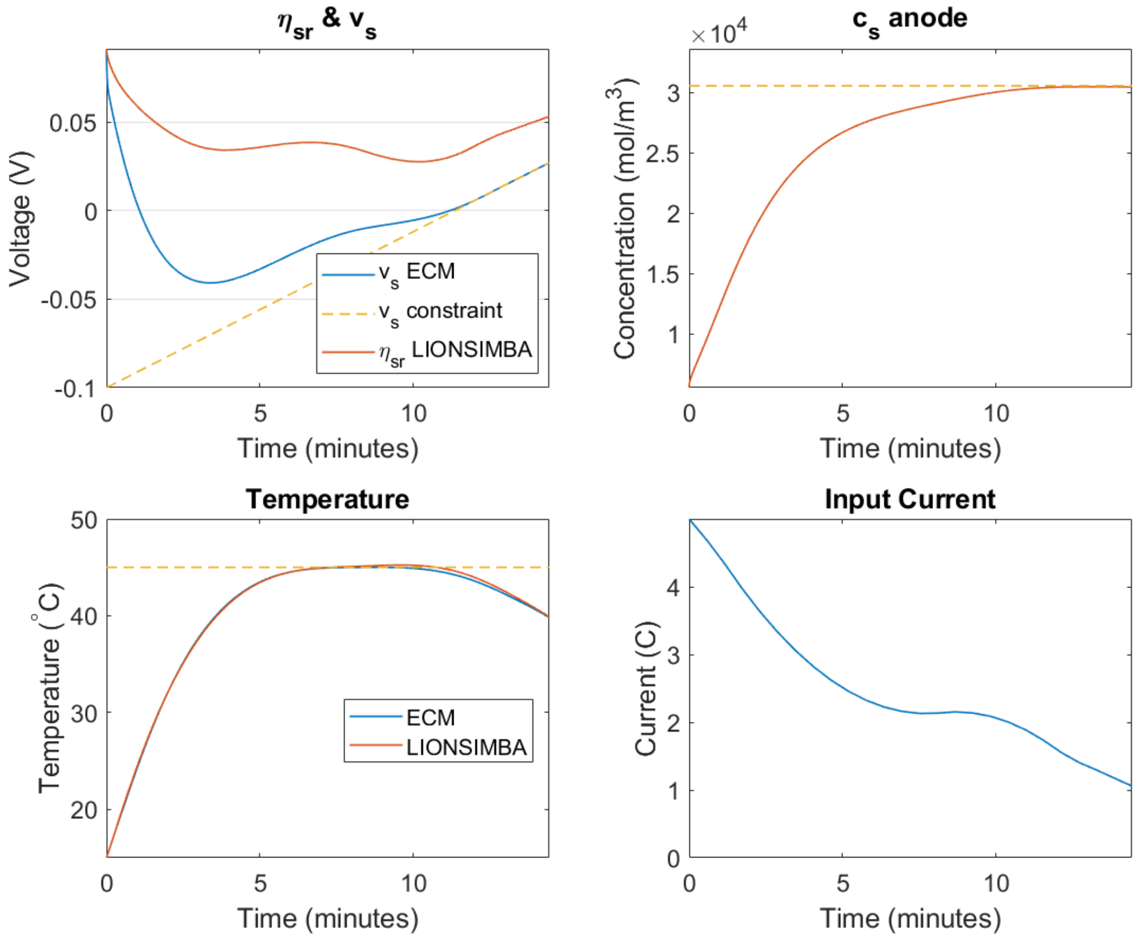
CHALMERS
UNIVERSITY OF TECHNOLOGY



Minimum-time optimal charging of lithium-ion batteries

A simulation based method for safe fast charging with a focus on system identification and constraint customisation

Master's thesis in Systems, Control and Mechatronics

Simon Fitz

Albin Hallberg

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Master's thesis 2018

Minimum-time optimal charging of lithium-ion batteries

SIMON FITZ
ALBIN HALLBERG

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2018

SIMON FITZ
ALBIN HALLBERG

Supervisor: Tobias Olsson, Combine AB
Examiner: Sébastien Gros, Department of Electrical Engineering

Master's Thesis 2018
Department of Electrical Engineering
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Fast charging results for charging from 20-80% at 15 °C

Typeset in LaTeX
Gothenburg, Sweden 2018

Simon Fitz, Albin Hallberg
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Lithium-ion batteries are currently one of the most commonly used type of batteries and in this thesis the main goal has been to find a minimum-time open-loop optimal control strategy for charging a lithium-ion battery. The idea was to use a complex but accurate model of a lithium-ion battery for evaluation and try to fit a simple equivalent circuit model which could be used for optimisation. The optimal control problem was solved using CasADi and it was possible to achieve charging times up to 42% faster than with the conventional CC-CV charging method while still keeping within the same constraints. The biggest decrease in charging time was at lower temperatures for long charging intervals. The method should be applicable for a physical battery although constraints will be more difficult to choose.

Keywords: Fast charging, Minimum-time, Open-loop optimal control, Lithium-ion battery

Acknowledgements

First we would like to thank Lukas Wikander, a Ph.D student working with LIONSIMBA who has helped us with various questions and problems that arose with LIONSIMBA during the thesis work. We would also like to thank our examiner, Sébastien Gros for help with coming up with ideas as well as taking time to be our examiner. Last but not least we would like to thank Tobias Olsson and Combine for providing their support throughout the project.

Simon Fitz & Albin Hallberg, Gothenburg, December 2018

Contents

1

Introduction

1.1 Background

Lithium-ion batteries are interesting as they currently are one of the most commonly used type of batteries [1]. There are multiple different kinds of lithium-ion batteries with varying properties but the one thing they all have in common is that they need to be charged. There exists numerous strategies for charging such as Constant Current(CC) and/or Constant Voltage(CV) charging. More complex charging strategies exist such as multistage constant current charging which make use of predefined current steps and results in a charging current similar in appearance to a descending staircase.

The most common model used in control applications is an Equivalent Circuit Model (ECM), but due to the extensive simplifications in the model compared to a physical battery, a lot of important information is unobservable and therefore unavailable for optimisation. This is a concern when solving optimisation problems since the solution is based on the predetermined boundaries of the states, which are far from trivial to decide, and can therefore cause the solution to be unsuitable for real world applications. To try to deal with this issue, two different models are used for two different purposes, an ECM which is used for optimisation and an Electrochemical Model (EM) for simulation. The latter simulates the battery in more depth and can therefore validate if the charging is somewhat safe and feasible.

1.2 Purpose

The main purpose of the thesis is to investigate how an optimal open-loop control strategy for charging a lithium-ion battery cell can be found. There are many parameters which needs to be taken into consideration to not cause damage to the the battery. The main method to complete this is to develop a minimum-time solver for the ECM where constraints on the available states limits the charging and then evaluate the resulting charging current in simulation in the EM. By doing this, the hope is to achieve a non-destructive charging strategy. As the battery characteristics change with temperature and State of Charge (SoC) the optimisation will be performed for different initial conditions and for different intervals of SoC to see how they compare.

1.3 Limitations

- Existing battery models will be used.
- Only a single cell will be considered.
- The theory behind the software tools used for solving non-linear optimization problems will be excluded from the work.
- No analysis of the battery's expected lifetime will be made.
- No cooling/heating system will be taken into consideration.
- No physical testing will be performed.

2
Theory

This chapter details the theory on which the thesis is based, starting with some general theory about lithium-ion batteries followed by a more detailed description of the models used for simulation and optimisation as well as the principle behind the optimisation method used for minimum-time optimal control.

2.1 Lithium-ion batteries

A normal lithium-ion battery is made up of multiple cells in order to reach the desired voltage and capacity, each cell consisting of a cathode, separator and anode. In figure 2.1, the process of charging a battery is shown. The current is notated as flowing from the anode through the charger to the cathode, which means that the electrons flow in the opposite direction. Simultaneously, the lithium ions migrate from the cathode to the anode. The battery is fully charged when the anode has reached its maximum lithium-ion storage capability.

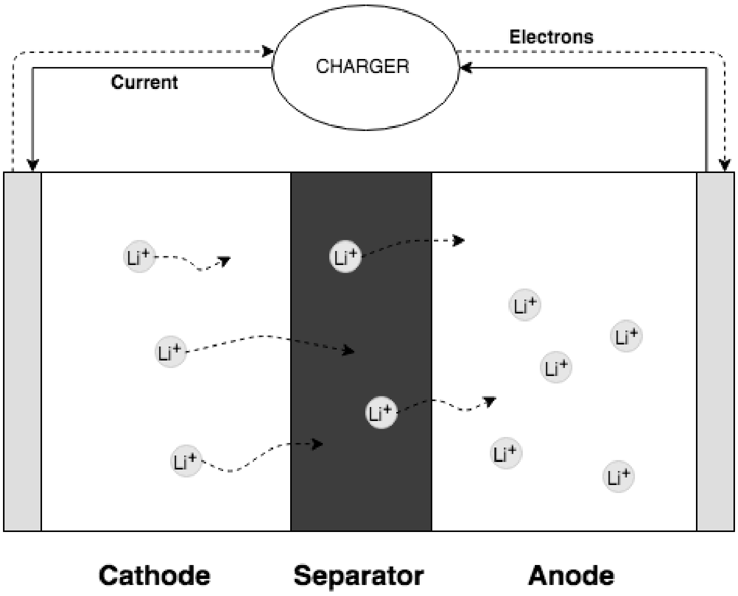


Figure 2.1: The process of charging in a lithium-ion battery, showing the lithium ions moving from the cathode to the anode

The characteristics and performance of a cell, such as capacity, lifetime and charging/discharging performance varies with the materials being used; the term "lithium-ion battery" represents a wide variety of chemistries and cell designs [2]. As a result, the optimal charging profile for one type of battery might be different from another depending on which materials are used. There are also different types of lithium-ion cells in regards of structure; cylindrical, prismatic and pouch cells are all common. As an example, the Tesla Model 3 uses a high number of cylindrical cells for their cars whereas the Nissan Leaf and the Chevrolet Bolt uses pouch cells [3][4][5].

Lithium-ion batteries consists of dual-intercalation cells, making them rechargeable since the ions can be stored in both the cathode and the anode. The term intercalation can be described as moving ions in and out of an interstitial site of a lattice. The anode is often made of graphite and the cathode of a combination of active materials, where numerous alternatives exists such as $LiCoO_2$ (Lithium Cobalt Oxide) or $LiNCAO_2$ (Lithium Nickel Cobalt Aluminium Oxide) [6].

2.2 Battery models

Models which describe and monitor different processes and states of the battery are very useful and in this thesis two different models are discussed and used, the EM and the ECM. The most significant difference between the two is that the EM models what happens chemically in the battery while the ECM only replicates the behaviour of the cell's terminal voltage. Because of this, the EM has a much higher complexity which is why the optimisation is based on the ECM and the EM is used for verification and testing.

2.2.1 Electrochemical model

Doyle et al.[7] developed a model in 1993 of the lithium ions' intercalation process and expanded models have since then been developed further, the principle behind the one used in this thesis is shown in figure 2.2. The model consist of five sections, a current collector at each side, a cathode, a separator and an anode. The current collectors are made of a conductive material, like copper or aluminium, that distributes the current uniformly. In the positive and negative electrode, i.e. cathode and anode, the lithium ions can be stored in the solid phase, which is modelled and visualised as circular solid-phase particles. The separator makes sure no electrons, only lithium ions, are able to pass. The cathode-separator-anode section is filled with electrolyte, which enables the lithium ions to move. The lithium ions can therefore be in either the electrolyte or in the particles in the cathode or the anode. Included in the figure as well is the Solid Electrolyte Interphase (SEI) film, which is not taken into consideration here but can play a large role in aged batteries.

Also seen in figure 2.2 is the charging process, including the movement of ions. The ions starts with exiting the particles of the cathode, making their way past the separator into the anode to finally intercalate into the particles in the anode.

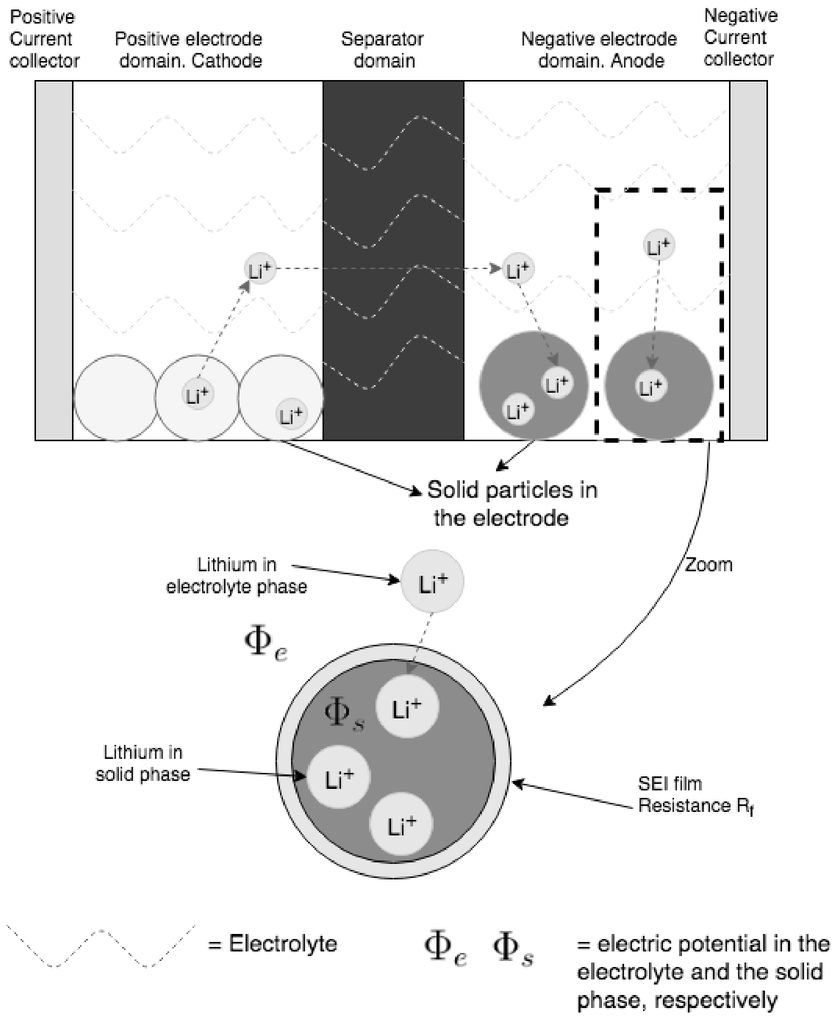

Positive Current collector
Positive electrode domain. Cathode
Separator domain
Negative electrode domain. Anode
Negative Current collector
Li+
Solid particles in the electrode
Zoom
Lithium in electrolyte phase
Φ_e
Φ_s
Lithium in solid phase
SEI film Resistance R_f
= Electrolyte
Φ_e Φ_s = electric potential in the electrolyte and the solid phase, respectively

Figure 2.2: A model of a battery with the components relevant during charging, showing the lithium ions moving from the cathode to the anode. Increased impedance of the SEI over time is one of the main causes of increased overall impedance in an ageing battery.

The behaviour of the battery is described by a set of partial differential equations which cover the electric potentials in the solid phase and electrolyte, the transport of ions in the electrolyte and solid phase, the conservation of charge and the relationship between the molar flux and the solid phase intercalation overpotential [8]. The overpotential is crucial in fast charging since it is closely related to a side reaction that causes lithium plating, a phenomenon that damages the cell and degrades the capacity. The side reaction that causes lithium plating, η_{sr}, can by making some general assumptions be described as the difference in electric potential in the solid phase and the electrolyte in the anode and should stay above zero, according to the following equations:

$$\eta_{sr} = \Phi_s - \Phi_e$$

$$\eta_{sr} > 0$$

Another important part of fast charging is the highest allowed ion concentration in the particles in the anode, which is connected to the battery's state of charge. This can be seen in equation 2.1, where t is the time, L_n is the length of the anode and c_s is the lithium-ion concentration in the particles. The cell is fully charged when all particles in the anode have reached their maximum concentration. During fast charging the concentration in the particles closest to the separator is maximised first which raises the risk of overcharging those particles. Overcharging a battery is a safety risk where problems like swelling and, in extreme cases, explosions can occur. Exceeding the maximum ion concentration even by a small amount causes destruction of the electrode materials which negatively affects the mobility of the electrons and lithium-ions. This results in higher internal resistance, capacity fading as well as a higher rate of heat generation [9][10].

$$SoC(t) = \frac{1}{L_n c_s^{max,n}} \int_0^{L_n} c_s^{avg}(x,t)dx \tag{2.1}$$

2.2.2 LIONSIMBA

The EM model has been implemented in MATLAB by Torchio M et al. [11], calling it LIONSIMBA, Lithium-ION SIMulation BAttery toolbox. The battery cell is modelled according to the battery parameters used in the works of Northrop et al. [12][13]. The materials used in the anode is graphite (C_6) and in the cathode $LCoO_2$, or LCo, which is a type commonly seen in mobile phones [6]. In LIONSIMBA it is possible to set up and initialise the battery, run simulations and observe processes of interest like the side reaction voltage, η_{sr}, as well as the lithium-ion concentration in the anode.

LIONSIMBA simulates a predefined number of particles in the cathode and anode separately. As an example, figure 2.3 shows the side reaction voltage η_{sr} as well as the lithium-ion concentration in ten simulated particles in the anode. In the example the battery was charged with a constant current for 1000 seconds. The particle closest to the separator is closest to potentially reaching harmful levels which is visible in both graphs; to the left where its corresponding value of η_{sr} is closest to zero, and to the right where its lithium concentration is closest to the maximum limit. Henceforth, only the particle closest to the separator is observed when comparing performance of different charging strategies.

