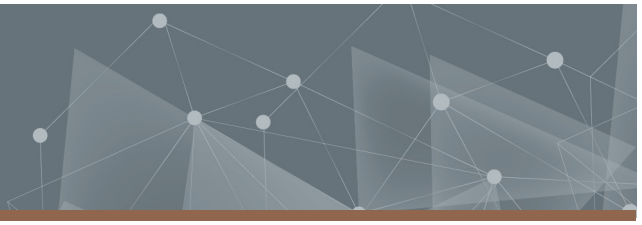
CHALMERS
UNIVERSITY OF TECHNOLOGY



Computationally Efficient Real Time Embedded Computer Battery Simulator

Master's thesis in Complex Adaptive Systems

John Sandgren

DEPARTMENT OF ELECTRICAL ENGINEERING

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

Master's thesis 2022

Computationally Efficient Real Time Embedded Computer Battery Simulator

JOHN SANDGREN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2022

Computationally Efficient Real Time Embedded Computer Battery Simulator
JOHN SANDGREN

Supervisor: Hans-Klaus Weiler, Volvo Cars Torslanda
Examiner: Jimmy Ehnberg, Electrical engineering

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

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2022

Computationally Efficent Real Time Embedded Computer Battery Simulator
JOHN SANDGREN
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Automotive vehicles are a key part of how modern society meets many of the transport related challenges it faces, but this is problematic as a vast majority of the vehicles in use today are driven by combustion engines that produce pollutants that affects our planet in many ways. One of the solutions that the automotive industry is moving towards are electrified vehicles (EVs) that are propelled entirely through electricity stored onboard the vehicles in the form of batteries. Lithium-Ion (Li-Ion) batteries are the most prevalent in modern battery electric vehicles (BEVs), which presents a new set of engineering tasks to solve, as the Li-Ion battery chemistry can become volatile due to many factors such as high temperatures or abusive usage. To ensure safe and optimal operation of the battery, BEVs use control systems that focus solely on the battery, called the battery management system (BMS).

This project is a collaboration between Chalmers University of Technology and Volvo Cars Torslanda, with the goal of developing a battery simulator that would make the testing and further development of Volvos' BMS easier and less time consuming. The model was developed to run on a Raspberry Pi embedded computer and for this purpose the Python programming language was used. The chosen battery model was an equivalent circuit model (ECM), which describes the battery as a circuit consisting of a resistors coupled in series with one or two resistance-capacitance (RC) circuits, depending on the order of the model. To simulate a battery, circuit parameters were extracted from Hybrid Pulse Power Characterization (HPPC) tests for different temperatures and States of Charge (SoC). The accuracy was measured in relation to avaiable test data while the performance was measured by performing simulations on a Raspberry Pi 4B.

Both the first and second order models show promise in the voltage output domain, achieving better accuracy in higher temperature ranges. The simple implemented temperature model was inaccurate for all tests, having large errors at all temperatures except for at 25°C. The performance difference between the first- and second-order models was about 5.5%, with both models being able to achieve real time updates on the Raspberry Pi4B.

From the obtained results, the conclusion drawn was that for development purposes, the first order model should be used as the accuracy gain was minimal in relation to the lost performance. It was also concluded that more extensive HPPC tests will lead to a much more performant model. The temperature modeling should also be split into a separate program, to allow for more complex thermal modeling.

Keywords: battery, BMS, ECM, simulation, HPPC.

Acknowledgements

I would like to begin with thanking Koushik Damodara Shenoy, as we started writing this thesis together before it was split into two projects! He helped me give the theoretical background I would need to complete to project.
I would also like to extend major thanks to my Chalmers supervisor and examiner Jimmy Ehnberg, whom gave me advice in both how to plan and handle all the work related to writing this thesis.
Similarly, Hans-Klaus Weiler at Volvo has been instrumental in giving a helping hand in how to work with larger projects, will also giving insight into a world outside of academics.
Finally, I would like to thank my friends, family and especially partner which have helped and supported me through out this process. It would not have been possible without you.

John Sandgren, Gothenburg, June 2022

List of Acronyms

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

CAN	Controller Area Network
CPU	Central Processing Unit
ECM	Equivalent Circuit Method
EV	Electrified Vehicles
HPPC	Hybrid Pulse Power Characterization
LSM	Least Squares Method
OCV	Open Circuit Voltage
OOP	Object Oriented Programming
OS	Operating System
SoC	State of Charge
SSH	Secure Shell Protocol
RC	Resistance-Capacitor
RSS	Residual Square Sum

Contents

List of Figures

List of Tables

1
Introduction

The prevalence of automobiles has made transportation of people and goods more convenient and quick during the 20th century. As the number of automotive vehicles has increased their importance has also increased in turn, which now makes them an integral part of how modern society operates and functions on a day-to-day basis. Despite the advantages provided by the vehicles there are negative consequences to the increased dependence on automotive vehicles. Chief among these consequences are the effects that the pollutants released by the vehicles have on the environment and these effects are of great concern and require appropriate actions if an environmental crisis is to be avoided. The chemical reactions that are used within diesel- and gasoline-engines lead to the generation of few different carbon-equivalent substances, primarily carbon dioxide (CO_2) and methane (CH_4), which are both greenhouses gas that lead to increasing global temperatures [1]. Within the EU, the transport sector is responsible for about 30% of the total CO_2 emissions, where personal vehicles are responsible for 60.7% of the transportation emissions [2]. This means that the car industry has a large environmental effect which has to be curbed if global environmental change is to be stopped, and hopefully be reversed.

