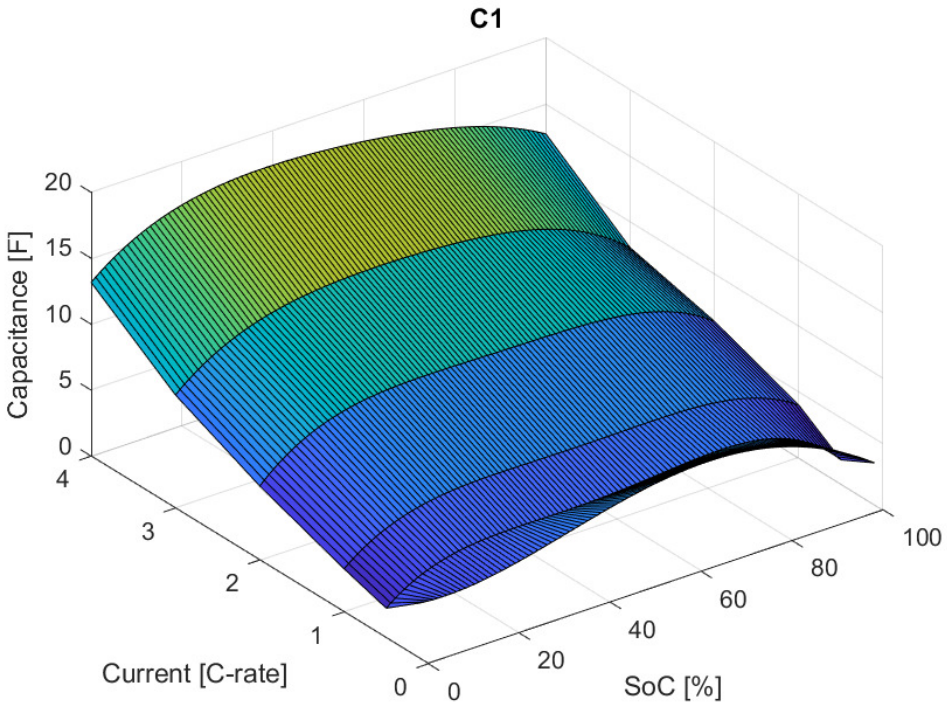


Figure 4.4: Interpolated C_1 (2nd order NEOP-ECM)

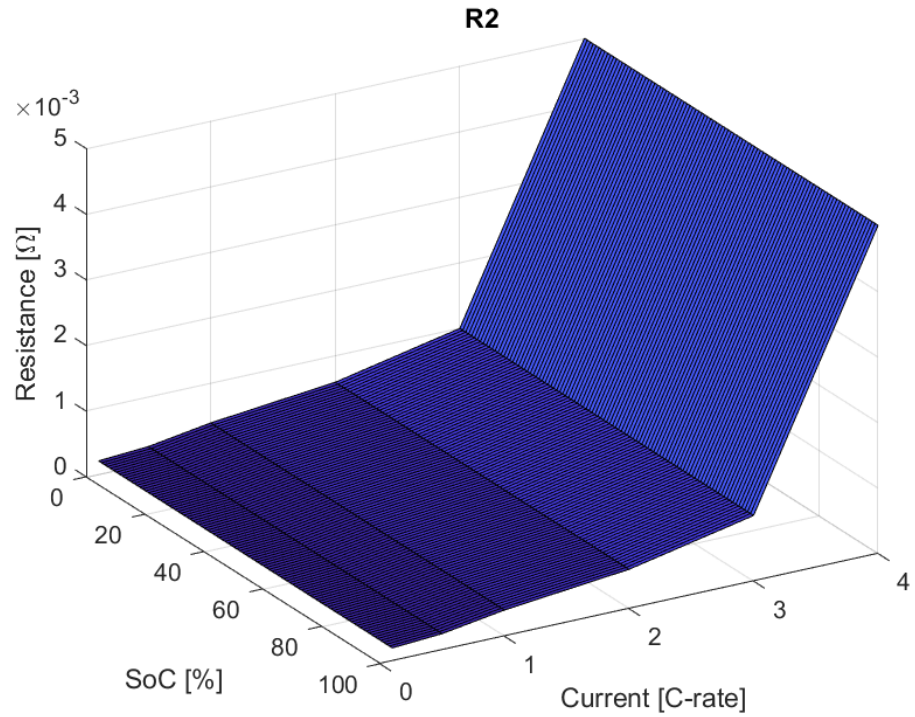


Figure 4.5: Interpolated R_2 (2nd order NEOP-ECM)

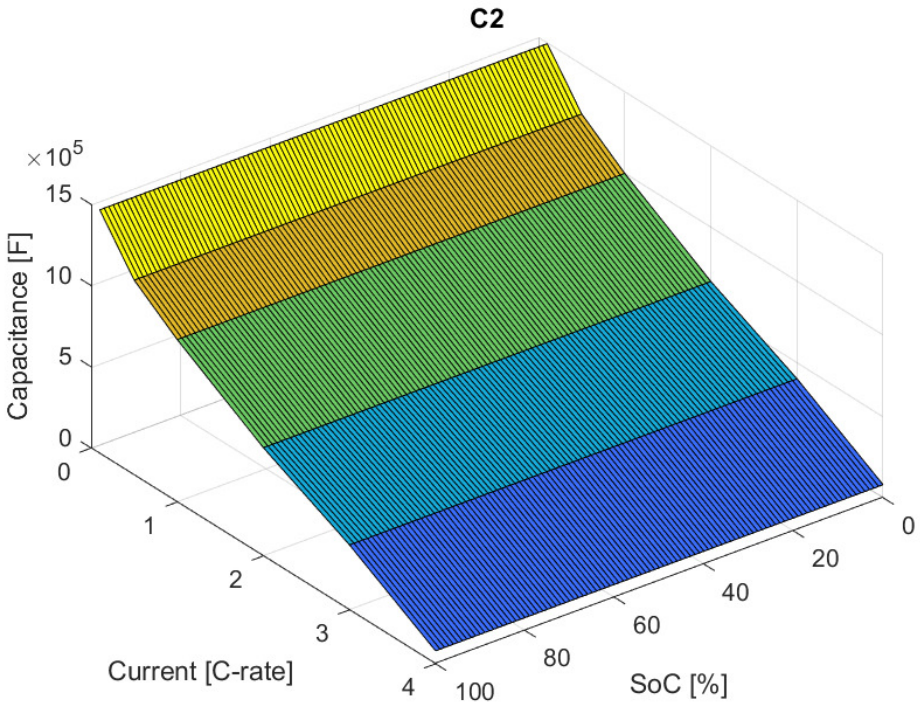


Figure 4.6: Interpolated C_2 (2nd order NEOP-ECM)

4.2 MCC-CV protocol (reference algorithm, marginless)

Using data from PyBaMM, it is possible to find the cell voltage corresponding to when the negative electrode overpotential reaches 0 V for a given charge rate. In this marginless simulation, this relation is assumed to be accurate, so the step-down is triggered at the voltage level that is reached when the negative electrode overpotential is 0 V.

The selected constant current-levels used in the MCC-part of the algorithm are 3 C, 2 C, 1.5 C and 1 C, after which the algorithm can safely be allowed to reach the maximum voltage of 4.2 V (i.e. without the negative electrode overpotential becoming negative) and switch over to CV charging. The current profile of the algorithm can be seen in Figure 4.7.