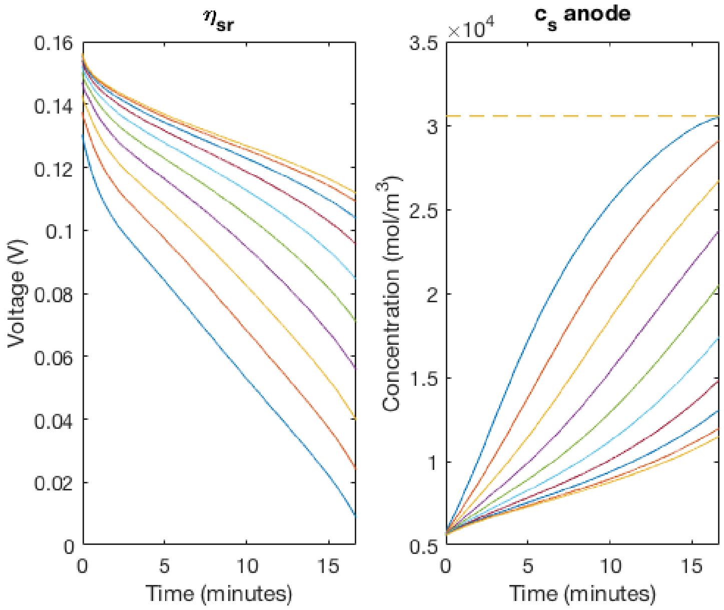


Figure 2.3: The figure shows the side reaction voltage η_{sr} and the lithium-ion concentration in the solid phase for 10 particles in the anode after charging the battery with constant current for 1000 seconds. η_{sr} is never below zero and the concentration never exceeds the limit for this battery.

While this model describes the dynamics in an actual battery fairly well, it is too complex for the objective of finding a minimum-time optimal charging current. A simpler model is needed for the optimisation, and an equivalent circuit model is used for this.

2.2.3 Equivalent circuit model

The ECM is an electric model which uses passive electric components to model the terminal voltage of the battery. The model is commonly used and figure 2.4 shows the circuit [14].

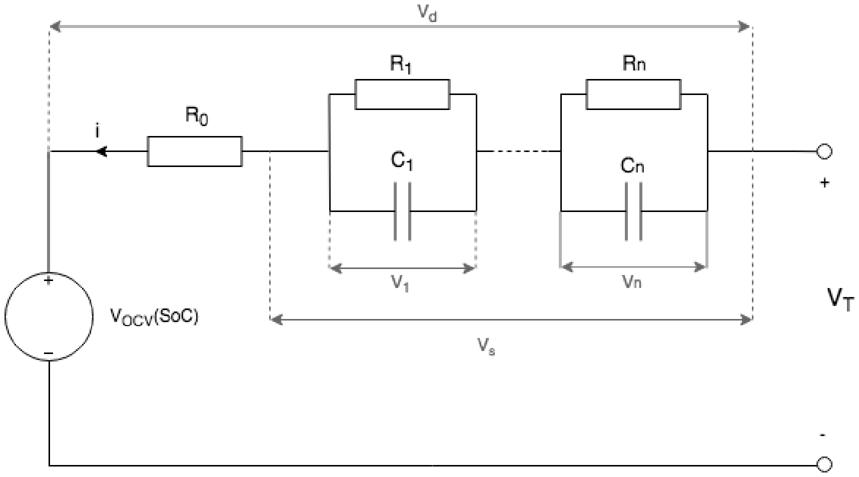


Figure 2.4: The equivalent circuit model (ECM) which models the dynamics of a battery using common electrical components.

The model can, by removing all resistors and capacitors, be seen in its simplest form as an ideal voltage source with the same voltage as the Open Circuit Voltage (OCV) of the battery. The OCV of the battery is the voltage the battery has when rested and unloaded, and is mainly dependent on SoC. In the equations the SoC is notated as z, which ranges from 0 to 1, and is modelled based on the principle of coulomb counting as shown in equation 2.2. The current is positive for charging and Q is the capacity of the battery, usually measured in Ah.

$$z(t) = z(t_0) + \frac{1}{Q}\int_0^t i(\tau)d\tau \tag{2.2}$$

An ideal voltage source alone does not capture the dynamics of the battery which is why the resistors and capacitors are added. Adding a resistance, R_0, in series with the ideal voltage source captures the voltage drop that occurs because of the internal resistance of the battery. This is shown in figure 2.5 where a current pulse is used as input. The biggest part of the voltage change comes from $R_0 i(t)$ for the part when the current is applied. The OCV rises as the SoC increases and the terminal voltage coincides with the OCV after a period without input. The nonlinear dynamics that the series resistor and the ideal voltage source do not capture, is captured by the RC-pairs as the voltage notated as v_s, where $v_s = v_T - v_{OCV} - R_0 i(t)$.

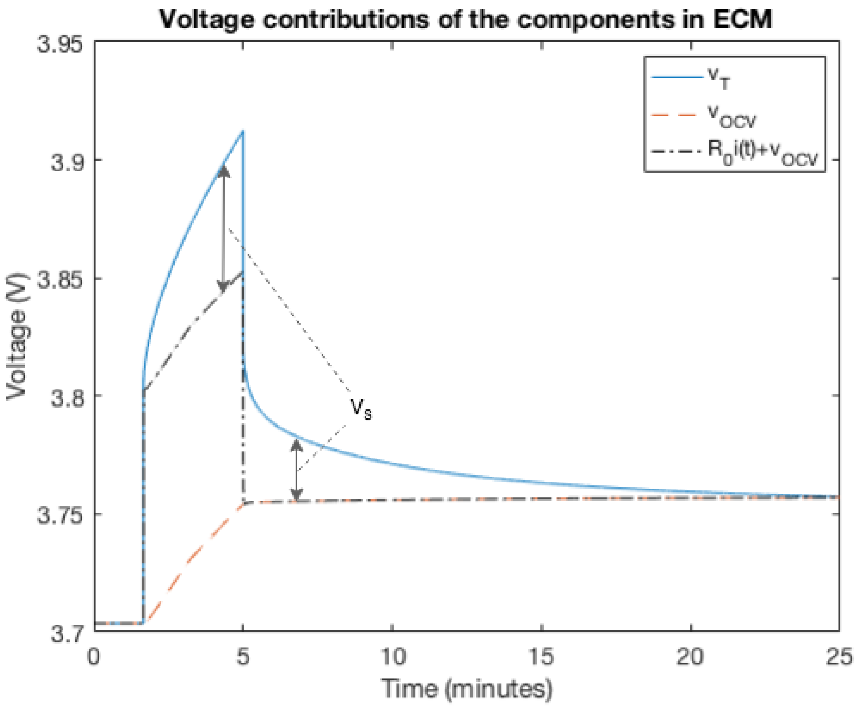


Figure 2.5: The terminal voltage, v_T, is seen here as the voltage response after a current pulse has been applied to the battery. The series resistor, R_0, captures the main part of the rise and drop in voltage while the voltage over the RC-pairs, v_s, model the non-linear parts. As the input current is positive, the SoC of the battery will rise as is seen by the small rise in voltage for the open circuit voltage, v_{OCV}.

The resulting terminal voltage, v_T, of the battery is then calculated according to equation 2.3.

$$v_T(t) = OCV(SoC(t)) + i(t)R_0 + \sum_i^n v_i(t) \tag{2.3}$$

Determining v_i can be done by using Kirchhoff's current law and then extending with R_i since $v_i = i_{R_i}R_i$, which can be seen in equation 2.4. The reformulated version used later in the state-space model can be seen in equation 2.5.

$$i_{R_i}(t)R_i = (i(t) - C_i\frac{dv_i}{dt})R_i \tag{2.4}$$

$$\frac{dv_i}{dt} = -\frac{v_i(t)}{C_iR_i} + \frac{i(t)}{C_i} \tag{2.5}$$

2.2.4 Thermal model

For the ECM to be useful for optimisation, a model for how the temperature of the cell behaves is needed as well. One commonly used model consists of two states, one for the core temperature and one for the surface temperature of the cell [15][16]. However, for modelling and parameter estimation simplicity, a model with only one state is used, which is assumed to be the average temperature of the cell. Equation 2.6 shows how the model works.

$$C_c \frac{dT_c}{dt} = \frac{T_{amb} - T_c}{R_c} + Q_{ECM} \tag{2.6}$$

C_c is the specific heat capacity of the cell and R_c is the heat conduction resistance between the cell and the air around it. T_c is the temperature of the cell and T_{amb} is the ambient air temperature. Q_{ECM} is the generated heat from all components in the ECM, which is calculated as $Q_{ECM} = (R_0 i + v_1 + ... + v_n)i$.

2.3 State-space model

To simulate the battery, the ECM and thermal model needs to be combined into a single state-space system. As all the equations are defined already it can simply be defined as seen in equation 2.7.

$$\begin{bmatrix} \dot{v}_1 \\ \vdots \\ \dot{v}_n \\ \dot{z} \\ \dot{T}_c \end{bmatrix} = \begin{bmatrix} -\frac{v_1}{R_1 C_1} + \frac{i}{C_1} \\ \vdots \\ -\frac{v_n}{R_n C_n} + \frac{i}{C_n} \\ \frac{i}{3600Q} \\ \frac{T_{amb} - T_c}{R_c C_c} + \frac{Q_{ECM}}{C_c} \end{bmatrix} \tag{2.7}$$

2.4 Minimum-time optimal control

The goal with optimal control is to find a set of inputs which minimises a pre-defined cost function without breaking any of the constraints. In the case of fast charging, the variable which needs to be minimised is time. This results in a problem which has the general form as seen in equation 2.8,

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimise}} \quad & T \\ \text{subject to} \quad & \dot{x} = f(x, u) \\ & g(x, u) \leq 0 \\ & x(T) = x_f \end{aligned} \tag{2.8}$$

where x are the states, u the input which we control the system with and T is the time it takes to reach the target state x_f.

2.4.1 Reformulating the problem

As it is not possible to solve a minimisation problem directly when on the form shown in equation 2.8 it needs to be reformulated. One common way to solve minimum-time optimisation problems is to rewrite it on the form shown in equation 2.9.

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimise}} \quad & \xi \\ \text{subject to} \quad & \dot{x} = \xi f(x,u) \\ & g(x,u) \leq 0 \\ & x(t_f) = x_f \end{aligned} \tag{2.9}$$

Two things are changed, the variable ξ, which speeds up or slows down the dynamics of the system, is added and an assumption of what t_f will be is made. The solver will minimise ξ by finding inputs which reaches the final state with as slow dynamics as possible, but as the dynamics of the real system does not change, the actual final time will be $T = t_f \cdot \xi$. Since ξ is minimised the same result will be obtained as if the problem defined in equation 2.8 was solved.

2.5 C-Rate for Batteries [C]

A common way of describing charging and discharging currents in the field of batteries is to use C-rates and it is therefore used in this thesis as well. C-rate is used by letting 1 C be defined as the capacity of the battery, so if the battery has capacity of 2.3 Ah, charging with 1 C would in this case be to charge with a current of 2.3 A. The reason why this is useful is that the capacity varies a lot between different batteries, and the rate at which you can charge with is linearly dependent on the capacity.

3
Method

To be able to perform the minimum-time optimisation with the ECM, model parameters needed to be estimated. In this thesis the model is based on the battery simulated in LIONSIMBA, which means that the measurements for parameter estimation had to be performed in LIONSIMBA. The first part of this chapter describes the method used for system identification followed by two different modelling approaches and finally some details about how the optimisation was performed.

3.1 System identification

Three different types of tests were needed to identify all parameters of the ECM and temperature model. The first step was to estimate the OCV as a function of SoC and temperature, followed by estimating the values of all components in the ECM. Lastly, the two parameters for the temperature model needed to be identified.

3.1.1 Open circuit voltage

The OCV has a big impact on the estimated values which makes it important to have accurate data. LIONSIMBA supports the possibility to initialise the battery at a specific SoC and therefore the open circuit voltage was measured by initialising the battery at different SoCs ranging from 0-100%, as well as different temperatures, and then measuring the voltage. The result can be seen in figure 3.1. The temperature dependence is very small compared to the SoC dependence but was chosen to be taken in consideration as it made a difference in some cases, especially with smaller inputs. The capacity of the battery is predefined in LIONSIMBA to be 29.23Ah, meaning that a charging rate of 1 C is 29.23A.

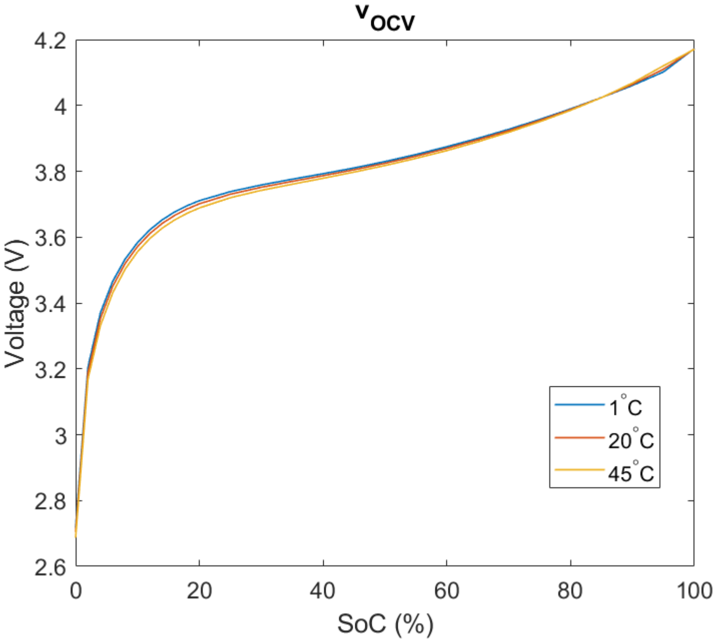


Figure 3.1: v_{OCV} as a function of SoC at different temperatures

3.1.2 Estimating component values for the ECM

The principle behind the parameter estimation is to use the voltage difference, v_d, between the terminal voltage, v_T, and the already measured OCV. As an example, figure 3.2 shows v_T and the OCV for the corresponding SoC as well as the difference between them. The input used is a step beginning with a 10 second rest followed by an input of 2 C for 1000 seconds.

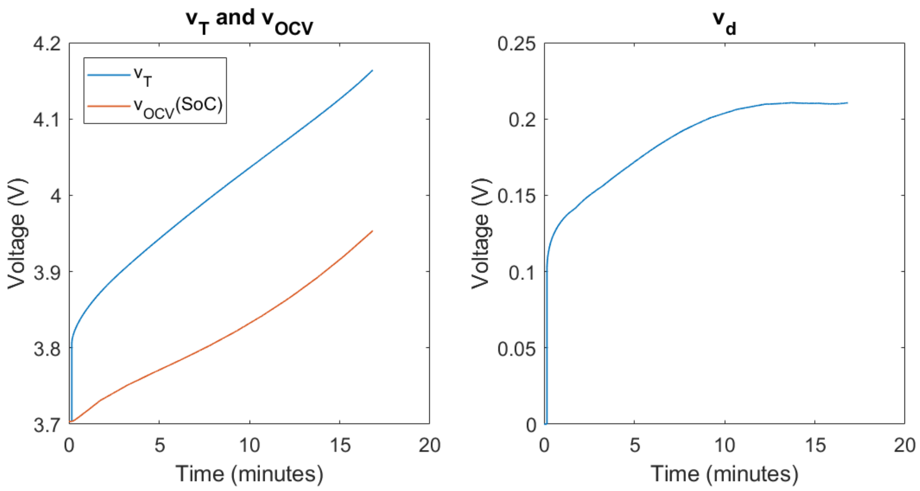


Figure 3.2: The terminal voltage and the open circuit voltage to the left after an input current of 2 C for 1000 seconds had been applied. To the right is v_d, the difference between v_T and v_{OCV}, which is the voltage that the parameter estimation is based on.

The principle for the estimation is the same independent of the number of RC-pairs used, but here it is shown for two RC-pairs as that is what was used for reasons shown further on. To estimate the parameters for the ECM based on the measurements, a discrete time state-space system which models the voltage v_d was used. The state-space model is shown in equations 3.1 & 3.2, which is a discretised version (assuming zero order hold for i) of the first two states of the state-space model described in equation 2.7.

$$\mathbf{x}(k+1) = \begin{bmatrix} e^{-\frac{T_s}{R_1C_1}} & 0 \\ 0 & e^{-\frac{T_s}{R_2C_2}} \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} R_1(1-e^{-\frac{T_s}{R_1C_1}}) \\ R_2(1-e^{-\frac{T_s}{R_2C_2}}) \end{bmatrix} i(k) \tag{3.1}$$

$$v_d(k) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \mathbf{x}(k) + R_0 i(k) \tag{3.2}$$

In the equations above, the state vector $\mathbf{x} = [v_1 \quad v_2]^T$ consists of the voltages over the two RC-pairs. To identify the values of the parameters, the transfer function of the state-space representation, calculated as $G(z) = C(zI-A)^{-1}B + D$, was used, which can be seen in equation 3.3.

$$G(z) = \frac{V_d(z)}{I(z)} = R_0 + \frac{R_1\left(\mathrm{e}^{\frac{\mathrm{Ts}}{C_1\,R_1}} - 1\right)}{z\,\mathrm{e}^{\frac{\mathrm{Ts}}{C_1\,R_1}} - 1} + \frac{R_2\left(\mathrm{e}^{\frac{\mathrm{Ts}}{C_2\,R_2}} - 1\right)}{z\,\mathrm{e}^{\frac{\mathrm{Ts}}{C_2\,R_2}} - 1} \tag{3.3}$$

It was then simplified through substitution of expressions as follows:

$$a = e^{\frac{T_s}{C_1R_1}}, \quad b = e^{\frac{T_s}{C_2R_2}} \implies$$

$$\frac{V_d(z)}{I(z)} = R_0 + \frac{R_1\,(a-1)}{a\,z-1} + \frac{R_2\,(b-1)}{b\,z-1} \tag{3.4}$$

Next, both sides were multiplied with $(az-1)(bz-1)$ to get rid of the denominators. After transforming back to the time domain, this resulted in coefficients on the form that can be seen in equation 3.5.

$$v_d(k+2) = c_1 i(k+2) + c_2 i(k+1) + c_3 i(k) + c_4 v_d(k+1) + c_5 v_d(k) \tag{3.5}$$

To make use of this, a vector of regressors, Φ, was defined

$$\Phi = [i(k+2) \quad i(k+1) \quad i(k) \quad v_d(k+1) \quad v_d(k)]$$

as well as a parameter vector, θ.

$$\theta = [c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5]^T$$

Finally the observation, $Z = v_d(k+2)$ was defined. For all vectors k ranges from 0 to $N-2$, where N is the number of samples. This resulted in a system of equations on the form

$$Z = \Phi\theta$$

By doing a least squares fitting to find θ, it was then possible to find the values of all 5 parameters by solving equation 3.6 for R_0, R_1, R_2, a and b. By how a and b is defined in equation 3.4 it was simple to calculate C_1 and C_2 when a, b, R_1 and R_2 were known.

$$\theta = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} R_0 \\ -\frac{R_0\,a+R_2\,a+R_0\,b+R_1\,b-R_1\,a\,b-R_2\,a\,b}{a\,b} \\ \frac{R_0+R_1+R_2-R_1\,a-R_2\,b}{a\,b} \\ \frac{a+b}{a\,b} \\ -\frac{1}{a\,b} \end{bmatrix} \tag{3.6}$$

3.1.3 Choosing the number of RC-pairs

In the ECM, the number of RC-pairs used affects how well the model can follow the dynamics of the battery. In Lin X et al. [16] a comparison between one, two and three RC pairs was made and the result showed that the difference between one and two pairs was fairly large when compared to the difference between two and three. To provide more ground for the decision, the three different alternatives were compared to LIONSIMBA as well. By constructing methods to calculate the parameter values for all three cases according to the method described above, a comparison could be made for different types of input currents and the result was evaluated by comparing the improvement in percent with two vs. three RC-pairs over a single one in terms of Root Mean Square Error (RMSE). In each example shown the parameters were estimated for the specific input current used. Figure 3.3 shows an example of what the general result looked like, in this case an input current of 2 C for 1000 seconds following the initial rest was used at a starting temperature of 15 $°C$ and initial state of charge of 20%.

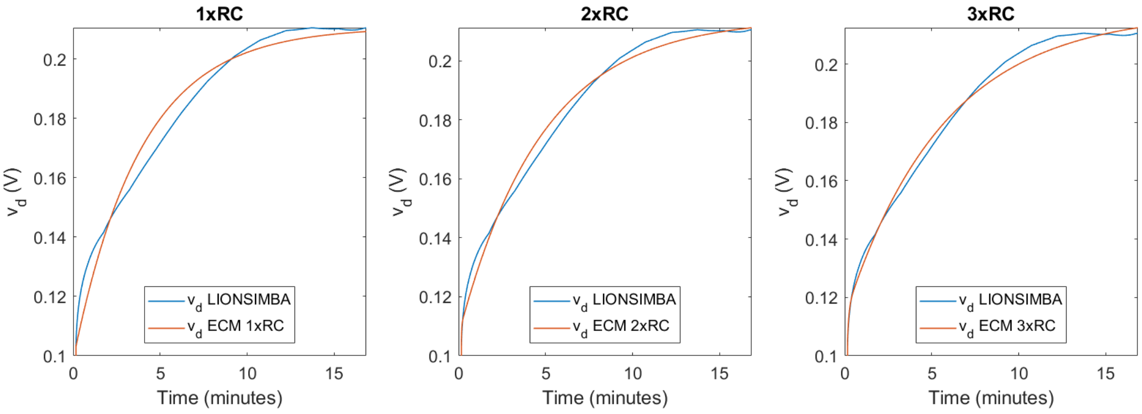


Figure 3.3: Comparison between one, two and three RC-pairs with an input of 2 C for 1000 seconds.

In this case the difference between two and three RC-pairs were fairly small, a 37% decrease in RMSE with two RC-pairs and a 44% decrease with three. Another example can be seen in figure 3.4 where the fast charging curve for charging from 10-95% at 1 $^{\circ}C$ was used as input.

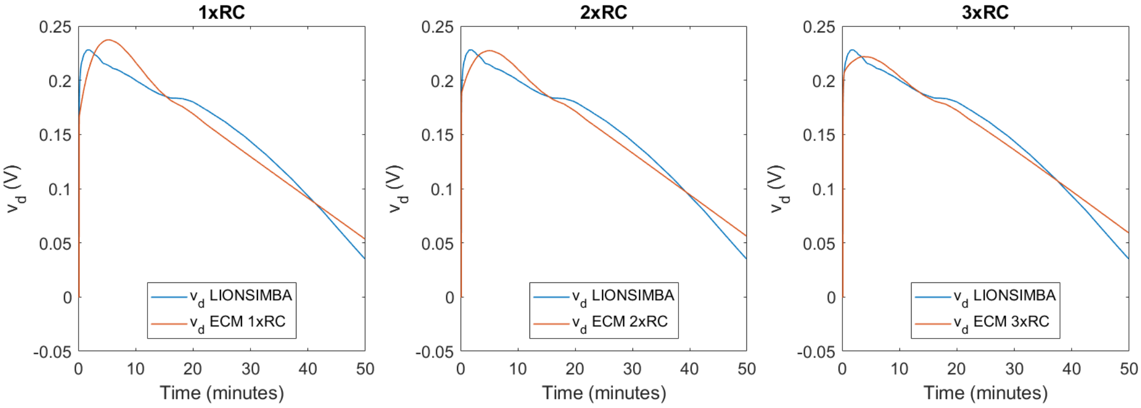


Figure 3.4: Comparison between one, two and three RC-pairs using a charging curve for charging from 10-95% at 1 $^{\circ}C$ as input.

In this case, there was a 25% decrease in RMSE with two RC-pairs and a 33% decrease with three. Based on these results and tests with other inputs it was decided that two RC-pairs was a good compromise between complexity and accuracy.

3.1.4 Estimating thermal model parameters

Estimating the two parameters in the thermal model was done in a similar manner as the ECM components. By expanding the previously defined thermal model shown in equation 2.6,

$$\frac{dT_c}{dt} = (T_{amb} - T_c)\frac{1}{R_c C_c} + Q_{ECM}\frac{1}{C_c}$$

it was simple to choose a vector of regressors,

$$\Phi = [(T_{amb} - T_c) \quad Q_{ECM}]$$

a parameter vector,

$$\theta = \begin{bmatrix} \frac{1}{R_c C_c} & \frac{1}{C_c} \end{bmatrix}^T$$

and the observations were chosen as $Z = \dot{T}_c$. Which, as before, gave us a system on the form

$$Z = \Phi\theta$$

which could be solved for R_c & C_c.

3.2 Parameter fitting

The dynamics of the battery change depending on both SoC and temperature. To illustrate this, figure 3.5 & 3.6 shows how the estimated values vary with either SoC or temperature, with the other at a constant value. The parameter estimations are based on a 500 second step with a current rate of 0.25 C.

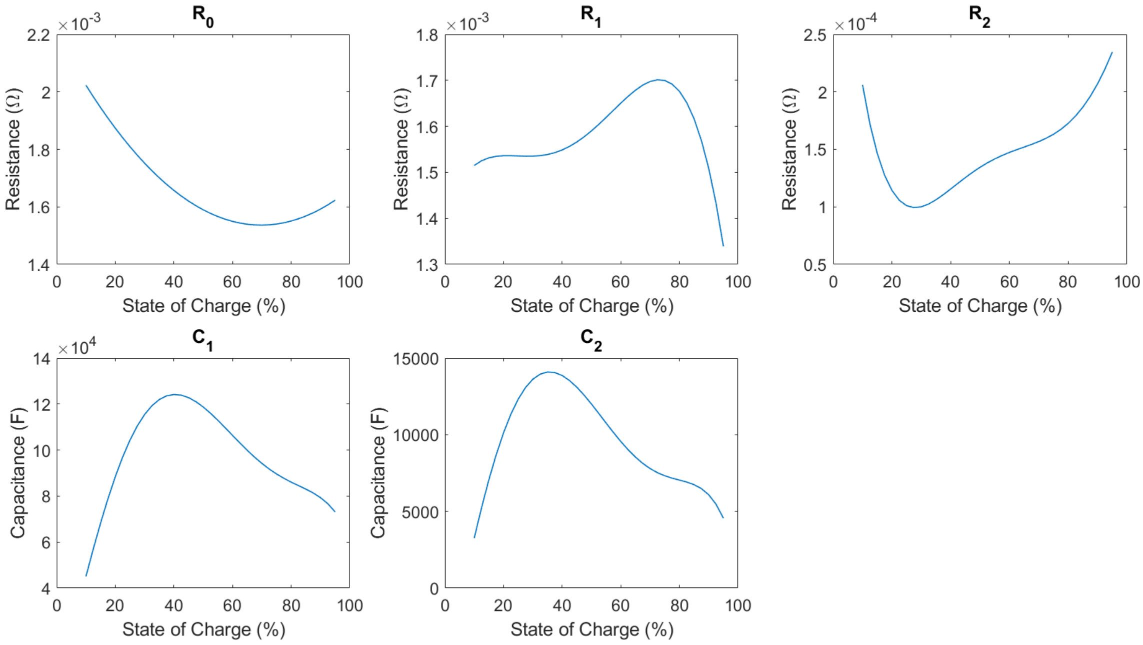


Figure 3.5: ECM parameter values as a function of SoC at $T = 15°C$

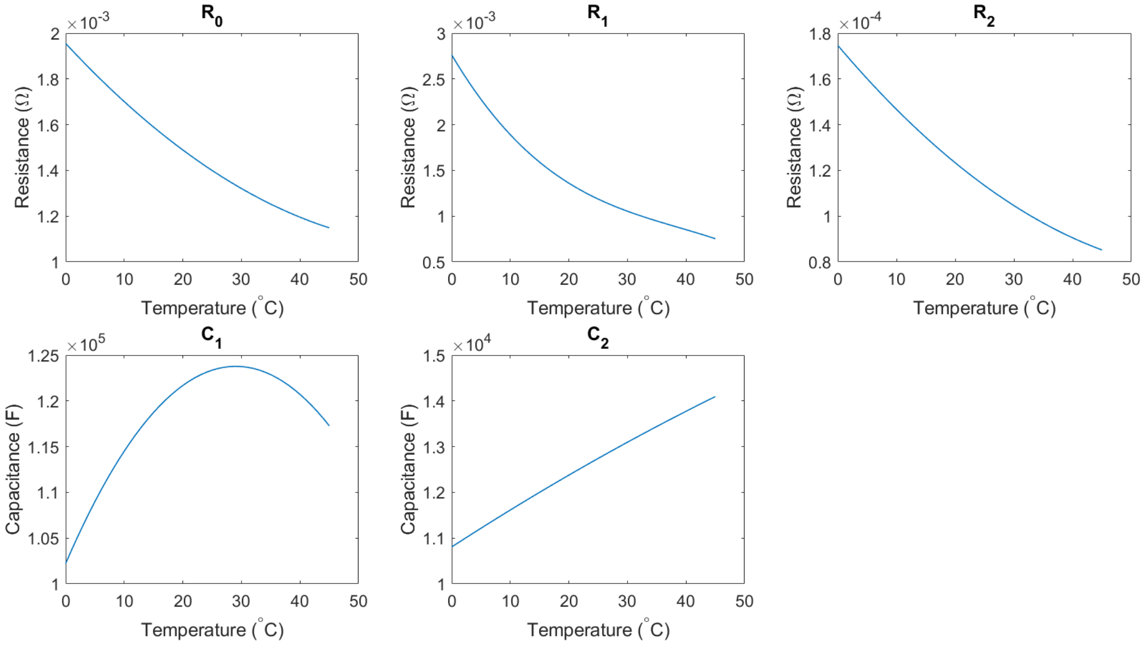


Figure 3.6: ECM parameter values as a function of temperature at $SoC = 50\%$

To handle these variations, two different approaches towards the modelling were tested.

3.2.1 Parameters defined as polynomials

The first approach was to create functions depending on both SoC and temperature for each separate parameter. To do this, estimations of the parameter values were made at different points of temperature and SoC. SoC was ranged from 10-95% in 5% intervals and at each SoC temperature was ranged from 0 to 45 $°C$ in 2.5 $°C$ intervals. After data had been collected, MATLAB's fit function which can generate polynomials of different degrees in each variable were a simple solution to create functions depending on temperature and SoC for the parameters. By evaluating the RMSE for different degrees of polynomials in each variable it was decided that polynomials according to table 3.1 was the best choice in regards of complexity and accuracy. Figure 3.7 shows the measurement points as well as the surfaces based on the polynomials.

Table 3.1: Polynomial degrees for paramter fitting

Parameter	Degree in SoC	Degree in temperature
R_0	2	2
R_1	4	3
R_2	5	2
C_1	5	2
C_2	5	2

Different types and lengths of input currents for the estimation were evaluated but as both SoC and temperature had to remain close to their initial values for it to make sense, the amount of energy which could be used at each point was fairly limited. The best result was achieved with a 500 second step with a charging rate of 0.25 C. Other types of input currents were tested as well, such as single or multiple pulses, short or long, however non of these got close to the performance of a long step.

As both temperature and SoC are included as states in the state-space model, it was fairly simple to make the parameters depend on the current state when optimising the charging. Worth mentioning is that it had a big impact on the computational time as the model became a lot more complex with 5th degree polynomials in it.

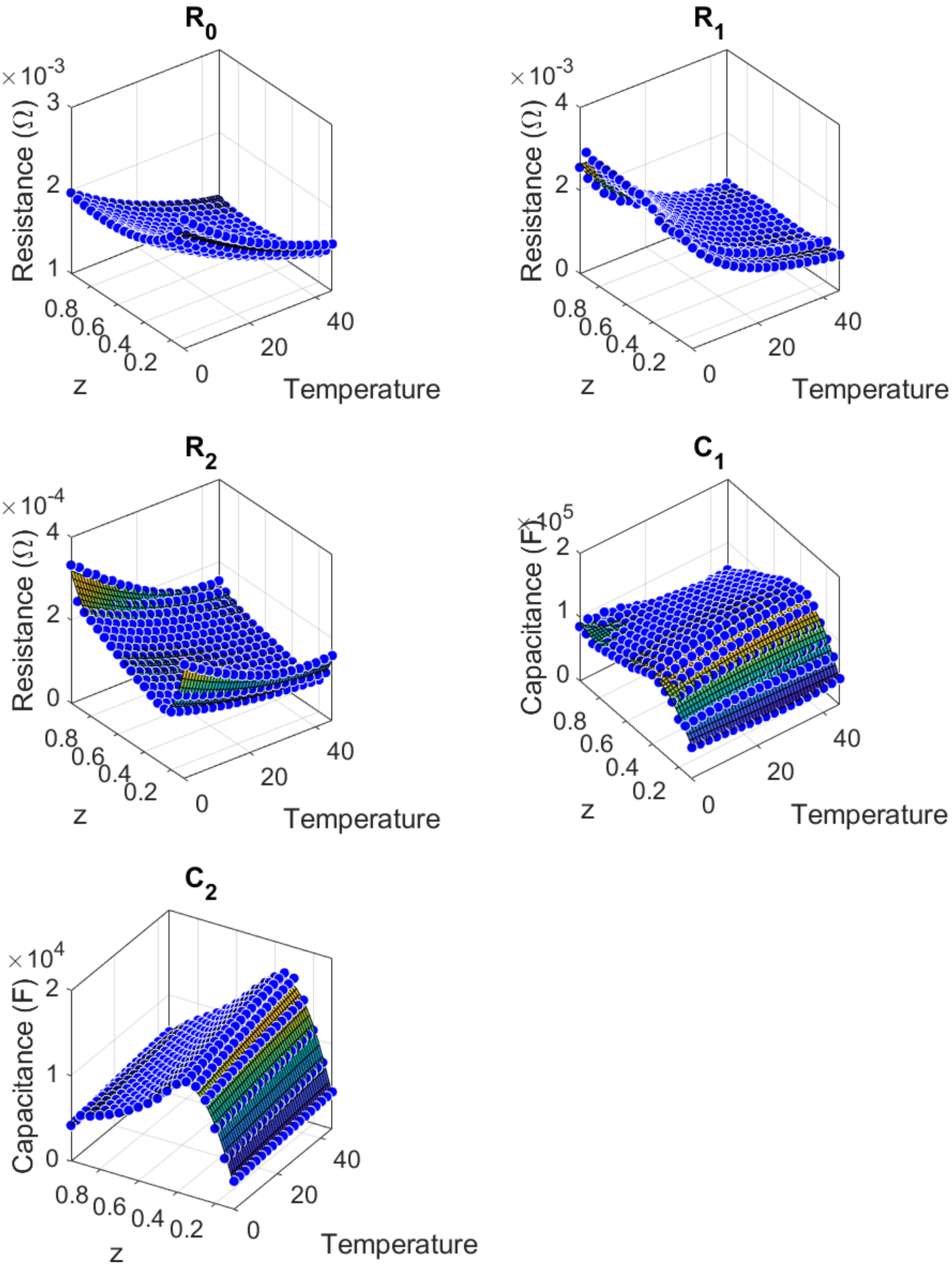


Figure 3.7: ECM parameter values as functions of temperature and SoC

3.2.2 Static parameters

The second approach tested was to use constant parameters. By leaving the parameters constant it was instead possible to estimate the best parameters for the whole charging sequence instead of the alternative approach above. This resulted in one set of parameters for each case evaluated in the results section.

3.2.3 Performance comparison

To compare the two modelling methods, several different types of input currents were used. As a first example, figure 3.8 shows the result when an input current of 2 C for 1000 seconds was used, with an initial SoC of 20% at 15 $°C$. In each of the cases shown the static parameters have been optimised for the specific current.

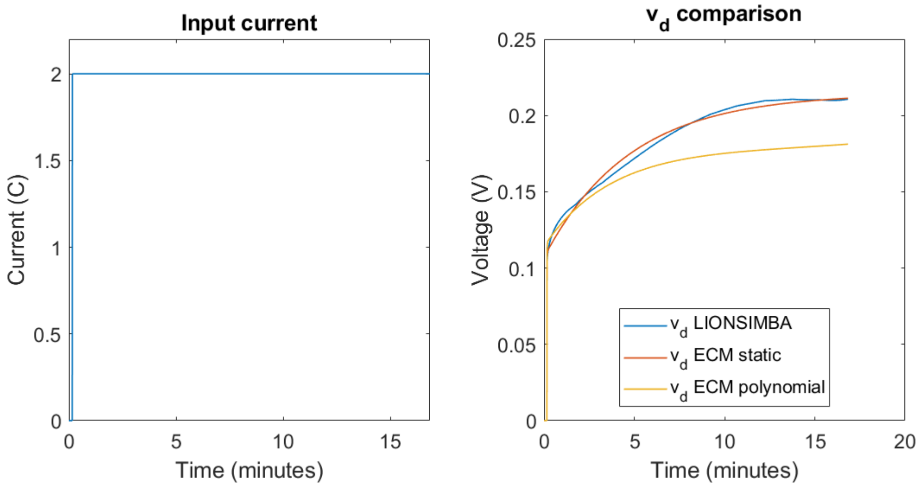


Figure 3.8: Comparison of performance with the estimated ECM parameters as polynomials or kept static using a current step as input.