The charging was started at 0% SoC and reached 30, 50, 70 and 90% SoC in the durations shown in Table 4.2.

Table 4.2: Charging time for the marginless MCC-CV charging algorithm

State-of-Charge	Time (hh:mm:ss)
30 %	0:06:33
50 %	0:14:53
70 %	0:27:01
90 %	0:50:29

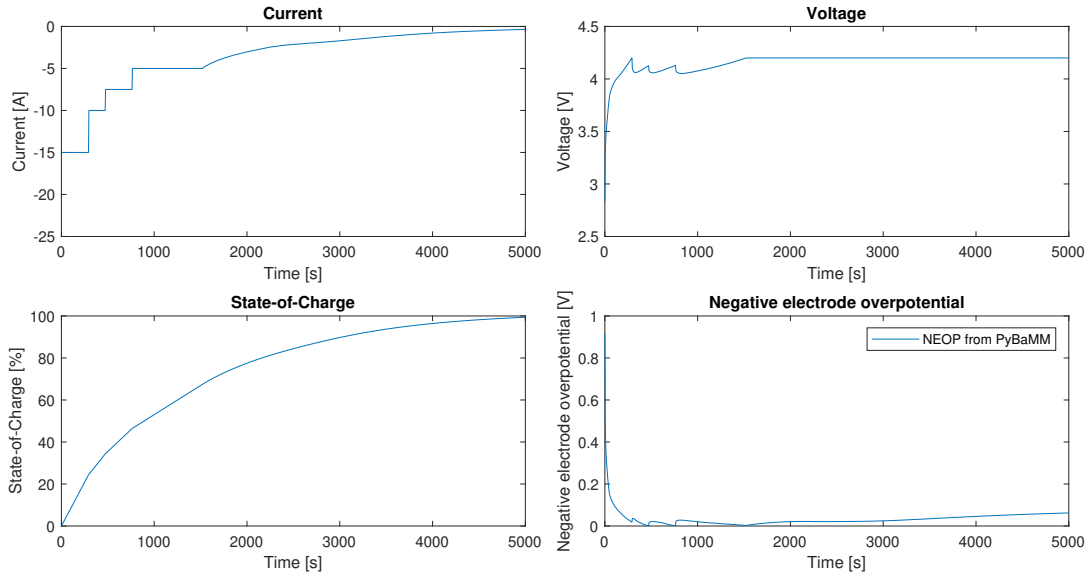


Figure 4.7: Charge curve and variables of the MCC-CV charging algorithm (marginless)

4.3 MCC-CV protocol (reference algorithm, with margin)

In reality, it is not possible to assume that an algorithm with no margin to the negative electrode overpotential becoming negative will be able to prevent plating from happening. This is due to the capacitive effects seen in the characteristics of the negative electrode overpotential, causing the resulting potential to depend on the chargerate, initial-SoC and several other factors. To account for this and to provide more realistic results, another algorithm designed to keep the negative electrode overpotential above 0.025 V instead of 0 V was developed. In addition to the previous current levels 3 C, 2 C, 1.5 C and 1 C, it was also needed to introduce an additional constant-current stage of 0.5 C before the maximum voltage, 4.2 V, could safely be reached.

The charging process can be seen in Figure 4.8, with the performance summarized in Table 4.3.

Table 4.3: Charging time for the MCC-CV charging algorithm with margin

State-of-Charge	Time (hh:mm:ss)
30 %	0:07:12
50 %	0:18:37
70 %	0:42:37
90 %	1:19:45

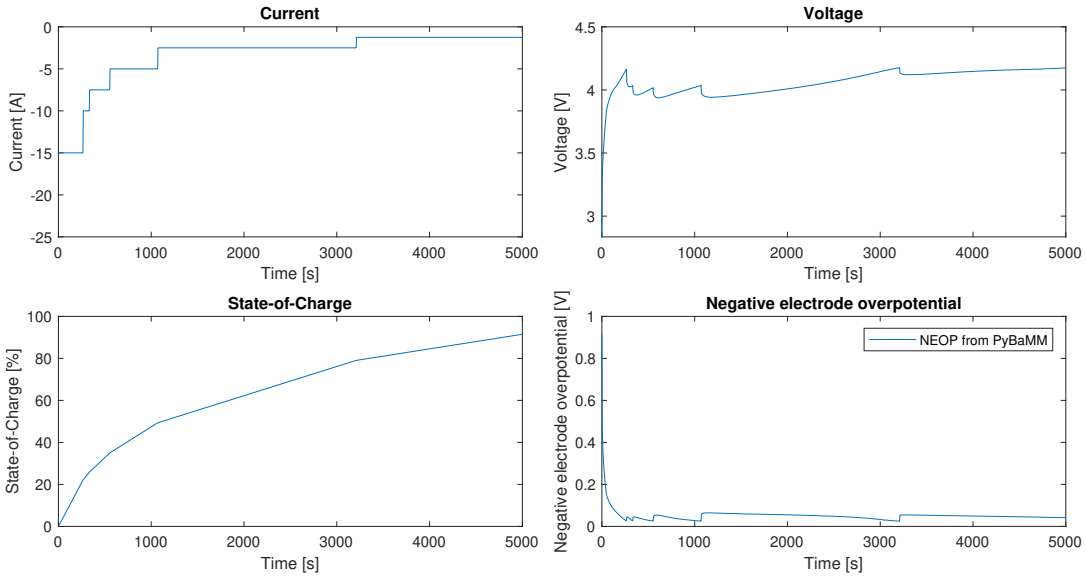


Figure 4.8: Charge curve and variables of the MCC-CV charging algorithm (with margin)

4.4 PI-controlled charging based on PyBaMM overpotential data

It was also attempted to control the current based on the overpotential returned from the FPM in PyBaMM. As this variable is considered the "true" solution, this will return a very close to ideal-case lithium plating-aware control algorithm. In addition, this test scenario can be used to benchmark the PI control system.

This charging algorithm was evaluated in two cases; first, with the charging current limited to 3 C (15 A), and the starting current set to 15 A as well, as shown in Figure 4.9. The results from this test cycle is therefore suitable to compare with the MCC-CV algorithm described above. In the second case, the starting current was set to 13 A (2.6 C) to avoid the simulation becoming unsolvable and hence producing errors. The errors encountered are likely due to other internal parameters reaching unrealistic values. However this issue was not investigated in depth. With the selected starting current, the algorithm ran without limited charging current, reaching a maximum current of 25.59 A. This simulation can be seen in Figure 4.10.

The percentual difference in Table 4.4 is the comparison with the ideal MCC-CV simulation.

Table 4.4: Charging time for the PyBaMM-based PI charging algorithm

State-of-Charge	Time (hh:mm:ss): Max 3 C	Time (hh:mm:ss): ∞
30 %	0:06:05 (-7.1%)	0:05:17 (-19.3%)
50 %	0:12:24 (-16.7%)	0:11:44 (-21.2%)
70 %	0:23:20 (-13.6%)	0:22:52 (-15.4%)
90 %	0:47:00 (-6.9%)	0:46:44 (-7.4%)

4.5 PI-controlled charging based on overpotential from the ECM (marginless)

As the aim of the project was to investigate whether a more sophisticated estimation method/model can improve the charging time compared to conventional charging algorithms such as the MCC-CV protocol (described in Section 2.3 with results shown in Section 4.2), while being more computationally efficient than using an FPM such as the DFN model (described in Section 2.4 with results shown in Section 4.4), it was also attempted to control the charging current based on the output of the ECM model of the negative electrode overpotential.