In this case the model with static parameters was clearly more accurate, and figure 3.9 and 3.10 shows two more cases with the same result. For these two cases optimised fast charging currents were used as input.

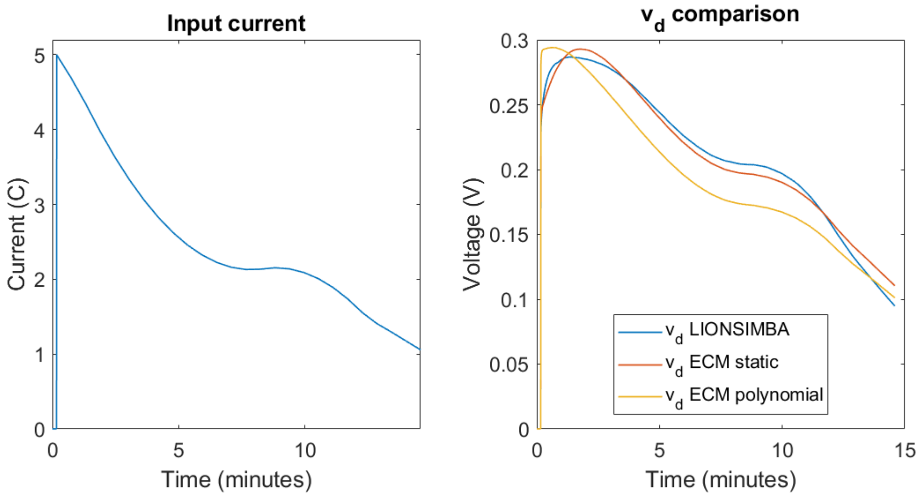


Figure 3.9: Comparison of performance with the estimated ECM parameters as polynomials or kept static using a charging curve as input.

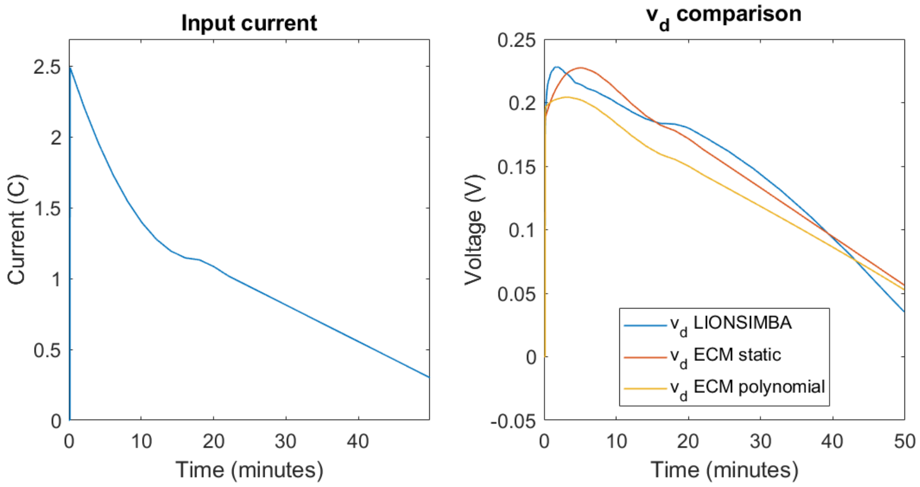


Figure 3.10: Comparison of performance with the estimated ECM parameters as polynomials or kept static using a second charging curve as input.

Lastly, just for illustration, figure 3.11 shows the result when the same input which was used for finding the data for the polynomials is used, which leads to close to identical performance.

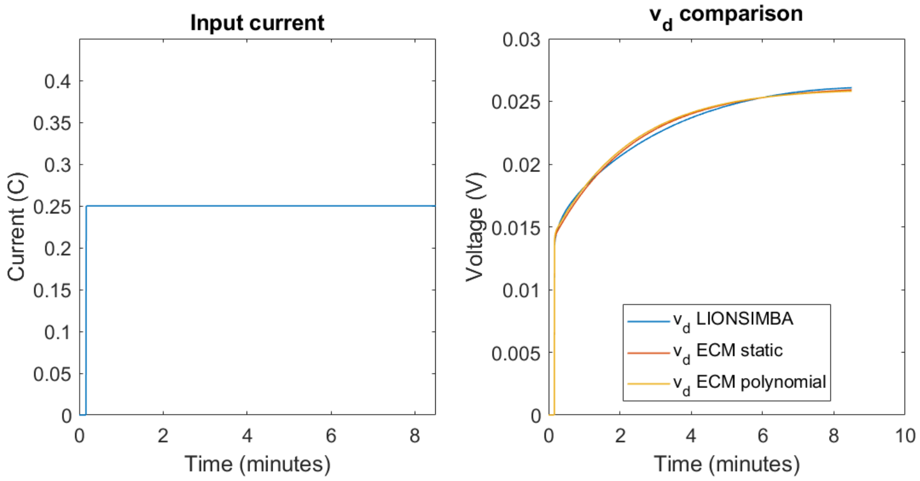


Figure 3.11: Comparison of performance with the estimated ECM parameters as polynomials or kept static using the same current step that was used for estimating the polynomial parameters as input.

Based on these results the model with static parameters was the obvious choice. The complexity was lower and the accuracy was higher for all cases tested.

3.3 Solving the optimal control problem

To solve the minimisation problem based on the state space model in equation 2.7 a plug-in for MATLAB called CasADi was used [17][18]. CasADi is a toolbox developed for efficient numerical optimisation using Automated Differentiation (AD), which makes it extremely useful for this kind of problems, however, how CasAdi and

AD works is outside the scope of this thesis. The method used to solve the minimisation problem is called direct multiple shooting which is a numerical method for solving boundary value problems. A summary of the minimisation problem can be seen in Appendix A and here follows a more detailed description of the key parts.

The Non-Linear Programming (NLP) solver used in CasADi requires a decision variable array w, containing the inputs, the states and in this case the minimum-time variable ξ as well, structured as:

$$w = [u_0 \quad u_1 \quad ... \quad u_P \quad x_0 \quad x_1 \quad ... \quad x_N \quad \xi]$$

The reason for separating the number of inputs, P, and the number of simulation points, N, is to allow for the possibility to use a lower P since this kept the computational requirements reasonable.

The usual way the inputs are used is to assume that the input is piecewise constant, but with few control points this leads to a non-continuous current trajectory. By instead choosing the input between each control point as the linear interpolation between the points according to equation A, the current trajectory will be continuous. In the equation, k goes from 0 to $N-1$ and p goes from 1 to P during the same time. In all cases in the results section, P was set to 25 and N to 2000, which was enough for all cases tested.

$$u(k) = u_p + (u_{p+1} - u_p)\left(\frac{kP}{N} - p + 1\right) \tag{3.7}$$

The cost variable ξ was added by inserting it in the discretisation phase. A fourth order Runge-Kutta method was used for the integration and the progression of states was defined according to equation 3.8.

$$x_{n+1} = x_n + \xi\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{3.8}$$

The input required for the solver was the cost function, ξ in this case, the decision variable array w, and finally the equality and inequality constraints. Every element of w has a fixed lower and upper bound, setting the minimum and maximum values of the state variables and inputs. The basic constraints for direct multiple shooting are the following set of equality constraints:

$$g(w) = \begin{bmatrix} f(x_0, u_0) - x_1 \\ \vdots \\ f(x_{N-1}, u_{N-1}) - x_N \end{bmatrix} = 0$$

which forces the solution to follow the integrator function, defined as $f(x_k, u_k) = x_{k+1}$.

In order for the system to reach the desired final state, a constraint was added to x_N with the chosen requirements, the most important being the constraint which ensures that the battery reaches the desired SoC. With everything mentioned above the solver was able to converge to a solution, but additional constraints were needed as well in order to end up with a feasible charging curve.

As a final note, it is worth mentioning that CasADi supports something which they call warm start, which is to reuse the previous solution as a starting point, which drastically decreased the computational time after the first solution had been found.

3.3.1 Battery specific constraints

As the ECM does not model most of the states inside a physical battery, choosing constraints for the optimisation was one of the more difficult tasks. The only constraint which was added to a state directly and kept constant in all cases was to limit the maximum temperature that the battery was allowed to reach. Due to the quick decrease in lifespan which occurs when charging at higher temperatures, the upper limit was chosen as 45 $°C$ [19][20]. Another constraint which might be useful would be to limit the maximum input current if a specific charger with a known power output limit is used, especially in cases with high capacity batteries where higher C-rates are unreasonable.

The most important constraint was the one added to v_s, which can be seen as v_d without the voltage from the internal resistance of the battery. The side reaction voltage, η_{sr}, has some relation to the voltage v_s, although not linear. By constraining v_s it was possible to control the resulting charging current and end up with a working result, as the model itself would otherwise just maximise the current with regard to temperature only.

Several different types of constraints for v_s are possible, and in this thesis two different types were tested mainly. The first one was to limit $\ddot{v}_s$, which was calculated as seen in equation 3.9, together with a limit on the final value of v_s.

$$\ddot{v}_s = \frac{v_s(k+2) - 2v_s(k+1) + v_s(k)}{{T_s}^2} \tag{3.9}$$

The hope with this constraint was that it would result in a smooth charging curve, but problems arose as the NLP-solver had big problems with convergence for the first solution, which made it difficult to use. It was also hard to tune and in some cases impossible to achieve a well working result, and even if a result was obtained, conventional charging methods would often result in a shorter charging times.

The second type of limit tested was a simple linear constraint, of which an example can be seen figure 3.12. In the shown example the battery was charged from 20-80% at a starting temperature of 1 $°C$.

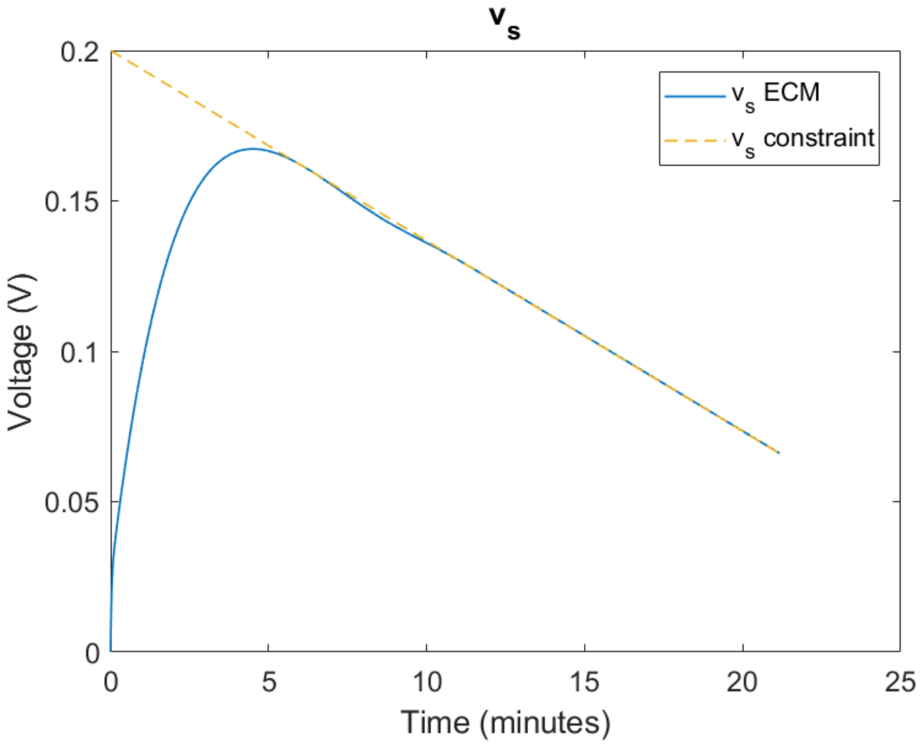


Figure 3.12: Linear constraint applied to v_s during a charge from 20-80% with a starting temperature of 1 $°C$.

The linear constraint was easy to tune with only 2 parameters, a starting value and a final value and there were never any problems with convergence. Because of these reasons it was decided to use a linear constraint for v_s.

Lastly, a constraint on the second derivative of the input current was added in the same way as described in equation 3.9. This was done mostly to get a smoother looking charging curve that would be more realistic to use in practice, as it had minimal effect on the charging time. The effect of the constraint is illustrated in figure 3.13.

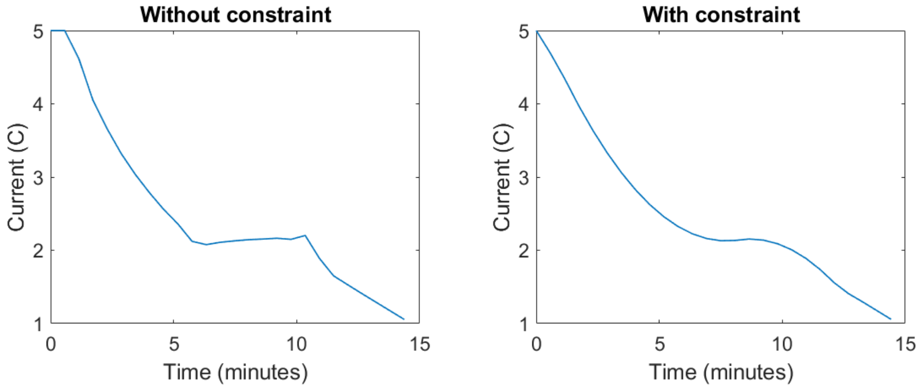


Figure 3.13: Comparison between not having and having a constraint on the the second derivative of the input current.

3.4 Comparison with constant current-constant voltage charging

A comparison between the optimised charging and a commonly used charging method was needed to see how much could be gained over a conventional method. The Constant Current-Constant Voltage (CC-CV) method [21] was chosen and implemented. The principle behind the CC-CV method is to charge the battery with a constant current until a predefined cut-off voltage is reached, after which it switches to a constant voltage. With the constant voltage applied, the current gradually decreases and finally stops when the difference between v_T and the applied voltage reaches 0. In all the evaluated scenarios, the CC part was maximised in regards of temperature and ion concentration in the anode to reach as short charging time as possible to make a fair comparison. The cutoff voltage was chosen, as is standard, as the maximum working voltage of the battery.

There are other methods for charging like MultiStage Constant Current Charging (MSCCC) which is similar to the CC-CV method, but instead of the constant voltage in the end, it uses descending levels of constant currents. In Khan AB et al. [22], the method is shown to be up to 12% better in regards of charging time than the conventional CC-CV method. The MSCCC method is not the only method which decreases the charging time compared to CC-CV but in this thesis the CC-CV method was used for its simplicity.

3.5 Program for fast charging

To systematically get consistent results a MATLAB script was written containing all the different steps required for the optimisation. The flowchart in figure 3.14 shows the basics of how the program work after the initial conditions and constraints have been chosen. By running the script multiple times it converges to a result in 3-4 runs such that the changes in charging current and parameter values are minimal. This program is what was used to generate the results in the thesis.

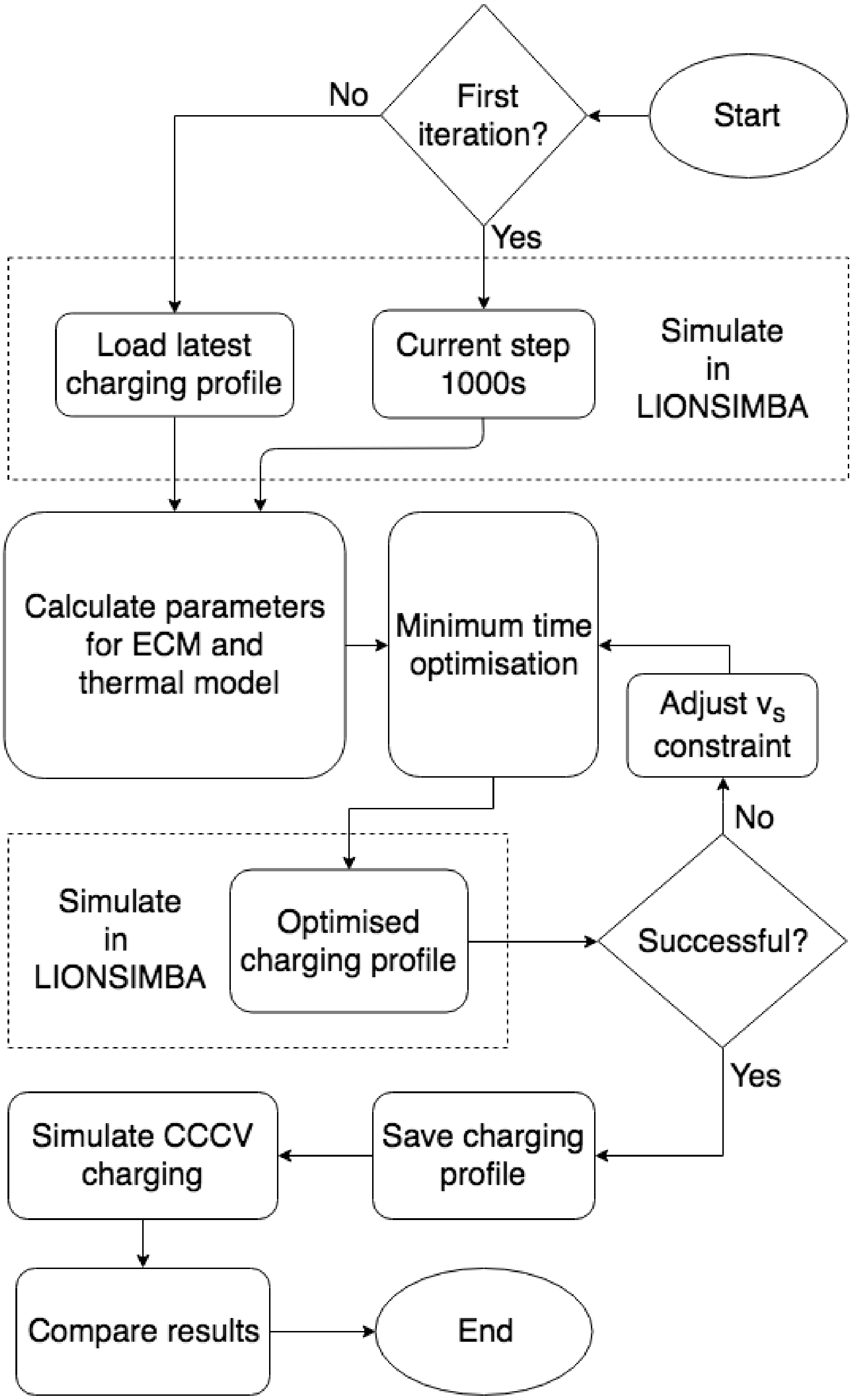


Figure 3.14: Program for optimal fast charging. After 3-4 runs, the change in parameter values and optimised input current is minimal.

4

Results

To evaluate how the optimised fast charging performed compared to the CC-CV method, 6 different scenarios were chosen. Two different SoC intervals, 20-80% and 10-95%, at three different temperatures, 1, 15 and 30 $°C$. The charging interval of 20-80% is one of the most common intervals in the context of fast charging and it was decided that a longer interval was needed as well.

The logic behind the selection of temperatures was to have one with high enough initial temperature such that temperature was the only constraint, one case with the lowest possible temperature and one in between. A summary of the charging times and the improvement over CC-CV can be seen in table 4.1 & 4.2 for 20-80% and 10-95% respectively, also included in the tables are the constraint applied to v_S for each case. The constraint has 2 parameters, initial value and final value, where the final value is in percent of the initial value.

After the 6 different scenarios had been studied it was also investigated how much effect errors in the estimated parameters had on the generated charging current to see how robust the ECM was as a base for optimal charging.

4.1 Charging from 20-80% at different temperatures

The cases shown here are for charging from 20-80% for the three different initial temperatures that were chosen.

4.1.1 Test case 1: 20-80%, 1 $°C$

The first case evaluated was 20-80% at a temperature of 1 $°C$. The resulting charging time was approximately 21 minutes. Figure 4.1 shows the result from the optimisation as well as what happened in the battery when the generated charging current was used as input in LIONSIMBA.

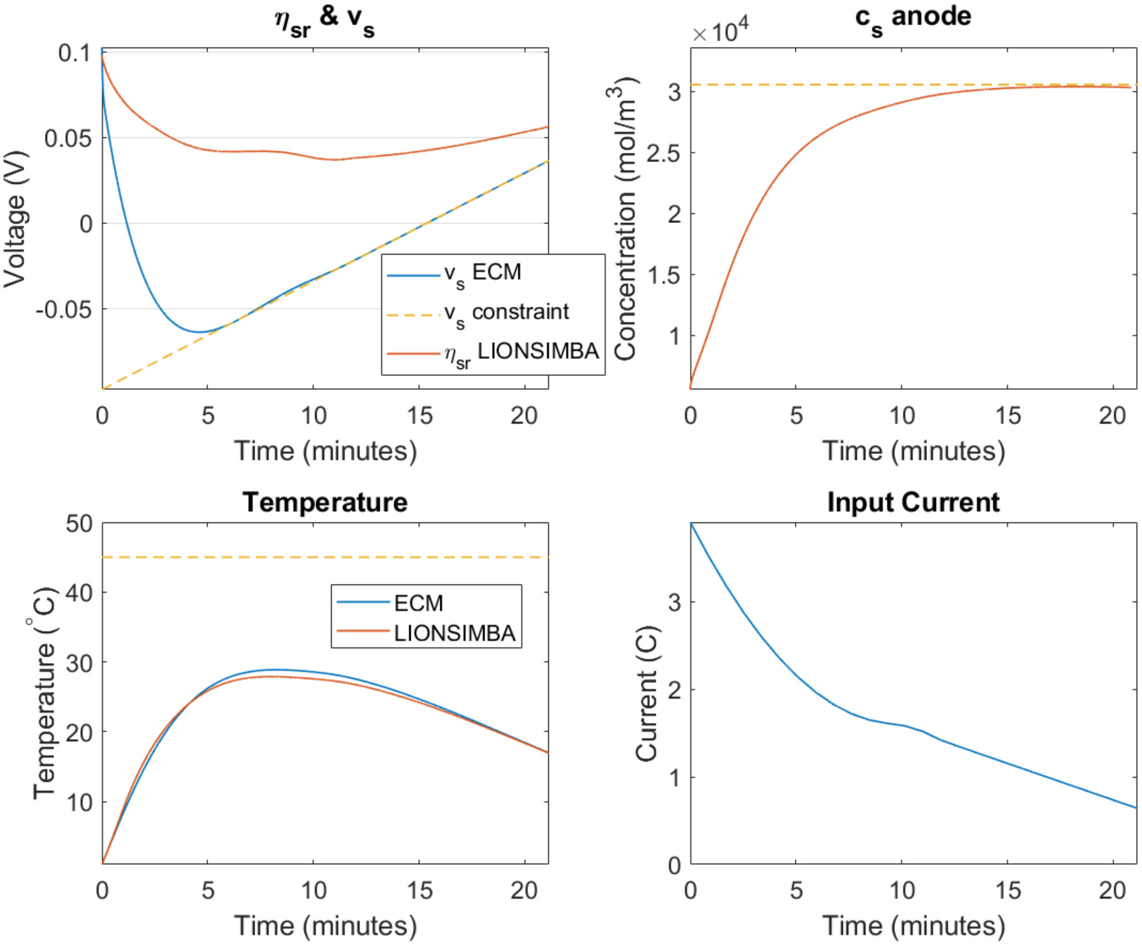


Figure 4.1: Results after an optimised input current calculated for the ECM with CasADi was used as input in LIONSIMBA. In this case charging from 20-80% at an initial temperature of 1 $°C$.

To start with, the top left plot shows η_{sr}, v_s from the ECM as well as the constraint used. The side reaction voltage in red was never close to reaching a value below zero which was the most important aspect. To simplify the comparison of v_s and η_{sr}, v_s was negated and given the same initial value as η_{sr}. The top right plot shows the main limiting factor in terms of charging rate in most cases, c_s, the ion concentration in the solid phase in the anode. The limit, as defined by default in LIONSIMBA, is plotted as a dashed line. In the bottom left plot the temperature is shown. The modelled temperature was fairly accurate for the most part but had some problems during the middle, although it was still within ±5%. Lastly, the current which was the result of the minimum time optimisation, is shown in the bottom right corner. Interesting to note is that the form of the charging current was similar in appearance to what was obtained by R. Methekar et al.[23].

Next, the optimised charging was compared to the more traditional CC-CV charging. The CC-CV charging was manually optimised, in this scenario and the other 5, by setting the constant current rate to the maximum value which did not make the charging exceed the maximum concentration in the solid phase or the maximum temperature. The fast charging was approximately 38% faster in this case. Figure 4.2 shows plots of the same parameters as in figure 4.1, but instead of comparing the ECM and LIONSIMBA, it compares the fast charging and CC-CV charging.

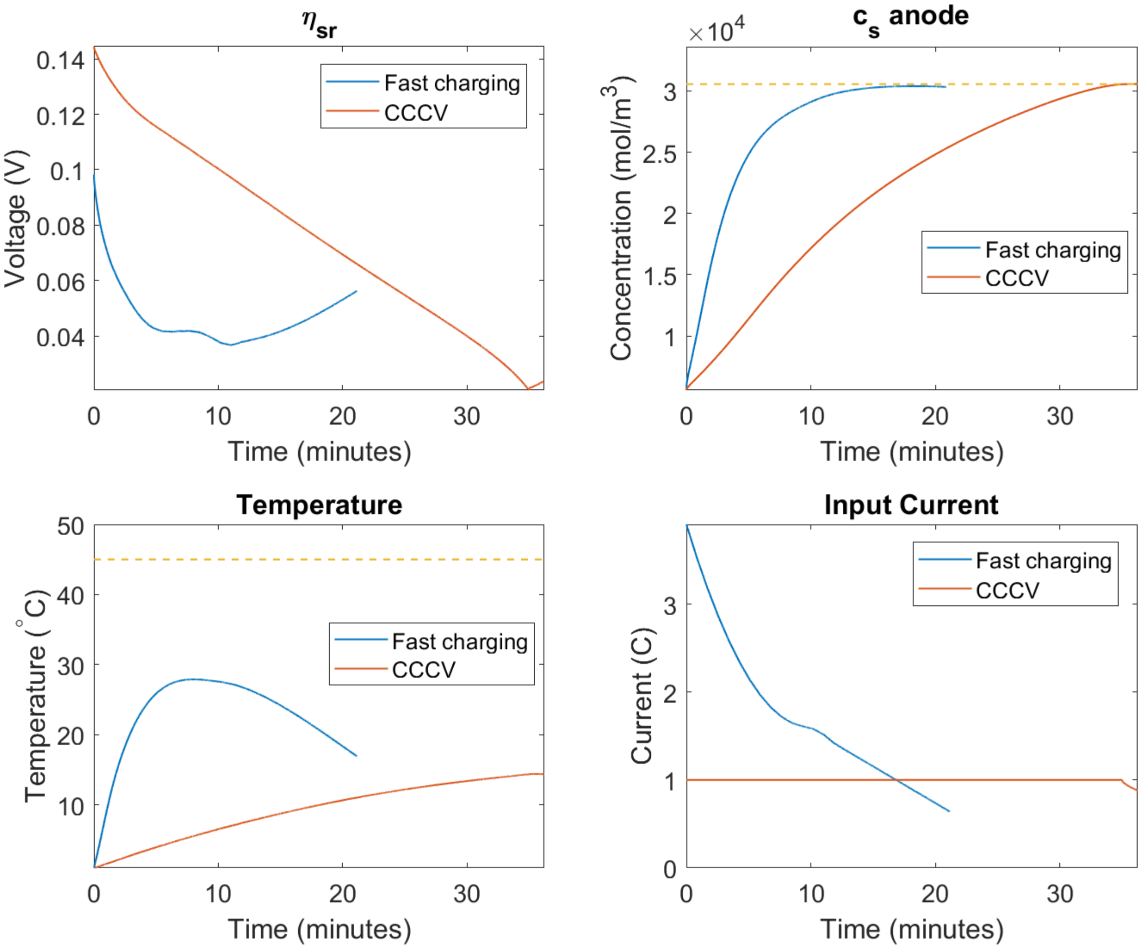


Figure 4.2: Comparison between the optimal charging and the optimised CC-CV method, simulated in LIONSIMBA, charging 20-80% at an initial temperature of 1 $°C$.

As the plots show, the different charging strategies produced very different results. When looking at the currents it can be seen that the CC-CV method barley reaches the CV stage due to only charging to 80% and thus mostly charges with a constant current. The temperature, unavoidably since the charging was faster, reached a higher value with the optimised current, but what is interesting to note is that the side reaction voltage had its lowest voltage at the end with CC-CV as opposed to fast charging which had its minimum somewhere around the middle and never reached a voltage that low.

4.1.2 Test case 2: 20-80%, 15 $°C$

In the second case, the starting temperature was instead set to 15 $°C$. The charging time was drastically improved over charging at 1 $°C$, and the resulting charging time, almost cut in half, was around 14 minutes. Figure 4.3 shows how the battery behaved in simulation.

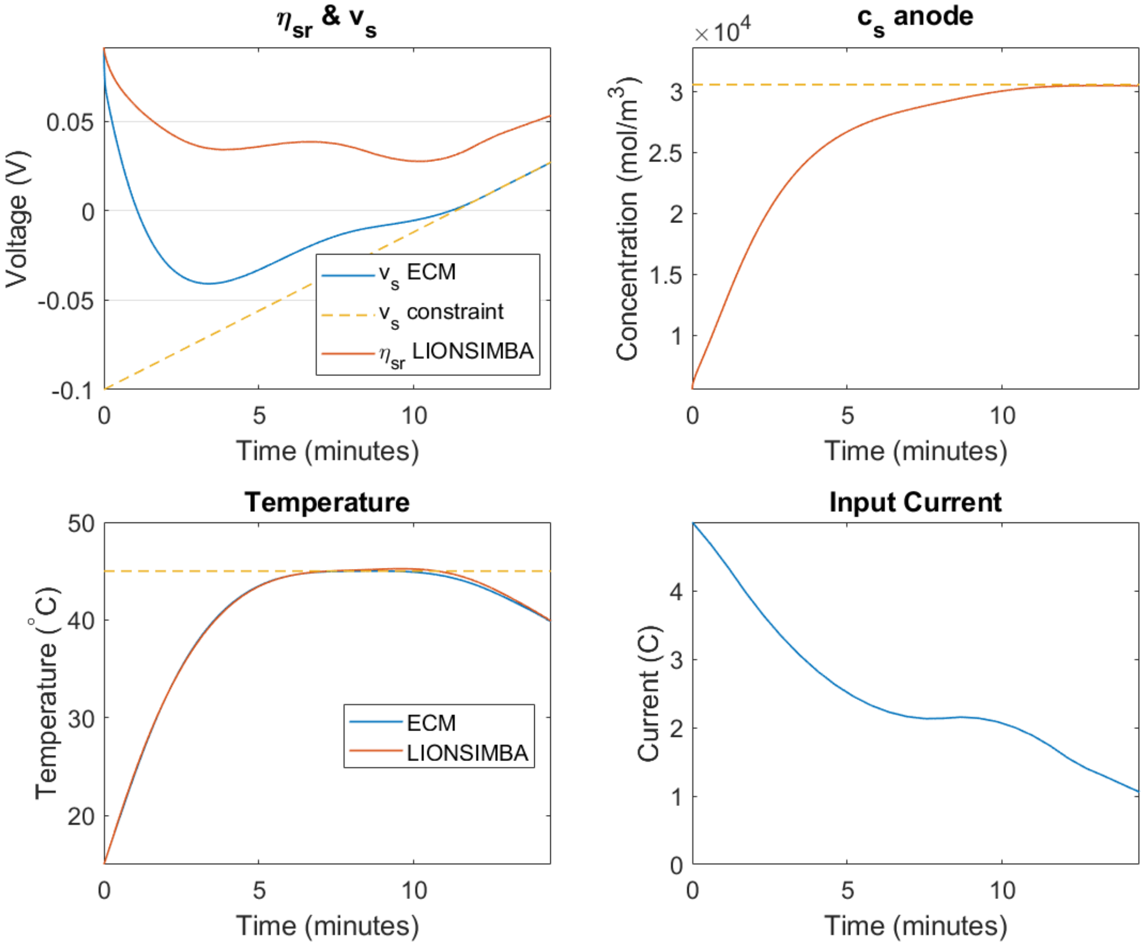


Figure 4.3: Results after an optimised input current calculated for the ECM with CasADi was used as input in LIONSIMBA. In this case charging from 20-80% at an initial temperature of 15 $°C$.

The result was similar to the the previous case, with the main difference being that temperature also became a limiting factor in the optimisation, and the constraint on v_s was only needed at the end to control the ion concentration in the solid phase. The side reaction voltage reached a slightly lower value but was still well above 0. When comparing it to CC-CV charging the difference was smaller than before, and the fast charging was approximately 22% faster. Figure 4.4 shows how they compared in simulation.

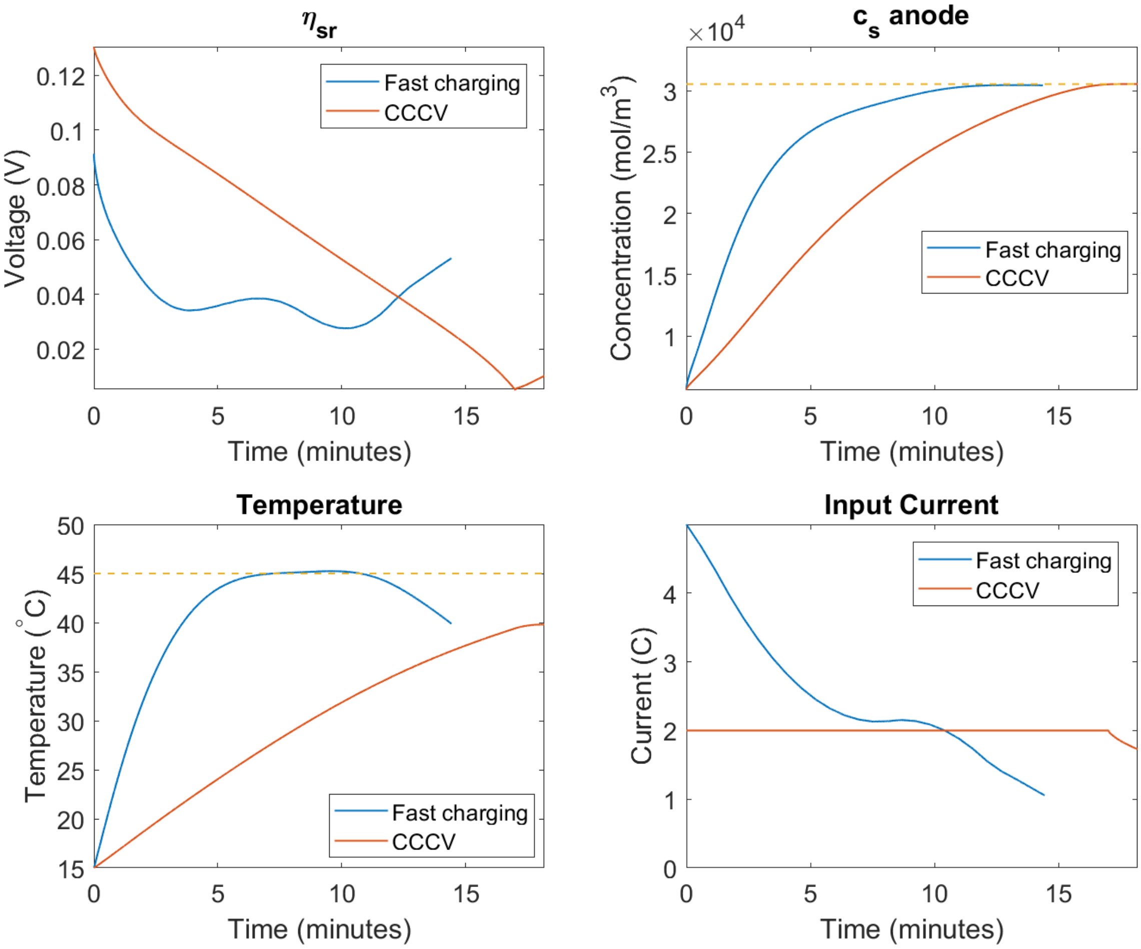


Figure 4.4: Comparison between the optimal charging and the optimised CC-CV method, simulated in LIONSIMBA, charging 20-80% at an initial temperature of 15 °C.