Like with the reference MCC-CV algorithm developed, this algorithm was also evaluated both when the controller attempts to control the negative electrode overpotential to 0 V (a marginless, more idealistic case), as well as to a 0.025 V margin, in order to provide a more realistic result. The results from the controller set to 0.025 V are presented in Section 4.6.

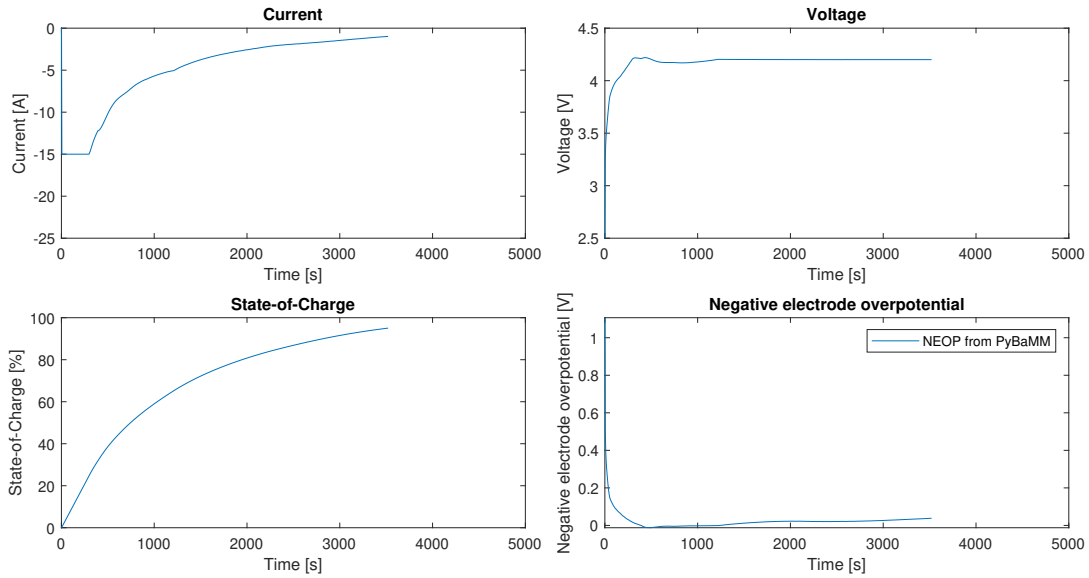


Figure 4.9: Charge curve and variables of the PyBaMM-based PI-controlled charging algorithm (max current = 3 C, no margin)

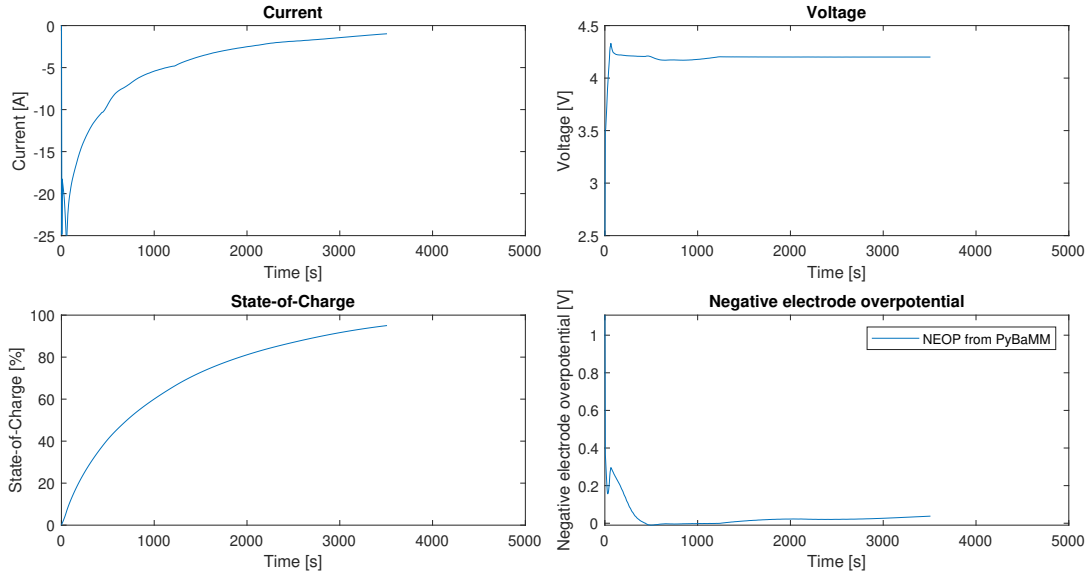


Figure 4.10: Charge curve and variables of the PyBaMM-based PI-controlled charging algorithm (unlimited current, no margin)

Despite controlling the NEOP with a reference of 0 V in this section, a margin can still be observed in the later stages of the charging sessions. This is due to switching controller to the cell voltage controller, as described in Section 3.8.

Like with all PI-control algorithms developed, the simulations were run both with a limitation to max 3 C charging current, as well as with unlimited current. The simulations are shown in Figure 4.11 and 4.12 respectively. While in the limited case, it was possible to start the charging session at 3 C charging current, this had to be slightly lower in the unlimited case to avoid the numerical issues, and was set to 9 A (1.8 C). The unlimited algorithm reached a maximum charging current of 23.45 A.

Table 4.5: Charging time for the ECM-based PI charging algorithm (no margin)

State-of-Charge	Time (hh:mm:ss): Max 3 C	Time (hh:mm:ss): ∞
30 %	0:06:05 (-7.1%)	0:05:17 (-19.3%)
50 %	0:13:48 (-7.3%)	0:14:00 (-5.9%)
70 %	0:24:24 (-9.7%)	0:24:44 (-8.5%)
90 %	0:47:52 (-5.2%)	0:48:24 (-4.1%)

The percentual difference in Table 4.5 are in comparison with the corresponding MCC-CV algorithm simulation.

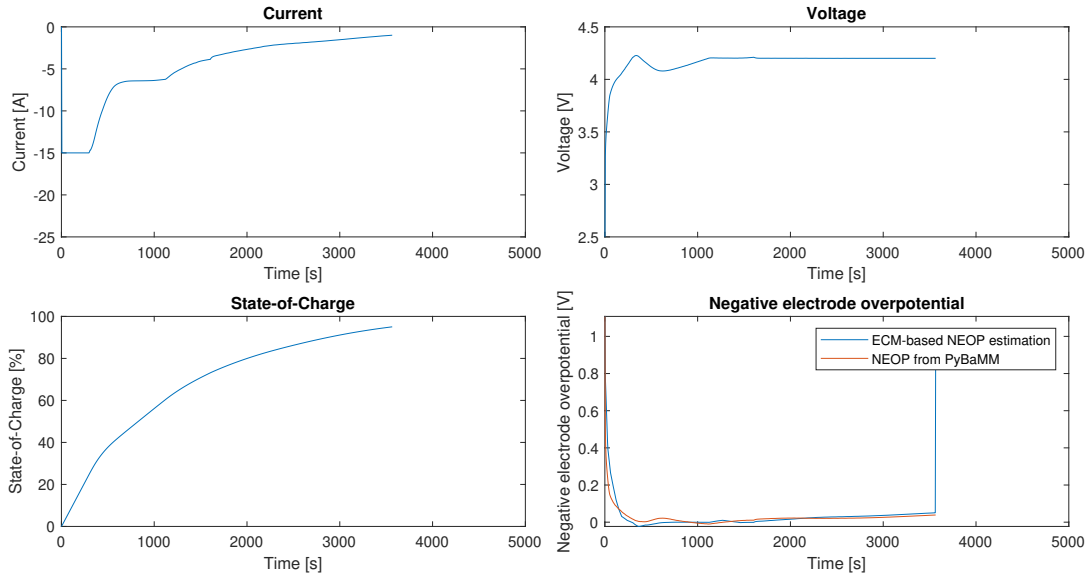


Figure 4.11: Charge curve and variables of the ECM-based PI-controlled charging algorithm (max current = 3 C, no margin)