4.1.3 Test case 3: 20-80%, 30 $°C$

In this case, no limit was applied to v_s as the only limiting factor was the temperature. The charging time in this case was around 18 minutes and the results from the simulation can be seen in figure 4.5.

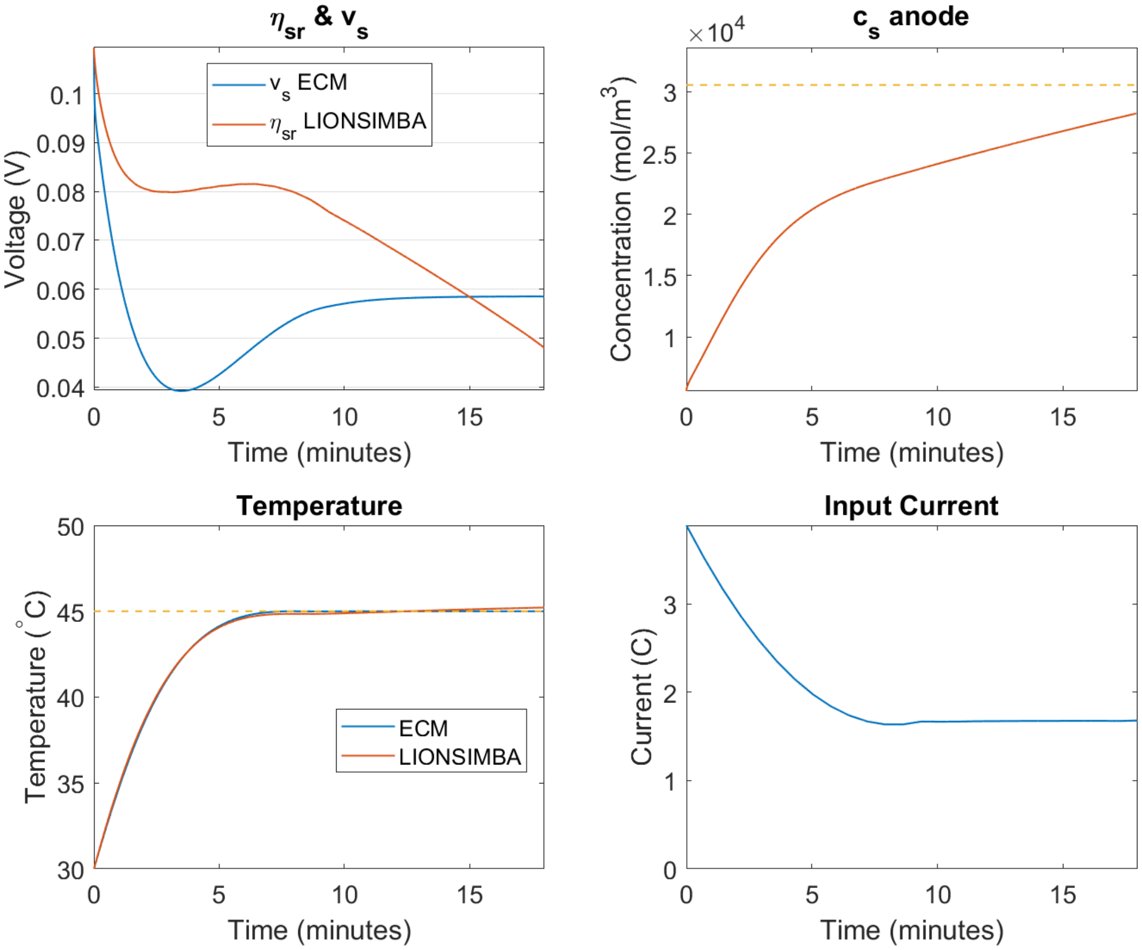


Figure 4.5: Results after an optimised input current calculated for the ECM with CasADi was used as input in LIONSIMBA. In this case charging from 20-80% at an initial temperature of 30 $°C$.

Figure 4.6 shows the comparison to CC-CV and in this case the constant current was maximised so that the battery reached approximately the same maximum temperature with both charging methods. The fast charging was 14% faster in this case. Both charging methods reached approximately the same η_{sr} as well as ion concentration but the average temperature was lower with CC-CV charging which might be healthier for the battery over time.

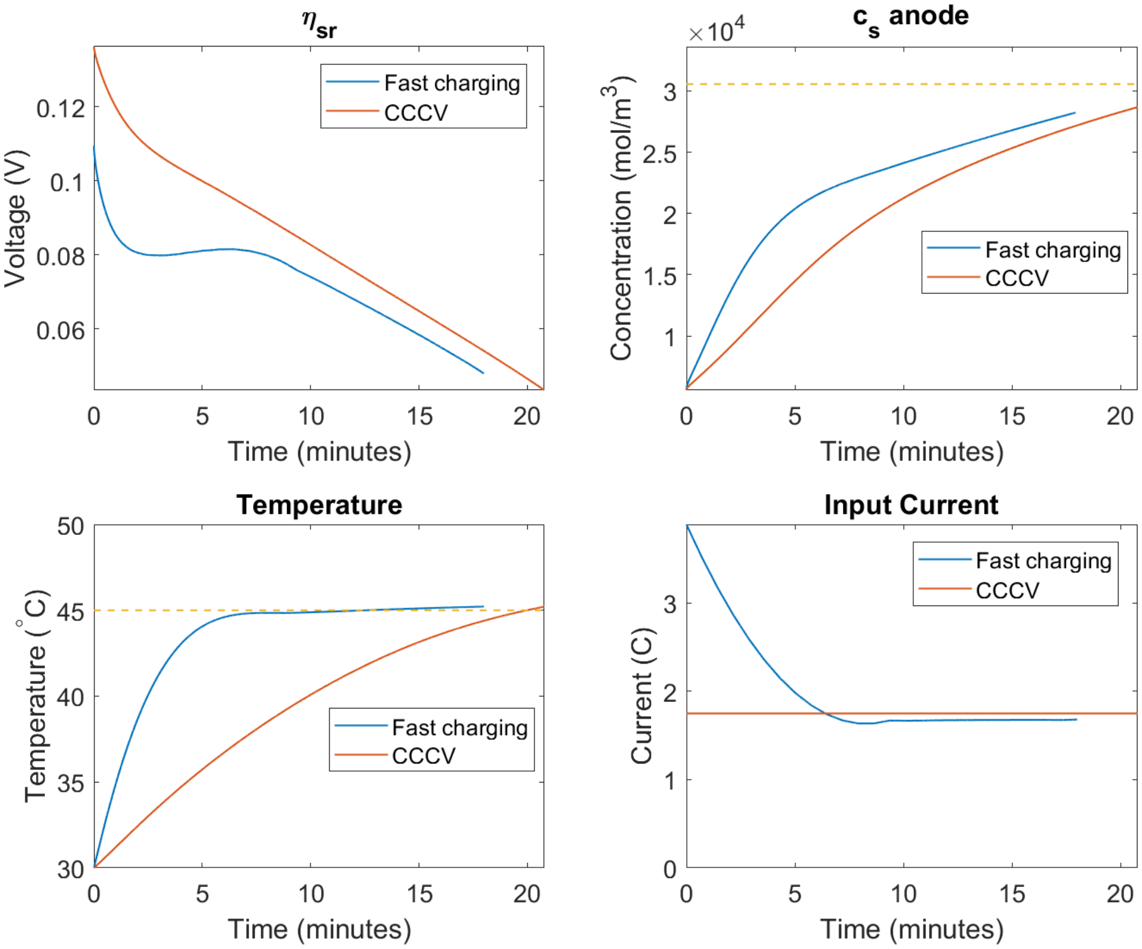


Figure 4.6: Comparison between the optimal charging and the optimised CC-CV method, simulated in LIONSIMBA, charging 20-80% at an initial temperature of 30 °C.

4.2 Charging from 10-95% at different temperatures

As the results were similar at 10-95%, graphs showing the ECM and CC-CV comparisons can be seen in appendix B, except for one case which is shown below. The overall improvement in charging times can be seen in table 4.2.

4.2.1 Test case 6: 10-95%, 30 $°C$

This case was interesting as it indicated that the ECM might not be a good model at higher temperatures. As before with high initial temperature, the limit on v_s was removed and temperature was the only constraint. Figure 4.7 shows the result from the simulation as before.

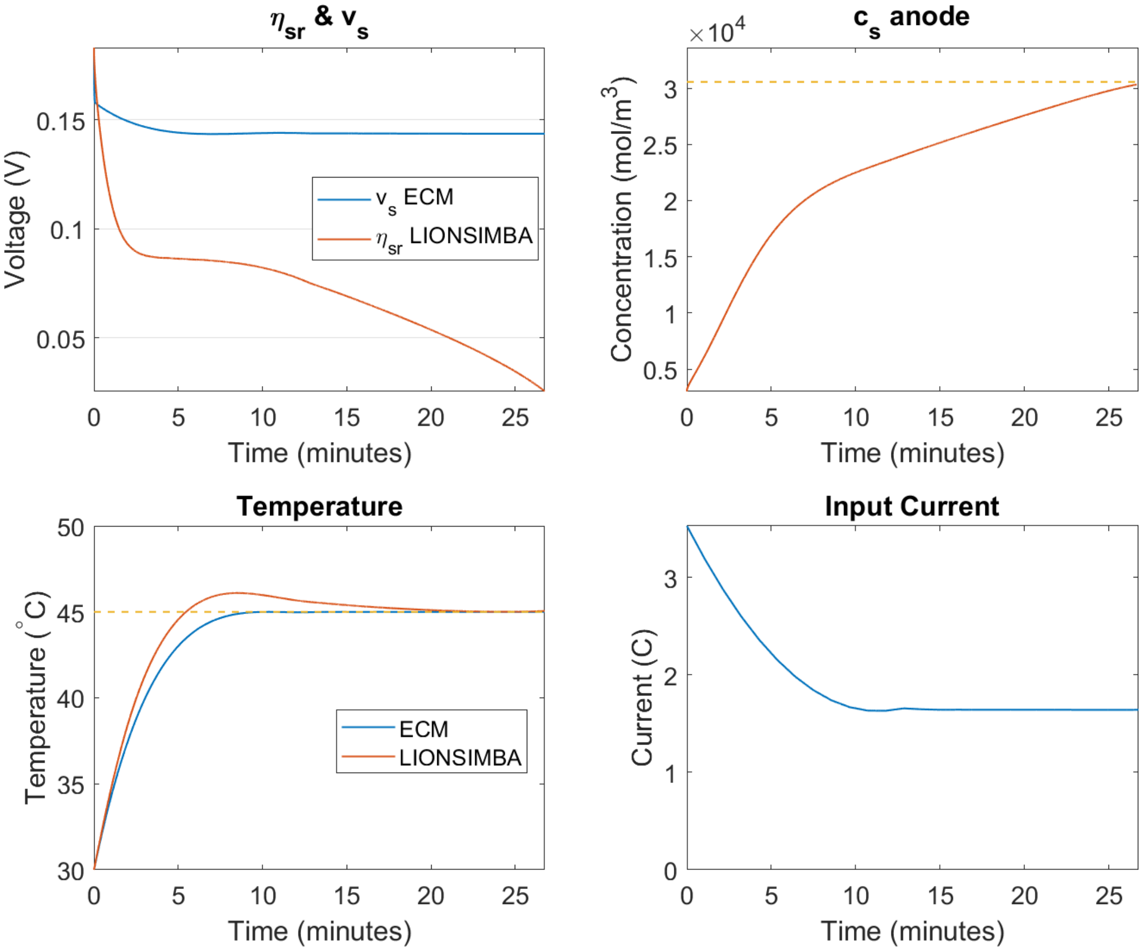


Figure 4.7: Results after an optimised input current calculated for the ECM with CasADi was used as input in LIONSIMBA. In this case charging from 10-95% at an initial temperature of 30 $°C$.

The accuracy of the temperature model was worse in this case compared to the previous ones and underestimates the temperature. The temperature model is directly dependant on v_d, and a comparison of v_d for the ECM and LIONSIMBA can be seen in figure 4.8.

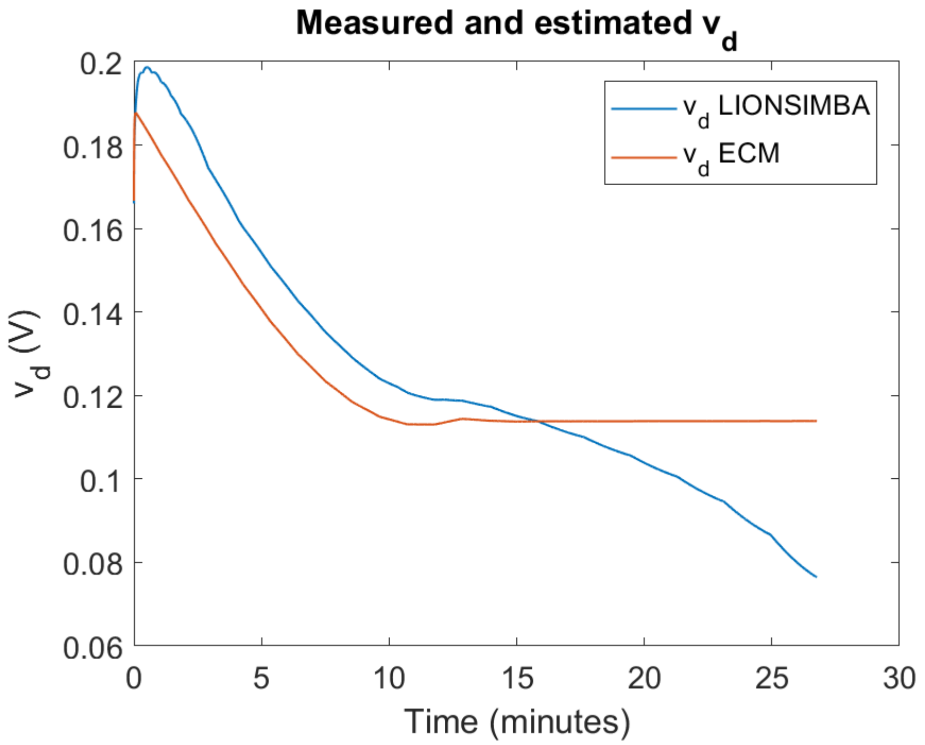


Figure 4.8: Comparison of v_d for the ECM and LIONSIMBA. When the current is constant the ECM reaches an equilibrium while the voltage keeps decreasing in LIONSIMBA.

As the figure shows, the behaviour of the two models was fairly different in this case. As the current was constant for the latter part of the charging, v_d in the ECM reached a constant value. Because of how it was modelled it reached an equilibrium where $\frac{v_1}{R_1C_1} = \frac{i}{C_1}$ and $\frac{v_2}{R_2C_2} = \frac{i}{C_2}$. The problem was that the voltage continued to decrease in LIONSIMBA which was why the model fit badly since this behaviour could not be modelled with the ECM used here. A similar behaviour could be seen when charging from 20-80% at the same initial temperature as well, although not as extreme.

Finally, the fast charging is compared to CC-CV charging in figure 4.9. When the battery was charged to 95% the CC-CV method entered the CV stage for a longer duration of time as can be seen in the bottom right plot. This was also the case where the improvement over CC-CV was smallest, only about 7%.

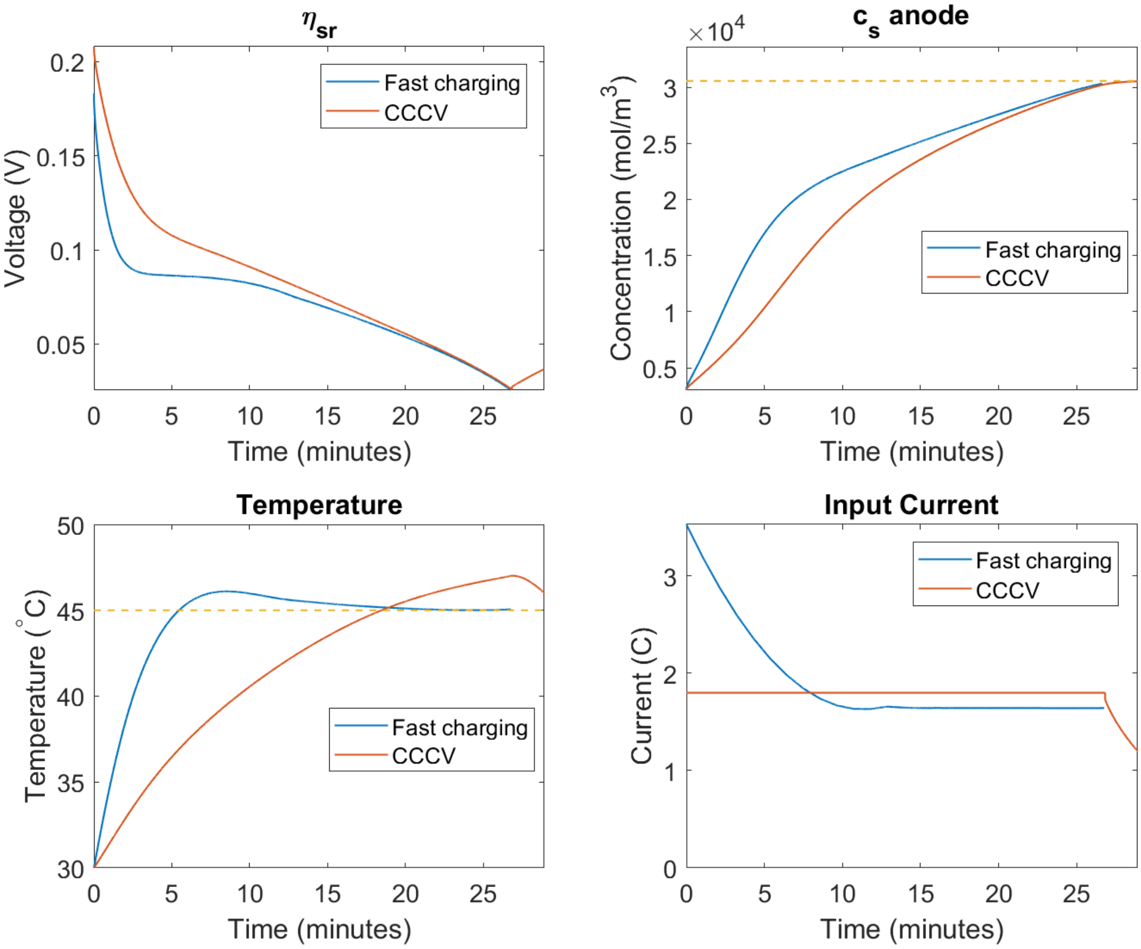


Figure 4.9: Comparison between the optimal charging and the optimised CC-CV method, simulated in LIONSIMBA, charging 10-95% at an initial temperature of 30 $°C$.