4.6 PI-controlled charging based on overpotential from the ECM (with margin)

In the case with a margin, the controller attempted to control the negative electrode overpotential to 0.025 V. The starting charging current was in both cases 15 A (3

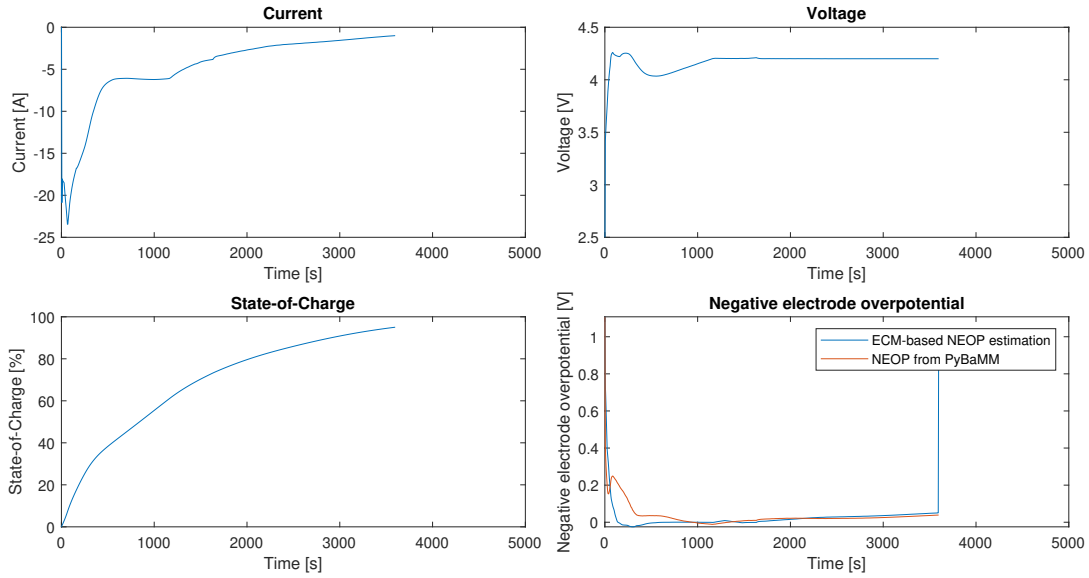


Figure 4.12: Charge curve and variables of the ECM-based PI-controlled charging algorithm (unlimited current, no margin)

C) and the unlimited algorithm then reached a maximum current of 25.90 A. Like with all PI-controlled algorithms, the simulation was run both with a 3 C current limited, as well as with unlimited current. The simulations are shown in Figure 4.13 and 4.14 respectively.

The percentual difference in Table 4.6 is a comparison with the equivalent MCC-CV algorithm simulation.

Table 4.6: Charging time for the ECM-based PI charging algorithm (with margin)

State-of-Charge	**Time (hh:mm:ss): Max 3 C**	**Time (hh:mm:ss): ∞**
30 %	0:06:13 (-13.7%)	0:06:01 (-16.4%)
50 %	0:17:04 (-8.3%)	0:17:36 (-5.5%)
70 %	0:31:16 (-26.6%)	0:31:52 (-25.2%)
90 %	0:57:32 (-27.9%)	0:58:08 (-27.1%)

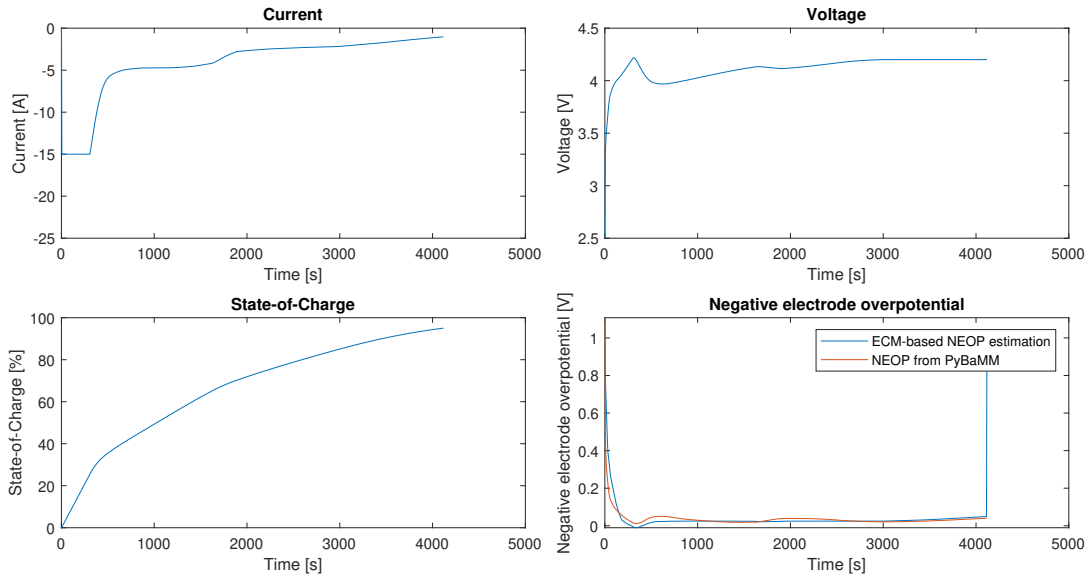


Figure 4.13: Charge curve and variables of the ECM-based PI-controlled charging algorithm (max current = 3 C, with margin)

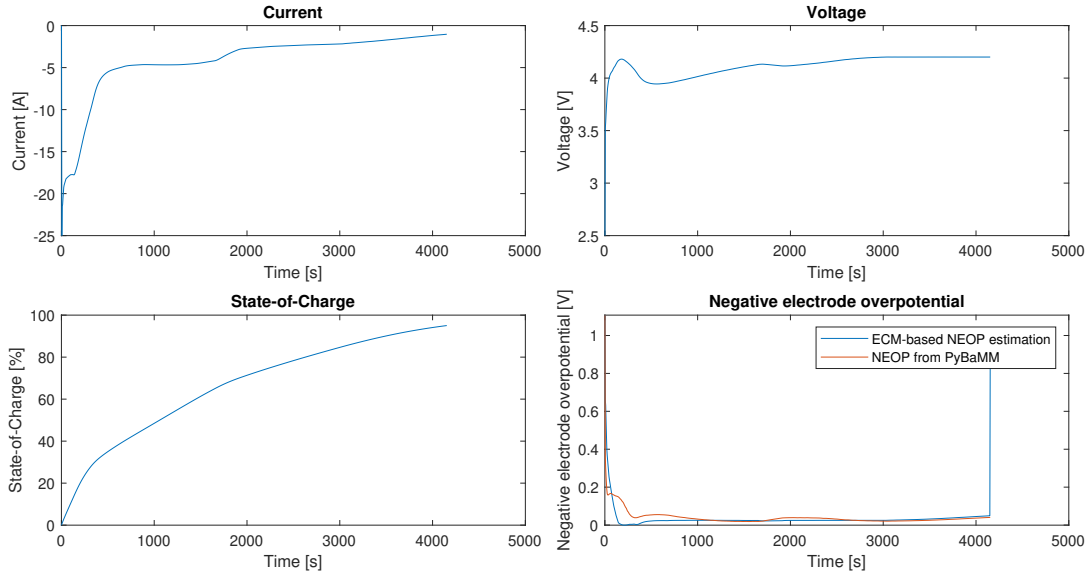


Figure 4.14: Charge curve and variables of the ECM-based PI-controlled charging algorithm (unlimited current, with margin)

5

Discussion

This chapter provides an analysis and discussion of the results presented in Chapter 4, as well as of the project as a whole. This includes the parameter estimation, the various charging algorithms that have been developed and simulated, as well as sustainability and ethical aspects of the work. Furthermore, aspects that have not been part of the project but should be considered for future work on the topic are discussed.

5.1 Parameter estimation

By analyzing the results of the polynomial adapted ECM-parameters and observing the corresponding mean squared error (MSE) for each current-level, it is clear that the chosen polynomials are of high accuracy in all cases with the exception of the parameter C_1 at 0.1 C, which behaves strangely compared to its remaining current-levels. This can be seen by C_1's representation in Figure 4.1 and 4.4. Possible explanations could for example be that some of the LG M50 parameters listed in Appendix A does not accurately represent the cell's properties over the whole current spectrum, and giving them a current dependency could therefore be a useful measure to improve their quality. Moreover, it was stated in the original source article that some of the cell's parameters were taken from a different battery cell, and despite the cells sharing similar properties, this makes the overall representation of the LG M50 cell less reliable. However, since the remaining ECM-parameters follows an expected trajectory of values even at lower current-levels, it is reasonable to think that if poor cell parameters is the underlying issue, it mainly targets the capacitive properties of the battery.

Another aspect to consider is that R_2 and C_2 were chosen manually with regards to identifying and matching the rise time and relaxation time of the estimated NEOP. These chosen values had an effect on the remaining ECM-parameters, confirmed when changing between different fixed values of R_2 and C_2. However, because C_1 is much smaller relative to C_2, its contribution to the complete ECM is too and the overall parameter estimation can therefore be considered fairly accurate even at lower current-levels.

The interpolated versions of the polynomials in Figure 4.2 - 4.6 improved the robustness of the ECM-parameters due to them capturing both SoC and charge de-

pendencies, instead of only considering dependencies on SoC. Another advantage of the interpolation was that the parameters became continuous w.r.t. current, compared to the polynomials which only were made for six different currents. Additional variants of interpolation could be used to include dependencies on for example temperature and SoH. However, this was not investigated.

5.2 Charging algorithms

Three main charging algorithms were developed during the project. The MCC-CV algorithm serves as a reference point to evaluate the performance of other algorithms against. The PyBaMM-based PI-controlled algorithm controls the system based on the "true" negative electrode overpotential, and should therefore provide the ideal performance numbers of the approach. Finally, and most interestingly, is the ECM-based PI-controlled algorithm. This is the main concept developed during the project, and its results are compared with the two other algorithms.

5.2.1 MCC-CV

The MCC-CV algorithm developed produced very good results, charging from 0 to 70% SoC in around 27 minutes in the marginless case, and in less than 43 minutes when the 0.025 V margin was applied. However it should be taken into account that a more realistic implementation of this algorithm would probably have to be even more pessimistic than the case with margin, as factors such as aging and temperature needs to be taken into account and cannot always be accurately measured or estimated. However, this is likely also the case with an ECM-based algorithm. Another issue which will be discussed further in Section 5.3 is the presence of gradients in a complete BEV battery pack, w.r.t. temperature and potentially also w.r.t. SoC and SoH, which may limit the overall charging speed of the battery pack. It should, however, be mentioned that this factor will apply for the other charging algorithms developed in this paper as well.

The constant-current levels of the algorithms were chosen to be 3 C, 2 C, 1.5 C and 1 C, as well as 0.5 C in the case with a 0.025 V margin. The primary reason for not choosing a higher initial charging current is due to the diminishing benefits of charging at higher peak currents, something which was evident from the PI-controlled charging. Another factor which would be relevant in a real-world application would be the infrastructure needed to support very high peak charging currents. With a pack size of 75 kWh, already charging at 3 C requires a supplied DC power of 225 kW, or a current of $\frac{225000}{400\sqrt{3}} = 324.76$ A from a 400 V, 3-phase power grid, not including conversion losses.

In the case with a 0.025 V margin, it was noted that charging at higher SoC was quite slow, for example from 70% to 90% SoC took nearly 40 minutes to complete. This can partly be attributed to the fairly drastic reduction in charging current due to the added 0.5 C CC stage, which reduced the current by 50%. While it was necessary to add another CC stage after the 1 C stage since the negative electrode

overpotential would otherwise become < 0.025 V, a higher current could possibly have been selected for the last stage in order to improve performance.

5.2.2 PyBaMM-based PI-controlled charging

Two cases of PI-controlled charging were investigated, one which did not have a limitation on the charging current, and one where it was kept at maximum 3 C, just as with the MCC-CV algorithm. For the current limited case, it was possible to start the charging at this current level, but in the unlimited case, the starting current had to be set slightly lower (13 A) to avoid numerical problems in the PyBaMM simulation. The cause for these errors is unknown. This could cause a slight time disadvantage compared to the MCC-CV algorithm, but as can be seen in the results in Section 4.4, the PI-controlled charging based on the overpotential data from the FPM in PyBaMM delivers substantially improved charging time regardless of end-SoC.

While a solutions based on a FPM currently cannot be feasibly implemented in a BMS due to its computational complexity, these results show the potential of charging speed improvement when controlling charging based on the negative electrode overpotential, while managing the risk of generating lithium plating. With the potential for controlling the charging current in this way, the challenge is to implement a sufficiently accurate but less complex model for estimating the negative electrode overpotential. Faster charging time could have been achieved if the overpotential was kept controlled around zero. However, this was not possible due to the maximum voltage constraint.

5.2.3 ECM-based PI-controlled charging

During the project, the idea to model the overpotential similarly to how the cell voltage has previously been modelled was found and investigated. As can be seen in Figure 4.11, despite efforts to finding accurate parameters for the model, the model has considerable difficulty in giving an accurate estimate of the negative electrode overpotential, mainly during the start of the charging session and especially when using unlimited current.

The underestimated overpotential causes the difference in charging time between the session limited to a maximum charging current of 3 C, and the session with unlimited current to be practically insignificant. As can be seen in Table 4.5 and 4.6, the charging times to all 4 SoC levels never differed more than 40 seconds, and for the end-SoC levels 50%, 70% and 90%, the 3 C current-limited case was actually faster than the unlimited case. It would seem that the higher maximum current caused a further decrease in the negative electrode overpotential, causing the controller to throttle the charging current more aggresively than in the current-limited case.