Table 4.1: Charging times in minutes & constraints used on v_s, 20-80%

Temp ($°C$)	CC-CV	Optimised	Improvement	v_s^{max} (V)	v_s^{min}/v_s^{max}
1	36	22	38%	0.20	33%
15	18	14	22%	0.19	33%
30	21	18	14%	-	-

Table 4.2: Charging times in minutes & constraints used on v_s, 10-95%

Temp ($°C$)	CC-CV	Optimised	Improvement	v_s^{max} (V)	v_s^{min}/v_s^{max}
1	86	50	42%	0.14	25%
15	37	26	30%	0.12	25%
30	29	27	7%	-	-

4.3 Parameter errors

The last thing investigated was how much the optimised current was affected by errors in the estimated parameter values. Only one case was chosen for testing, 10-95% at 15 $°C$. By allowing all 5 parameters in the ECM to have three values, 80%, 100% and 120% of the original value, 243 different combinations of values were generated. By performing the optimisation in the same way, with the same constraints for all cases, it was possible to see how the result changed and which cases resulted in the biggest changes in charging time. Figure 4.10 shows the extreme cases, the slowest, the fastest and the one generated with unchanged values. The shortest time was achieved when all parameters were at 80% and the slowest when all parameters was at 120% except C_1 which was at 80%. Several other tolerances were tested as well, ±10%, ±30%, ±40% and ±50% and in all cases the charging time changed with the same percentages in the worst cases as the tolerance applied to the parameters.

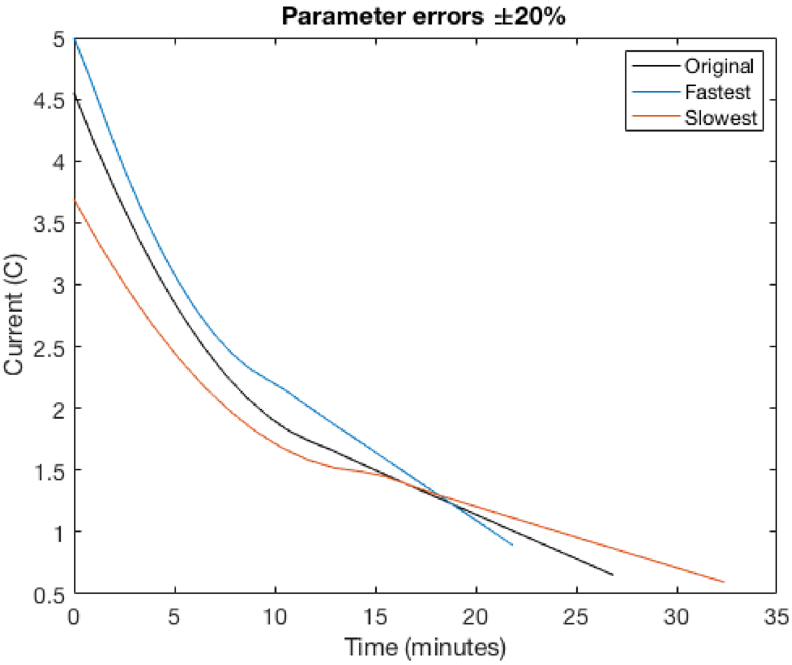


Figure 4.10: ECM parameter errors of ±20% on each parameter, resulting in 243 possible combinations. Seen here is the unchanged, the fastest and the slowest combination.

5
Discussion

There are many parts of this project where there are no definite answers for what method is correct. Because of this there are many areas where improvements are possible through further testing and other changes. In this chapter the results will be discussed as well as where the improvements can be made and what they might look like.

5.1 Results

In terms of results, it is clear that there are better alternatives than the traditional CC-CV charging method. In most cases the ECM worked well for estimating temperature, SoC and v_d with fairly consistent performance except at high temperatures. In terms of optimisation the main problem was at the endpoints of SoC where it was significantly harder to optimise and choose constraints, but as it is rare that you need to fast charge from 0-100%, it was deemed to be a small problem.

The charging times were shortest with 15 $°C$ as initial temperature and the change in charging time compared to the cases starting at 1 $°C$ was very large. If the same charging current is used at different temperatures it has been shown that the lifetime of the battery is severely decreased when charging at a low temperatures [24]. This is probably the reason why the charging has to be so much slower to still keep within the same constraints.

As can be seen in the tables showing the overall results, the improvement over CC-CV decreases as the initial temperature increases. The reason for this is likely that the benefit of starting with a rested battery with a low initial ion concentration can not be taken advantage of to the same extent when the temperature also limits the optimisation while the CC-CV method is mostly unaffected by the temperature constraint.

In most cases it was easy to find which limit to choose for v_s and it was interesting to see that the constraint for each case within the same SoC interval was almost the same, indicating the constraint choice could be mostly based on the SoC interval. The constraint in each case was found through trial and error to maximise the ion concentration but if a more complex constraint would be used, some kind of optimisation method for finding it could be developed. One problem that will arise if this is to be applied to physical battery is that it will be difficult to analyse what happens inside the battery and it will therefore be hard to tune the constraints.

5.2 Static vs. polynomial parameters

For the parameter estimation it was interesting to see that the much simpler method with constant parameters performed better than the more complex one. The main reason for this is probably that the battery behaves differently when it has been charged for some time compared to when it has been unused. The data for fitting the polynomials was at all points estimated when the battery was initialised as rested, which is probably why it worked poorly, especially in the latter half of the charging, but at the same time it would be hard to come up with a consistent method to reach each point so that battery would not be rested when the parameters were to be estimated.

5.3 LIONSIMBA

The main problem with LIONSIMBA was simulating the battery at low temperatures, which was where the largest improvements in charging time seemed possible according to the results. The reason why 1 $°C$ was selected as the lowest temperature was only because LIONSIMBA could not simulate high input currents at lower temperatures as it lead to behaviours which could not happen in a real battery, such as temperature increases of 30 $°C$ which only lasted a few seconds. Worth mentioning as well is that there exists two different temperature models in LIONSIMBA of different complexity. In this thesis the simpler version was used as the more complex one had even more trouble at low temperatures.

An interesting addition would be if another type of battery was added. The type which is simulated now is a $LiCoO_2$ battery which is primarily used in electronic devices such as phones and laptops, and it would be interesting to see how a battery which is used in electric vehicles would behave. Perhaps η_{sr} would become the limiting factor instead of the ion concentration.

5.4 Future work

One of the more interesting aspects of charging which was not investigated in this thesis is degradation of the battery after multiple cycles of charging and discharging. A function for simulating this exists in LIONSIMBA, although it is still being developed and not completely ready for use yet. After multiple cycles of charging a battery, the overall impedance of the battery will increase and one of the main contributors to this is the increased resistance of the SEI film, as seen in figure 2.2. It is not clear how much resistance increase fast charging causes which makes it difficult to account for, and is therefore in need of further research.

It would be very interesting to cycle the optimal charging profile obtained from this thesis on a physical battery and evaluate the degradation and overall performance. Real world testing overall would be interesting to evaluate how well the method actually works and if not, how it may be altered so that it does.

5.4.1 ECM

When it comes to the ECM part of the battery model, more RC-pairs can be added to increase the accuracy of v_d. The main problem, even with improved accuracy, is that the voltage v_s has no linear connection to η_{sr} even if v_d is modelled perfectly. Another important state that would be useful to estimate for optimisation and such is the ion concentration, which was the main limiting factor for charging time in this thesis. As the results indicated that the ECM was less accurate at higher temperatures as well as close to the minimum and maximum SoC that could be interesting to research further as well, it might be possible to replicate the behaviour in some way by adding more components to the model.

5.4.2 Temperature model

The temperature model coincides fairly well with the model in LIONSIMBA, but a physical battery, especially if some kind of battery pack is used instead of single cell, will probably require a different model. It could also be interesting to investigate if there is an optimal temperature to start charging at, but if so, more specific information regarding the battery pack is needed. Another possible scenario is if there exists a cooling and/or heating system, then it could be investigated how that would affect the charging.

5.4.3 Different scenarios

When the battery was charged in this thesis it was always initialised as rested. It was not investigated how the behaviour of the simulated battery cell would change if it was used in some way previous to charging it. This would interesting to test for multiple reasons and regenerative breaking is one scenario that could be simulated to see how it can be controlled to charge the battery safely.

5.4.4 Optimisation

The minimum-time optimisation works well with the current model complexity, and converges to a result in a reasonable amount of time. In regards of optimisation it would be interesting to further investigate the robustness of the solution as well as seeing if a feedback controller could be constructed for real world usage.

Other things which could be investigated further would be to try to find some kind of constraint that would depend only on measurable states instead of time as it is now, then it might be possible to come up with constraints that work in multiple scenarios without needing to be changed for each one. The constraint values which seemed to minimise the charging time seemed to be approximately the same for each SoC interval. It could be investigated if it is possible to choose the initial value for the constraint based on the initial SoC and the final constraint value based on the SoC that it should reach. The problem still remains however that this mapping would be difficult to do for a physical battery.

6
Conclusion

By performing optimisation on an equivalent circuit model of a lithium-ion cell simulated in LIONSIMBA it was possible to achieve charging times that were up to 42% faster than with traditional CC-CV charging while still keeping the battery within the same constraints. By iteratively estimating the model parameters based on the previous charging curve the equivalent circuit model accurately estimated the terminal voltage of the battery simulated in LIONSIMBA for most cases. To control the charging and avoid both lithium plating and overcharging a linearly decreasing constraint was applied to the voltage over the two RC-pairs in the equivalent circuit model. The result clearly shows that the method has potential and that it should be possible to apply it on a physical battery even though it will be more difficult to choose constraints for the optimisation.

Bibliography

[1] M. Armand and J.-M. Tarascon, “Building better batteries,” Nature, vol. 451, pp. 652–657, 2008.

[2] Ramadesigan, V., Northrop, P. W. C., De, S., Santhanagopalan, S., Braatz, R. D., & Subramanian, V. R. (2012). Modeling and Simulation of Lithium-Ion Batteries from a Systems Engineering Perspective. Journal- Electrochemical Society, (3), R31.

[3] Batteryuniversity.com, ’BU-1003: Electric Vehicle (EV)’, 2018. [Online]. Available: https://batteryuniversity.com/learn/article/electric_vehicle_ev. [Accessed: 21-Nov-2018].

[4] Batteryuniversity.com, ’BU-301a: Types of Battery Cells’, 2018. [Online]. https://batteryuniversity.com/learn/article/types_of_battery_cells. [Accessed: 21-Nov-2018].

[5] cleantechnica.com, ’A Tale of 3 Battery Packs’, 2016. [Online]. https://cleantechnica.com/2016/01/06/a-tale-of-3-battery-packs/. [Accessed: 21- Nov- 2018].

[6] Batteryuniversity.com, ’BU-205: Types of Lithium-ion’, 2018. [Online]. https://batteryuniversity.com/learn/article/types_of_lithium_ion. [Accessed: 21-Nov-2018].

[7] Erratum: Modeling of Galvanostatic Charge and Discharge of the Lithium/Polymer/Insertion Cell [J. Electrochem. Soc., 140, 1526 (1993)]. (2018). Journal of the Electrochemical Society, 165(11), X13.

[8] Chaturvedi N, Klein R, Christensen J, Ahmed J, Kojic A. Algorithms for Advanced Battery-Management Systems. (cover story). IEEE Control Systems [serial online]. June 2010;30(3):49. Available from: Complementary Index, Ipswich, MA. Accessed August 30, 2018.

[9] Zhang J, Yang X, Zhang G, Huang Q, Xiao C, Yang C. Investigation on the root cause of the decreased performances in the overcharged lithium iron phosphate battery. International Journal of Energy Research. 2018;(7):2448. doi:10.1002/er.4025.

[10] Qiu, J., Zhang, G., Yang, X., Yang, Y., Wang, Z. Wu, W. (2014). A study on structure-performance relationship of overcharged 18650-size Li.sub.4][Ti.sub.5][O.sub.12]/Li[Mn.sub.2][O.sub.4] battery. Journal of Thermal Analysis and Calorimetry, (3), 1413. https://doi.org/10.1007/s10973-014-4094-7

[11] Torchio M, Magni L, Raimondo D, Gopaluni R, Braatz R. LIONSIMBA: A matlab framework based on a finite volume model suitable for Li-ion battery design, simulation, and control. Journal Of The Electrochemical Society [se-

rial online]. January 1, 2016;163(7):A1192-A1205. Available from: Scopus®, Ipswich, MA. Accessed September 14, 2018.

[12] Subramanian, V. R., Boovaragavan, V., Ramadesigan, V., & Arabandi, M. (2009). Mathematical Model Reformulation for Lithium-Ion Battery Simulations: Galvanostatic Boundary Conditions. JOURNAL- ELECTROCHEMICAL SOCIETY, (4), A260.

[13] Suthar B, Northrop PWC, Rife D, Subramanian VR. Effect of Porosity, Thickness and Tortuosity on Capacity Fade of Anode. Journal of the Electrochemical Society. 2015;162(9):A1708.

[14] http://mocha-java.uccs.edu/ECE5710/ECE5710-Notes02.pdf . Accessed November 1, 2018.

[15] Xinran Tao, Design, Modeling and control of a thermal management system for hybrid electric vehicles, Ph.D Thesis, Clemson University, 2016.

[16] Lin X, Perez HE, Mohan S, Siegel JB, Stefanopoulou AG, Ding Y, et al. A lumped-parameter electro-thermal model for cylindrical batteries. J Power Sources, 2014

[17] Joel Andersson. "A General-Purpose Software Framework for Dynamic Optimization". PhD thesis. Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001- Heverlee, Belgium: Arenberg Doctoral School, KU Leuven, Oct. 2013.

[18] J. Andersson, J. Åkesson and M. Diehl, "Dynamic optimization with CasADi," 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), Maui, HI, 2012, pp. 681-686. doi: 10.1109/CDC.2012.6426534

[19] Leng F, Tan CM, Pecht M. Effect of Temperature on the Aging rate of Li Ion Battery Operating above Room Temperature. SCIENTIFIC REPORTS. 5. doi:10.1038/srep12967.

[20] M. Uitz, M. Sternad, S. Breuer, C. Taubert, T. Traussnig, V. Hennige, I. Hanzu, and M. Wilkening. Aging of Tesla's 18650 Lithium-Ion Cells: Correlating Solid-Electrolyte-Interphase Evolution with Fading in Capacity and Power Journal of the Electrochemical Society 2017 164: A3503-A3510.

[21] Berg Helena. 6.1.1.2 The Constant Current-Constant Voltage Charging Method. Batteries for Electric Vehicles - Materials and Electrochemistry. [S.l.]: Cambridge University Press, 2018.

[22] Khan AB, Choi W. Optimal Charge Pattern for the High-Performance Multistage Constant Current Charge Method for the Li-Ion Batteries. IEEE Transactions on Energy Conversion. 2018;33(3):1132-1140. doi:10.1109/TEC.2018.2801381.

[23] R. Methekar, V. Ramadesigan, R. D. Braatz, and V. R. Subramanian, Optimum Charging Profile for Lithium-ion Batteries to Maximize Energy Storage and Utilization (2010).

[24] Yang X-G, Zhang G, Ge S, Wang C-Y. Fast charging of lithium-ion batteries at all temperatures. Proceedings of the National Academy of Sciences of the United States. 2018;(28):7266. doi:10.1073/pnas.1807115115.