Another consequence of the somewhat inaccurate model is that it does not manage to keep the "true" negative electrode overpotential (from PyBaMM) above 0 V

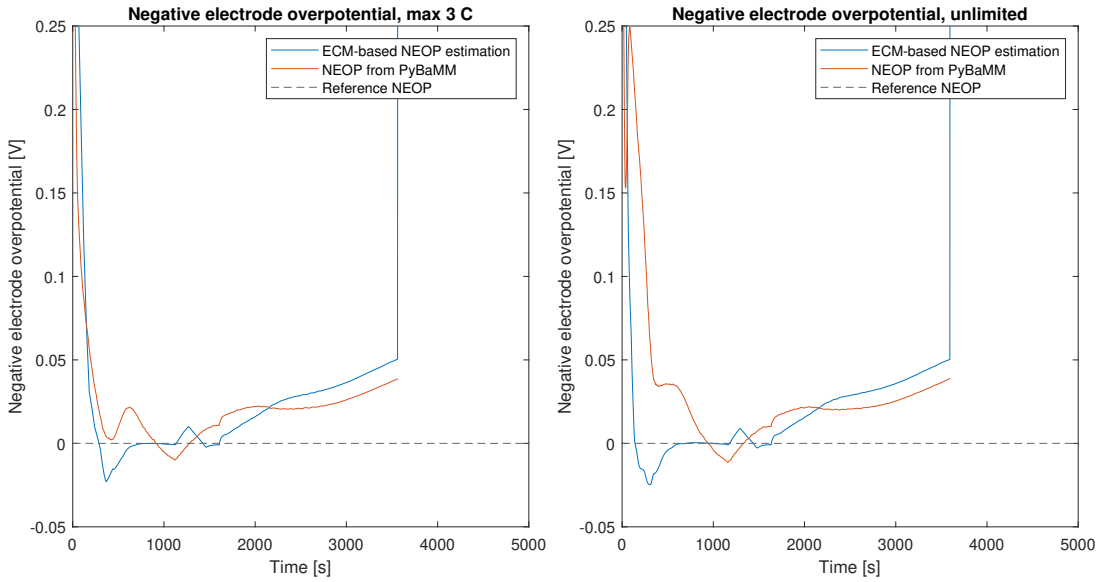


Figure 5.1: Cropped view of the NEOP results presented in Section 4.5

throughout the charging session. This can also be seen in Figure 5.1. While the minimum NEOP reached is only around -0.01 V in both the 3 C and the unlimited case, and only subzero for a fairly short time, this could still pose a risk of lithium plating.

The same consequences can be seen when looking at the NEOP in the simulations with 0.025 V margin. Here, the model manages even worse to maintain the NEOP over the reference value, crossing it 3 times in the 3 C current-limited case, and twice in the unlimited case. However as can be seen in Figure 5.2, the NEOP at least remains completely above 0 V.

This is likely part of the explanation for the large difference in charging time compared to the MCC-CV algorithm seen in Table 4.6. While the MCC-CV algorithm was designed to respect the 0.025 V margin completely, this was not achieved with the ECM-based PI control algorithm. Another cause could be the addition of the 0.5 C CC stage which had to be used in the MCC-CV algorithm, which was discussed in Section 5.2.1.

5.3 Further aspects to be considered

This chapter is a followup to the material discussed in Section 5.1 and 5.2 and highlights potential improvement aspects which could be considered as an extension of the project. This includes more sophisticated methods of modeling the NEOP and external aspects.

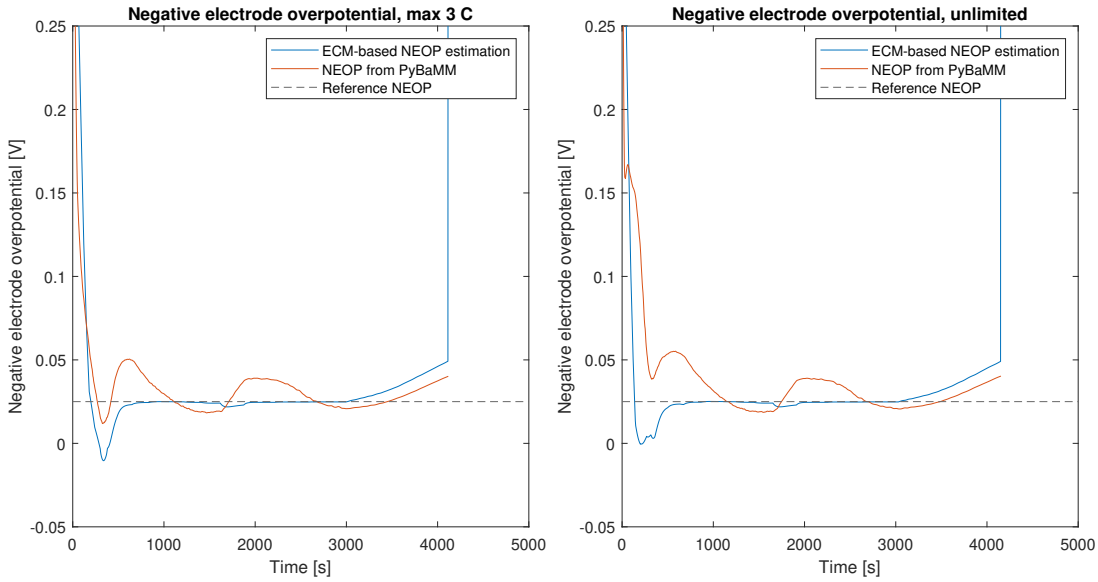


Figure 5.2: Cropped view of the NEOP results presented in Section 4.6

5.3.1 Adaptions to include more factors

The algorithms developed have, as mentioned, been derived and tested in fairly simplified conditions. Using a constant temperature, no aging on the battery cell and only considering charging of a single cell instead of a complete battery pack has been suitable for the purpose of showing the possibility of using ECM's to model the negative electrode overpotential, but the problem must be expanded to take these factors into consideration before it can be used in a realistically implementable charging algorithm.

This can be handled in a number of ways. One, which could be considered more of a "brute-force" approach, is to by simulating and testing in different conditions generate various lookup tables which can be used by the system, similarly to how the parameter dependence on SoC was modelled. However, variations in the environments the battery is used in may cause differences in how the battery degrades over time, for example, which could affect the accuracy of the parameters.

Another approach is to investigate possible relation between the parameters of the overpotential ECM and the cell voltage ECM. Since there are existing methods for performing real-time parameter estimation of the cell voltage ECM, such as in [17] and [18], if an accurate enough relation is found, real-time estimation of the cell voltage ECM parameters can then be used for creating the negative electrode overpotential ECM model online in the BMS.

5.3.2 Alternative NEOP estimation methods

As seen in Figure 3.1, the equivalent circuit model is not making use of certain available variables, such as cell voltage. If this was the case, the accuracy of the model could potentially be improved. However this would likely require a different

type of model than an ECM of the negative electrode overpotential.

One approach, which has not been investigated in this project, is to make use of a more conventional cell voltage ECM, and use its outputs as well as other available variables in an estimation block, which may be found using system identification. A system overview of such a system can be seen in Figure 5.3. The estimation block could possibly be developed using MATLAB's System Identification Toolbox.

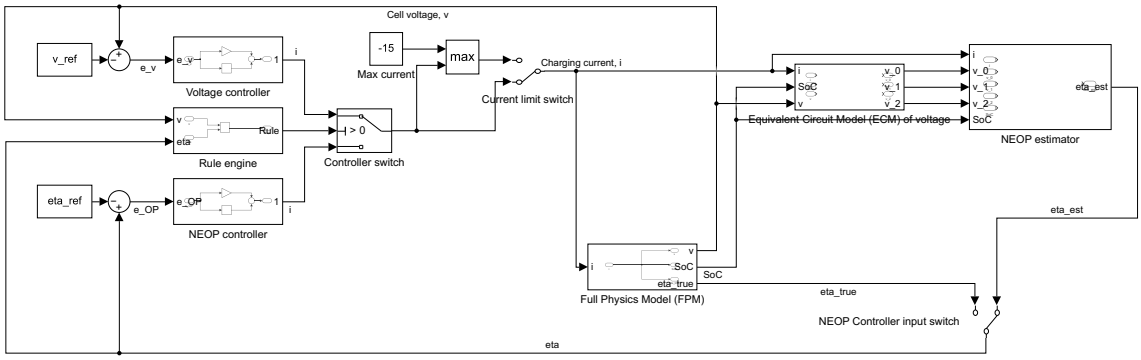


Figure 5.3: Overview of the system with alternative NEOP estimation

5.4 Sustainability and ethical aspects

From an environmental perspective there are major benefits of a transition towards BEVs for new vehicle production. Despite the battery manufacturing process of BEVs involving emissions of greenhouse gases, by having a very high drivetrain efficiency, it will over a car's lifecycle cause less climate footprint compared to combustion engine cars. There is, however, an important consideration that the production of the materials used in the batteries comes from methods which do not violate any basic human rights.

5.4.1 Sustainability aspects

During the last decades, it has become clear that climate change in the form of global warming can be correlated to the cumulative amount of CO_2 in the atmosphere. Without measures to reduce global greenhouse gas emissions, the impact of global warming will be severe [19].

The transport sector contributes significantly to global greenhouse gas emissions (16.2%) [20]. Decomposing this category further into transport sectors, shows that road transport is the by far largest polluter in the transport sector, accounting for 11.2 % of global greenhouse gas emissions. Passenger vehicles are the largest subcategory in road transportation, accounting for roughly 60% of the road transport emissions [21], or around 6.7% of all global greenhouse gas emissions combined. This includes passenger cars, but also buses and motorcycles. Electric vehicles have been proposed as a more environmentally friendly option to traditional internal combustion vehicles, due to their high energy efficiency. Additionally, the operating

emissions of an electric vehicle exclusively depend on the energy source used to produce the energy. If electric vehicles are used together with carbon-free electricity sources, this has the potential of drastically reducing the road transport emissions [20].

Another aspect that must be considered is the environmental impact on manufacturing different types of vehicles. It has been shown that due to the large battery packs needed for BEV's, the manufacturing of these vehicles emits more greenhouse gases than an equivalent internal combustion engine car [22]. However, owing to the high efficiency of the BEV, it will over a car's lifetime still yeild a significant reduction of emissions (36%) seen over the entire lifecycle [22]. In the cited article a 150 000 km lifespan were assumed using an average European electricity mix.

While improving charging time does not directly affect the energy efficiency of a BEV, it does allow for more energy to be charged during a given time, allowing the driver to drive further until the next charging stop, or shorten the time of the charging stop. This addresses two key concerns among prospective BEV buyers, namely charging time and range anxiety [4]. Alternatively, the manufacturer can reduce the cost of the vehicle by using a smaller battery pack and in turn reduce the vehicle's manufacturing climate footprint.

Whichever way the potential improvements would be implemented, it improves the general competitiveness of the electric vehicle. This could in turn increase the market share of BEVs, which would be beneficial from a global emissions standpoint, as described above, but also on a local level, as BEVs does not emit any particles from the drivetrain and are also often very quiet, which can reduce noise pollution in cities.

5.4.2 Ethical aspects

A common point of ethical discussion regarding battery-electric vehicles is the manufacturing of battery cells. There have been examples where for example the sourcing of cobalt, a material commonly used in the cathode of a Li-ion battery, has not been done in a way compliant with basic human rights. While ensuring that the materials used in an electric vehicle is sourced in an ethical way remains an important topic, this project has not investigated the choice of cell chemistry.

The production of batteries does however remain energy-intensive, and it is therefore important that the battery cells are not being used in a way which accelerates degrading mechanisms, as this may result in a need for a battery pack replacement before the end of the vehicle's lifetime, which would cause more batteries to be needed. Furthermore, if battery life exceeds the vehicle's lifetime, the battery pack can be re-purposed to for example stationary energy storage, without need for additional production of battery cells.

6

Conclusion

This master's thesis has attemped to improve fast charging in electric vehicles. The limiting factors of charge rate are various degradation mechanisms, which, if not considered during charging, can cause the battery cells to degrade at a faster rate than otherwise. The focus in this project has been on lithium plating, and trying to improve the charge rate through modelling of when irreversible lithium plating might occur.

Through the use of the battery modelling software PyBaMM, it was possible to run simulations of various charging sessions based on the Doyle-Fuller-Newman model and a parametrization of a LG M50 21700 lithium-ion battery cell. This allowed the generation of data of the negative electrode overpotential, to which an equivalent circuit model was fitted using charging data suitable for parameter estimation.

Through the use of closed-loop PI-control, it was then possible to control the charging current based on the output of the ECM. The PI-controller controls the current so that the negative electrode overpotential (the output from the ECM) is kept above 0 V and the cell voltage does not surpass the cell's maximum allowed voltage of 4.2 V.

While the project did not aim to deliver a complete charging solution ready for implementation in a car's BMS, the results do show that with further development and adaption of the proposed solution, it can lead to an improvement in charging time. A key challenge which presented itself during the larger part of the project is that good parameters are needed to make the ECM accurate enough to be useful, and estimating these parameters can be challenging, as they not only depend on factors such as temperature and SoC, but also on the charge rate.

The thesis has, however, shown that through modelling the negative electrode overpotential with an ECM, it is possible to achieve a shorter charging time compared to conventional MCC-CV algorithms.

Bibliography

[1] V. Sulzer, S. G. Marquis, R. Timms, M. Robinson, and S. J. Chapman, "Python Battery Mathematical Modelling (PyBaMM)," *Journal of Open Research Software*, vol. 9, no. 1, p. 14, 2021.

[2] B. Fridholm, "Adaptive model-based battery management - Predicting energy and power capability," Ph.D. dissertation, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden, 2019.