A

Summary of optimisation problem

The problem to be solved was:

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimise}} \quad & \xi \\ \text{subject to} \quad & x_{k+1} = F(x_k, u_k, \xi) \\ & g(w) \leq 0 \\ & x_N = x_f \end{aligned}$$

where

$$w = [u_0 \quad u_1 \quad \ldots \quad u_P \quad x_0 \quad x_1 \quad \ldots \quad x_N \quad \xi]$$

and the input at each step was defined as:

$$u_k = u_p + (u_{p+1} - u_p)\left(\frac{kP}{N} - p + 1\right)$$

where $0 \leq k \leq N$ and $0 \leq p \leq P$. The system dynamics were defined as:

$$\dot{x} = \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \dot{z} \\ \dot{T}_c \end{bmatrix} = \begin{bmatrix} -\frac{v_1}{R_1C_1} + \frac{i}{C_1} \\ -\frac{v_2}{R_2C_2} + \frac{i}{C_2} \\ \frac{i}{3600Q} \\ \frac{T_{amb}-T_c}{R_cC_c} + \frac{Q_{ECM}}{C_c} \end{bmatrix}$$

$$Q_{ECM} = (R_0 i + v_1 + v_2)i$$

and the integrator function, F, based on the dynamics was calculated with a fourth order Runge-Kutta method in CasADi according to the following code with the minimum-time variable ξ added:

```
%Function which defines the system dynamics
f_dyn = Function('f', {x, i}, {xdot});
%Minimum time variable
xi=MX.sym('xi');
% Fixed step Runge-Kutta 4 integrator
M = 4; % RK4 steps per interval
DT = T/N/M;
```

```
X0 = MX.sym('X0', 4);
U = MX.sym('U');
X = X0;
Q = 0;
for j=1:M
   k1 = f_dyn(X, U);
   k2 = f_dyn(X + DT/2 * k1, U);
   k3 = f_dyn(X + DT/2 * k2, U);
   k4 = f_dyn(X + DT * k3, U);
   %Add min-time constant
   X=X + xi*DT/6*(k1 +2*k2 +2*k3 +k4);
end
%Integrator function
Fdyn = Function('F', {X0, U}, {X}, {'x0','i'}, {'xf'});
```

The constraints defined by $g(w)$ were the following, starting with the state constraints:

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_{min} \\ \vdots \\ x_{min} \\ 0.25 \end{bmatrix} \leq w^T \leq \begin{bmatrix} 5C \\ \vdots \\ 5C \\ x_{max} \\ \vdots \\ x_{max} \\ 5 \end{bmatrix}$$

where $x_{min} = [-\infty \quad -\infty \quad 0.05 \quad 0]^T$ and $x_{max} = [\;\infty \quad \infty \quad 1 \quad 45]^T$. The constraint on v_s was added in the following way:

$$\begin{bmatrix} x_0^T \\ \vdots \\ x_N^T \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \leq v_s^{max}(k)$$

where $v_s^{max}(k)$ is the linearly decreasing constraint. A constraint on the second derivative of the input current was added as well:

$$\begin{bmatrix} |\ddot{u}_1| \\ \vdots \\ |\ddot{u}_{P-1}| \end{bmatrix} = \begin{bmatrix} |\frac{u_2 - 2u_1 + u_0}{{T_s}^2}| \\ \vdots \\ |\frac{u_P - 2u_{P-1} + u_{P-2}}{{T_s}^2}| \end{bmatrix} \leq \begin{bmatrix} \ddot{u}_{max} \\ \vdots \\ \ddot{u}_{max} \end{bmatrix}$$

The last two constraints, not contained in g(w), were the constraint which forces the solver to follow the integrator function:

$$\begin{bmatrix} F(x_0, u_0) - x_1 \\ \vdots \\ F(x_{N-1}, u_{N-1}) - x_N \end{bmatrix} = 0$$

and the constraint on the final state which ensured that the battery was charged:

$$[-\infty \quad -\infty \quad z_f \quad 0] \leq x_N^T \leq [\infty \quad \infty \quad z_f \quad 45]$$

After the setup is complete, a warm-start option is available for the solver if a previous solution exists, making it easier and faster for the solver to reach the new solution $w_{optimal}$. This is coded accordingly:

```
%  Create NLP solver
prob = struct('f', J, 'x', w, 'g', vertcat(g{:}));
%If warm start is enabled the previous result is used to improve
%computation time
options = struct;
if(prevSolutionExist)
    options.ipopt.warm_start_init_point = 'yes';
    solver = nlpsol('solver','ipopt',prob,options);
    sol = solver('x0', sol.x, 'lam_g0', sol.lam_g, 'lam_x0', ...
        sol.lam_x, 'lbx', lbw, 'ubx', ubw,'lbg', lbg, 'ubg', ubg);
else
    solver = nlpsol('solver','ipopt',prob,options);
    sol = solver('x0', w0, 'lbx', lbw, 'ubx', ubw,'lbg', lbg, ...
        'ubg', ubg);
end
w_opt = full(sol.x);
```

B

Test case 4 & 5

B.1 Test case 4: 10-95%, 1 $°C$

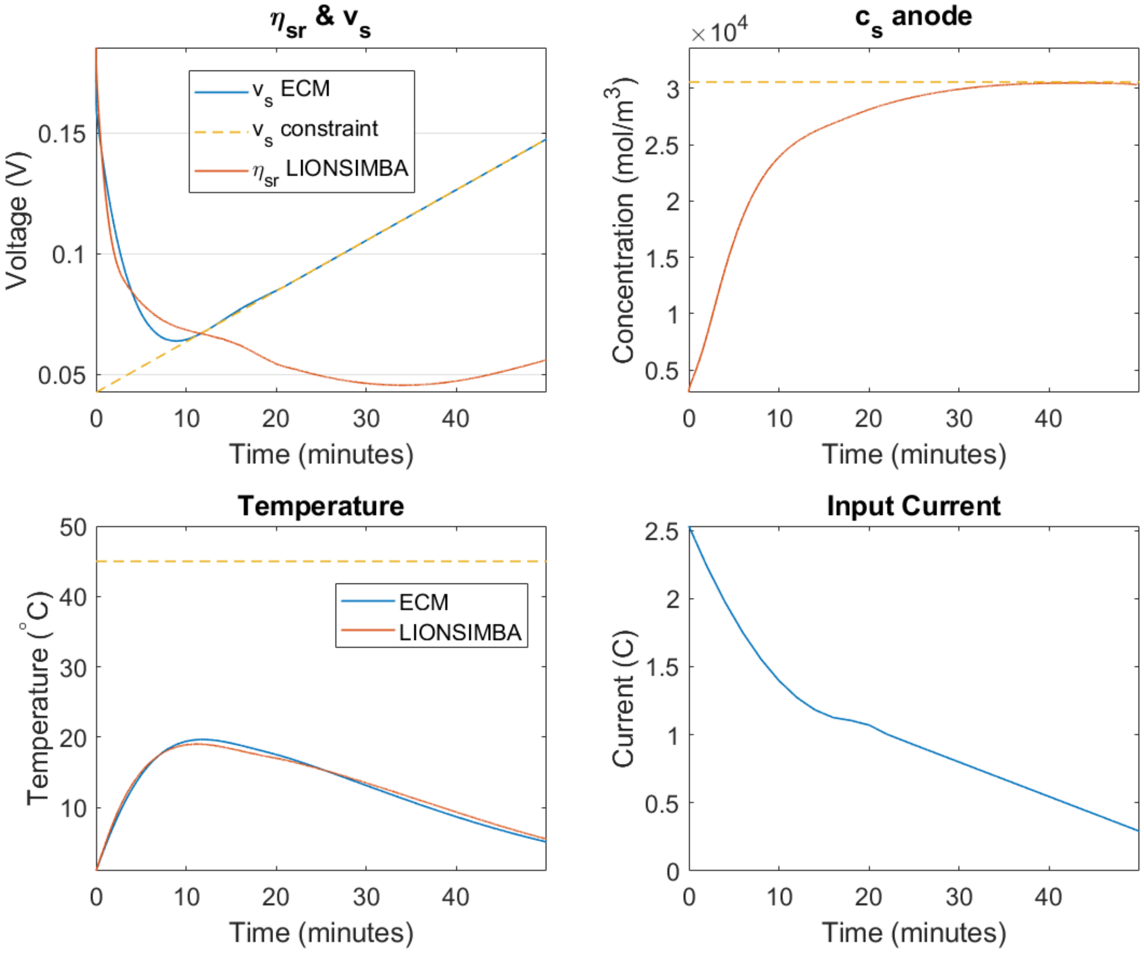


Figure B.1: Results after an optimised current profile is calculated for the ECM with CasADi and then used as input current in LIONSIMBA. In this case charging from 10-95% at an initial temperature of 1 $°C$.

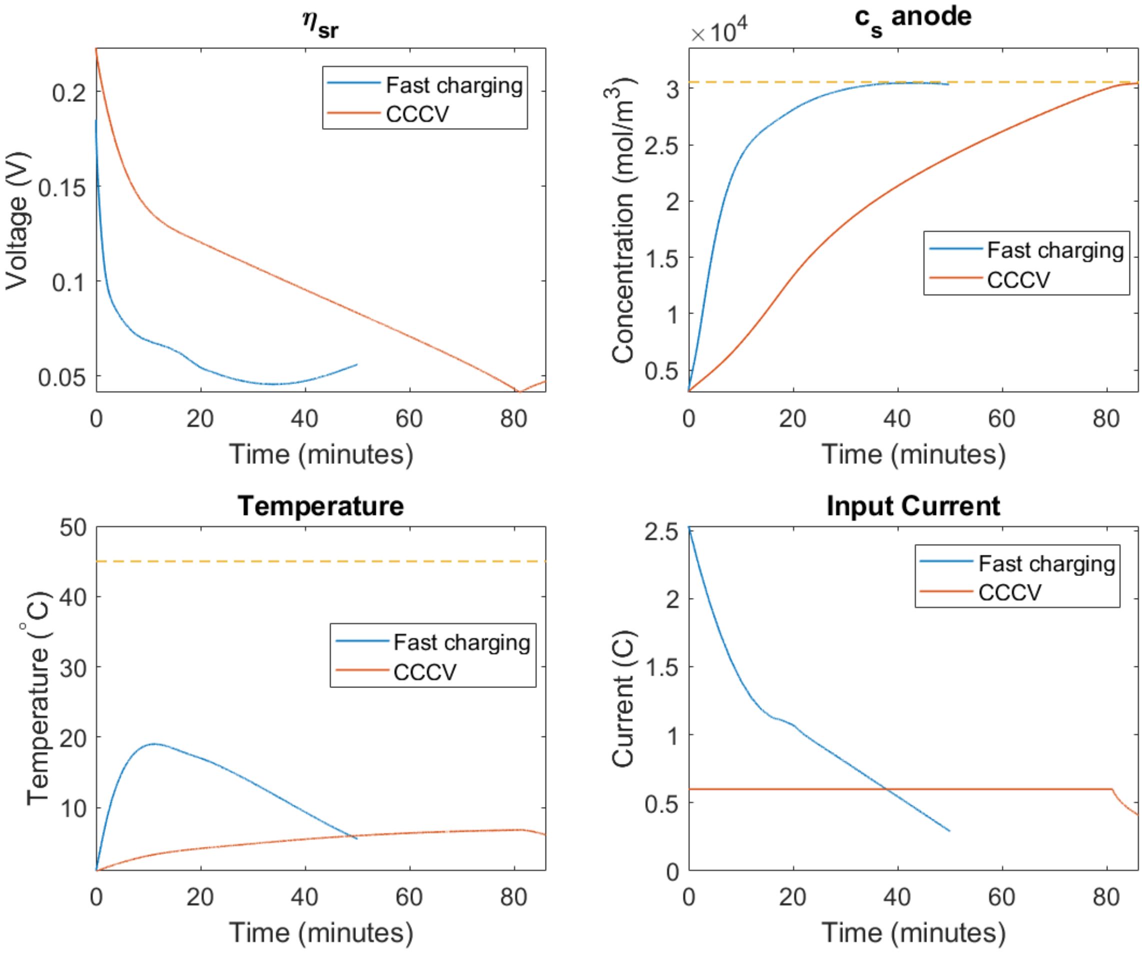


Figure B.2: Comparison between the optimal charging and the optimised CC-CV method, simulated in LIONSIMBA, charging from 10-95% at an initial temperature of 1 $°C$.

B.2 Test case 5: 10-95%, 15 $^\circ C$

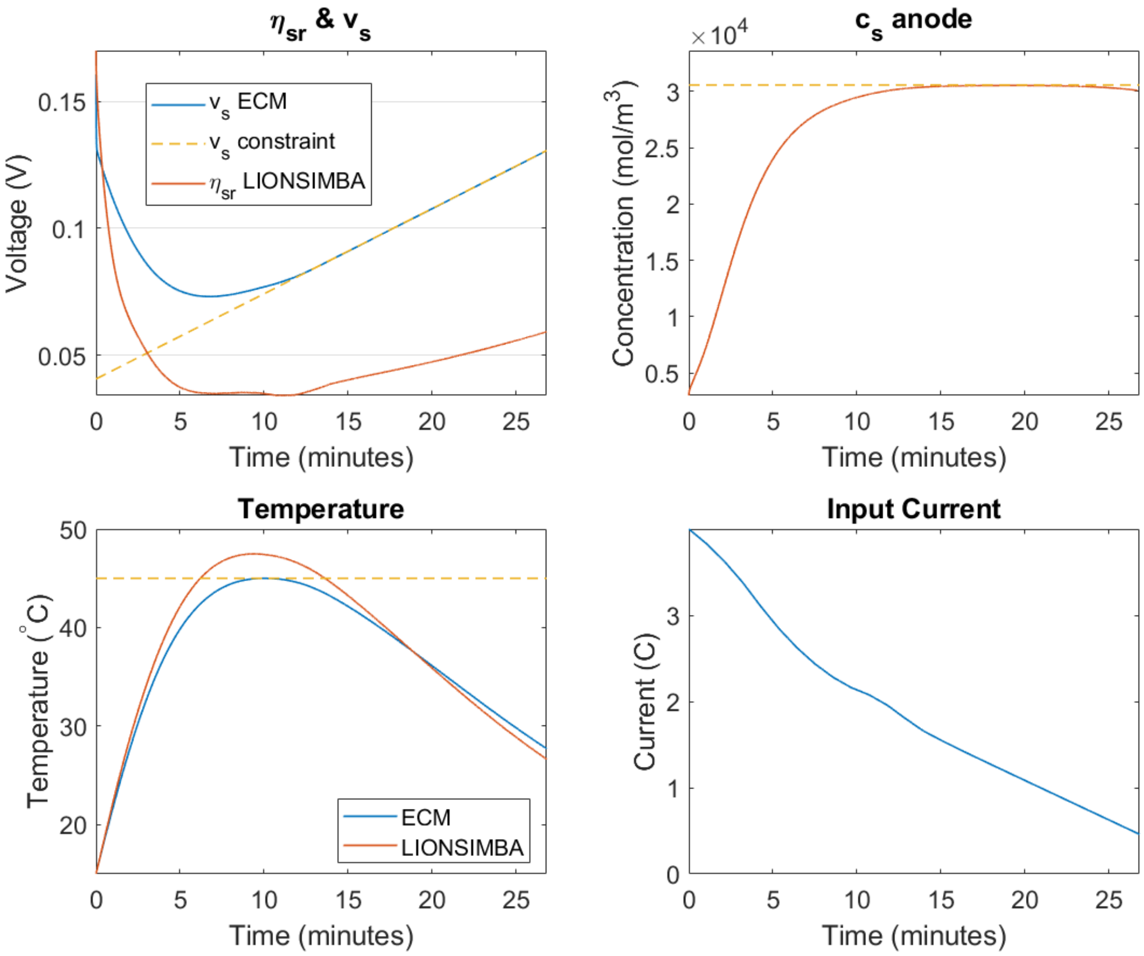


Figure B.3: Results after an optimised current profile is calculated for the ECM with CasADi and then used as input current in LIONSIMBA. In this case charging from 10-95% at an initial temperature of 15 $^\circ C$.

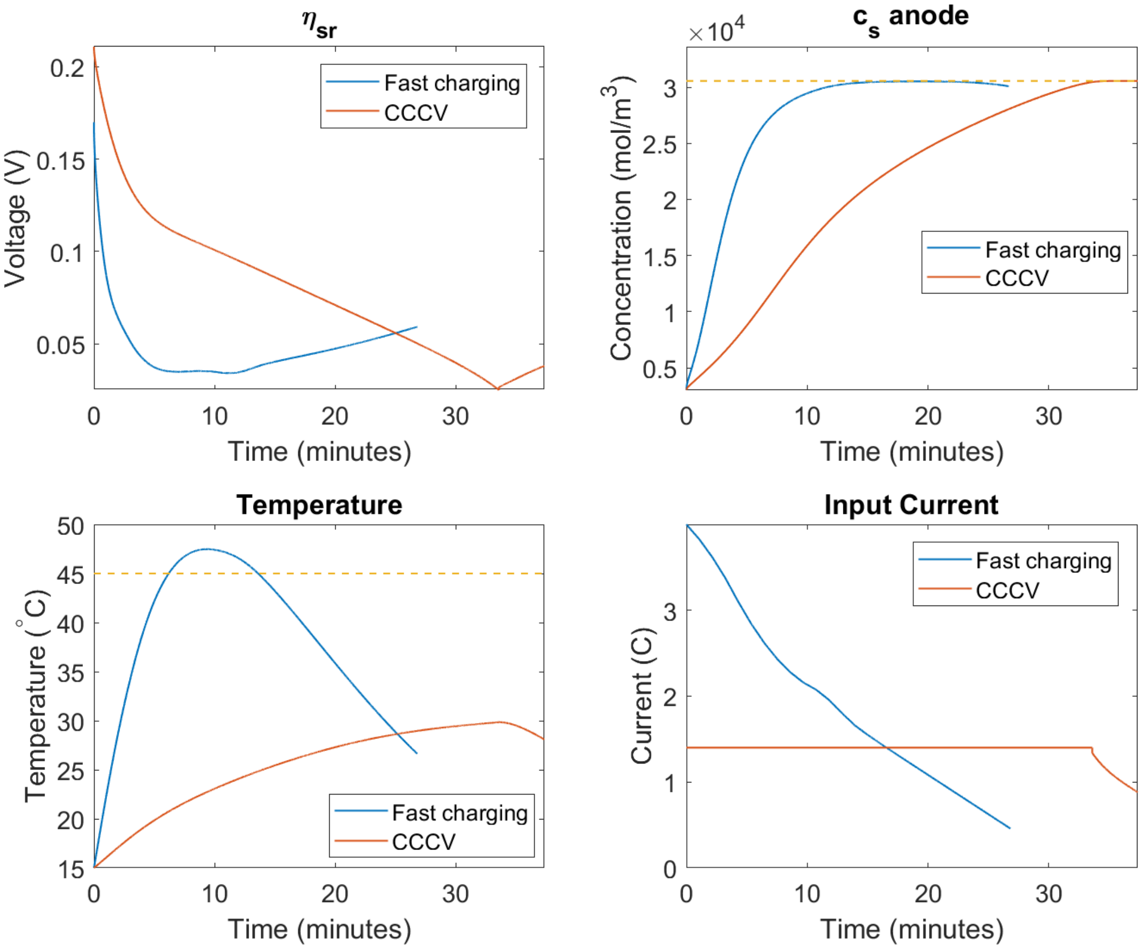


Figure B.4: Comparison between the optimal charging and the optimised CC-CV method, simulated in LIONSIMBA, charging from 10-95% at an initial temperature of 15 °C.