[3] E. Hosseinzadeh, M. X. Odio, J. Marco, and P. Jennings, "Unballanced performance of parallel connected large format lithium ion batteries for electric vehicle application," in *2019 International Conference on Smart Energy Systems and Technologies (SEST)*, September 2019. Available: https://doi.org/10.1109/SEST.2019.8849060

[4] A. Tomaszewska, Z. Chu, X. Feng, S. O'Kane, X. Liu, J. Chen, C. Ji, E. Endler, R. Li, L. Liu, Y. Li, S. Zheng, S. Vetterlein, M. Gao, J. Du, M. Parkes, M. Ouyang, M. Marinescu, G. Offer, and B. Wu, "Lithium-ion battery fast charging: A review," *eTransportation*, vol. 1, no. 100011, August 2019. Available: https://doi.org/10.1016/j.etran.2019.100011

[5] World Economic Forum, "Electric vehicles for smarter cities: The future of energy and mobility," February 2018. Available: https://www.weforum.org/reports/electric-vehicles-for-smarter-cities-the-future-of-energy-and-mobility

[6] P. Slowik, A. Isenstadt, L. Pierce, and S. Searle, "Assessment of light-duty electric vehicle costs and consumer benefits in the united states in the 2022-2035 time frame," October 2022. Available: https://theicct.org/wp-content/uploads/2022/10/ev-cost-benefits-2035-oct22.pdf?bcs-agent-scanner=e509b122-c131-594e-a607-9336032caa87

[7] R. Lindeberg, "Volvo $2.9 Billion IPO Marks Key Test in Electric Car Shift," *Bloomberg.com*, p. 12, October 2021. Available: https://www.bloomberg.com/news/articles/2021-10-18/volvo-cars-2-9-billion-ipo-marks-key-test-in-electric-car-shift

[8] A. Mukhopadhyay and B. W. Sheldon, "Deformation and stress in electrode materials for Li-ion batteries," *Progress in Materials Science*, vol. 63, pp. 58 –

116, June 2014. Available: https://doi.org/10.1016/j.pmatsci.2014.02.001

[9] A. Wang, S. Kadam, H. Li, S. Shi, and Y. Qi, "Review on modeling of the anode solid electrolyte interphase (SEI) for lithium-ion batteries," *npj Computational Materials*, vol. 4, no. 15, March 2018. Available: https://doi.org/10.1038/s41524-018-0064-0

[10] M. Doyle, T. F. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *Journal of The Electrochemical Society*, vol. 140, no. 6, pp. 1526 – 1533, June 1993.

[11] C. Zou, X. Hu, Z. Wei, T. Wik, and B. Egardt, "Electrochemical estimation and control for lithium-ion battery health-aware fast charging," *IEEE Transactions on Industrial Electronics*, vol. 65, pp. 6635 – 6645, August 2018. Available: https://doi.org/10.1109/TIE.2017.2772154

[12] J. M. L. Fonseca, G. Sambandam Kulothungan, K. Raj, and K. Rajashekara, "A novel state of charge dependent equivalent circuit model parameter offline estimation for lithium-ion batteries in grid energy storage applications," in *2020 IEEE Industry Applications Society Annual Meeting*, October 2020. Available: https://doi.org/10.1109/IAS44978.2020.9334862

[13] P. Notten, J. O. het Veld, and J. van Beek, "Boost charging Li-ion batteries: a challenging new charging concept," *Journal of Power Sources*, vol. 145, pp. 89 – 94, July 2005. Available: https://doi.org/10.1016/j.jpowsour.2004.12.038

[14] S. H. Lim, S. H. Kim, H. M. Lee, S. J. Kim, and Y.-J. Shin, "Design of neural network-based boost charging for reducing the charging time of li-ion battery," in *2020 International Conference on Data Mining Workshops (ICDMW)*, November 2020.

[15] B. K. Purushothaman and U. Landau, "Rapid Charging of Lithium-Ion Batteries Using Pulsed Currents: A Theoretical Analysis," *Journal of The Electrochemical Society*, January 2006. Available: https://doi.org/10.1149/1.2161580

[16] C.-H. Chen, F. B. Planella, K. O'Regan, D. Gastol, W. D. Widanage, and E. Kendrick, "Development of experimental techniques for parameterization of multi-scale lithium-ion battery models," *Journal of The Electrochemical Society*, vol. 167, no. 8, p. 080534, jan 2020. Available: https://doi.org/10.1149/1945-7111/ab9050

[17] Z. Li, X. Shi, M. Shi, X. Wang, Y. Wang, and H. Sun, "Online Estimation of Battery Equivalent Circuit Model Parameters Using Decoupled Least Squares Technique," in *2020 IEEE/IAS Industrial and Commercial Power System Asia (ICPS Asia)*, July 2020.

[18] Z. Cen, P. Kubiak, and I. Belharouak, "Online parameter estimation/tracking for Lithium-ion battery RC model," in *2016 International Renewable and*

Sustainable Energy Conference (IRSEC), November 2016.

[19] Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, R.K. Pachauri and L.A. Meyer (eds.)], *Climate Change 2014: Synthesis Report*. IPCC, Geneva, Switzerland, 2014.

[20] H. Ritchie and M. Roser, "CO_2 and Greenhouse Gas Emissions," 2020. Available: https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions

[21] IEA, Paris, "Transport sector CO_2 emissions by mode in the Sustainable Development Scenario, 2000-2030," 2022. Available: https://www.iea.org/data-and-statistics/charts/transport-sector-co2-emissions-by-mode-in-the-sustainable-development-scenario-2000-2030

[22] F. D. Pero, M. Delogu, and M. Pierini, "Life cycle assessment in the automotive sector: a comparative case study of internal combustion engine (ice) and electric car," *Procedia Structural Integrity*, vol. 12, pp. 521–537, December 2018. Available: https://doi.org/10.1016/j.prostr.2018.11.066

A

LG M50 cell parameters

Name [units]	Value	Reference	Notes
Macroscale geometry			
Negative electrode thickness [m]	85.2E-6	[16]	
Negative particle radial distance [m]	5.68E-6	[16]	
Negative electrolyte volume fraction [%]	25	[16]	
Negative active material volume fraction [%]	75	[16]	
Separator thickness [m]	12E-6	[16]	
Separator electrolyte volume fraction [%]	47	[16]	
Positive electrode thickness [m]	75.6E-6	[16]	
Positive particle radial distance [m]	5.22E-6	[16]	
Positive electrolyte volume fraction [%]	33.5	[16]	
Positive active material volume fraction [%]	66.5	[16]	
Electrode			
Negative solid phase lithium diffusivity [$m^2 s^{-1}$]	1.7E-15	[16]	
Negative solid phase electronic conductivity [Sm^{-1}]	215	[16]	
Negative maximum concentration [mol m^{-3}]	215	[16]	
Positive solid phase lithium diffusivity [$m^2 s^{-1}$]	1.5E-15	[16]	
Positive solid phase electronic conductivity [Sm^{-1}]	0.18	[16]	
Electrode length [m]	1.58	[16]	accounts for both sides of unwound electrode (double-sided coating)

Name [units]	Value	Reference	Notes
Electrode (cont.)			
Electrode width $[m]$	6.5E-2	[16]	
Cell cooling surface area $[m^2]$	5.31E-3	[16]	cylindrical
Cell volume $[m^3]$	2.42E-5	[16]	cylindrical
Current collector properties			
Negative current collector thickness $[m]$	12E-6	[16]	
Negative current collector conductivity $[S/m]$	58411000	CRC Handbook	copper
Positive current collector thickness $[m]$	12E-6	[16]	
Positive current collector conductivity $[S/m]$	36914000	CRC Handbook	aluminium
Density			
Negative current collector density $[kg/m^3]$	8960	CRC Handbook	copper
Positive current collector density $[kg/m^3]$	2700	CRC Handbook	aluminium
Specific heat capacity			
Negative current collector specific heat capacity $[J/(kgK)]$	385	CRC Handbook	copper
Positive current collector specific heat capacity $[J/(kgK)]$	897	CRC Handbook	aluminium
Thermal conductivity			
Negative current collector thermal conductivity $[W/(mK)]$	401	CRC Handbook	copper
Electrical			
Nominal cell capacity $[Ah]$	5	[16]	
Typical current $[A]$	5	[16]	
Current function $[A]$	5	[16]	default current function

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

CHALMERS
UNIVERSITY OF TECHNOLOGY