1.1 Background

As alternatives to the carbon dioxide generating combustion engines are being researched, electrified vehicles (EVs) have become more widely adapted and developed. Two of the more common EV-types are HEVs (hybrid electrified vehicles) which generates its propulsion energy through a combustion engine but optimizes the fuel efficiency with the aid of electrical systems and PHEVs (plug-in HEVs) which can store electricity from the grid and can partly be driven using the stored electricity. BEVs (battery electric vehicles) is a fully electric alternative to combustion vehicles, where the engine is driven entirely the stored electricity, and BEV have become more and more common place over the last decade. The reason behind the explosive growth of the BEV market is twofold. Firstly, it is due to the very active research within battery storage, specifically in regards to lithium-ion (Li-Ion) batteries, which has seen immense increases in the possible energy density of a battery as well as much lower production costs. [3]. Secondly, all the needed energy for propulsion as well as vehicle operation is taken from the on board battery units, which are charged through the electrical grid, which means that the usage of the car is as environmentally clean as the power source charging it is. This makes them a very promising solution to the amount of pollution being released by the automotive industry.

Among the different battery technologies, Li-Ion batteries have largely become popular in the automotive applications due to their high energy efficiency, long life cycle, high energy density and high power density [4]. However Li-Ion batteries have a few draw-

backs that require handling and management during use, as the batteries chemistry are prone to become volatile if the battery is abused, such as if the battery is overcharged. This type of abuse can in the best cases lower the life-span of the battery and in the worst cases lead to thermal runway, which is a rapid expulsion of thermal energy or in gentler terms, an explosion. To combat these negatives and ensure that the batteries are being used in a safe manner, Battery Management Systems (BMS) are often employed. The BMS oversees the battery, keeping track of the state with different sensors, including the an estimate of the charge left in the battery, and making sure it is operating in an optimal manner, giving the battery both a longer life and making it safer to use. The importance of a well developed BMS for usage with EVs cannot be understated and a crucial part of this development is thoroughly testing the many different parts of the BMS.

Testing of a BMS is complicated, as the software interfaces with sensors and different parts of the vehicle, and it is not reasonable to perform all tests on a production or prototype vehicle. An alternative to this is Hardware-in-the-loop simulations (HIL) which utilizes mathematical and software representations of everything that is not the hardware which is being tested, i.e. the BMS. A HIL-solution for a BMS could for example include emulations of the temperature and voltage of the battery, a driving cycle for the vehicle as well as environmental aspects such as outside temperature. HIL simulations are a great tool for extensive and proper testing, but as they require setup and as HIL-rigs can be expensive, the amount of testing each developer or engineer can perform on it is often limited.

1.2 Thesis Motivation and Aim

The thesis will be performed as a joint project between Chalmers University of Technology and Volvo Cars in Torslanda. The project will be used for testing of BMS software and hardware that is being developed at Volvo Cars.

The aim of the thesis is to develop a battery simulation model, which is meant to be used for verification for if a first or second order ECM model is best used on an embedded computer chip for BMS development.

1.3 Project Concept

The expanding research and developments in the battery technology has encouraged many organisations to come up with their own software and tools to develop batteries and the control systems for the battery management system. With the batteries being a complex problem due to their non-linear functioning the simulation and testing becomes a vital part of design and optimization of battery performance for an electric vehicle.

The functionality of the battery management is vast which includes managing the current drawn from the battery to the amount of coolant that needs to flow to cool the battery. This varied functionality holds an importance on understanding the working of the battery model. The purpose of the project is to develop a battery model that helps in testing and advancement of ECU that would run in real time on a microcontroller and to validate how closely the model mimics the behaviour of the battery.

2
Theory

2.1 Chemical Battery Theory

2.1.1 Electrochemistry and Redox reactions

A battery is an electrical power storage device that charges and discharges using electrochemical reduction-oxidation (redox) reactions. A battery consists of at least one electrochemical cell, where a cell contains at minimum two electrodes, a separator and an electrolyte. Electrodes are polarized solids that are submerged within the electrolyte fluid, which contains dissolved ions that makes the fluid electrically conductive through the movement of the ions, but not through the movement of electrons [5]. The electrodes are either an anode, which are charge givers, or cathodes that are charge receivers. Either of the electrodes can act as anode or cathode depending on whether the battery is charging or discharging. Electrical power is then charged or discharged from the battery based on how the electrons move between the electrodes. This work is characterised by the potential difference between the anode and cathode, according to

$$\Delta E_{cell} = V_{cathode} - V_{anode} \tag{2.1}$$

where ΔE_{cell} is the electric work (sometimes called electromotive force) in volts [V] with the standard potentials for the electrodes as $V_{cathode}$ and V_{anode} in volts [V].

Redox reactions are a subset of reactions that occur when the oxidization number of a chemical species changes. When a species experiences a loss of electrons, the species is said to have been oxidized and when it experiences an increase of electrons it has been reduced. This means that for each redox reaction, at least one reductant i.e an element that gains electrons and one oxidant that losses electrons is needed. The Nernst equation, see equation 2.2, can be used to evaluate how much electrical work that can be expected from the redox reactions occurring in the battery cell.

$$\Delta E_{cell} = -\frac{\Delta G_r}{nF} \tag{2.2}$$

where ΔG_r is the change in Gibbs energy due to the redox reaction in joules [J], n is the number of electrons that are exchanged in the reaction and $F = 96485.3329\ \frac{s \cdot A}{mol}$ is the Faraday constant.

2.1.2 Lithium-ion batteries

Lithium-ion batteries are a type of battery which contains lithium-ions as one of the main electrodes, as the name implies, and they have been heavily studied and developed over

the past few decades. These batteries are primarily used for tasks where a high energy density is required, as Li-Ion cells are very energy dense, meaning that the contain a high amount of energy per unit of mass, while also being capable of delivering high voltages (3.6V) in comparison to many contemporary batteries [6], [7]. In figure 2.1 we can see how the energy density of different batteries compare [8].

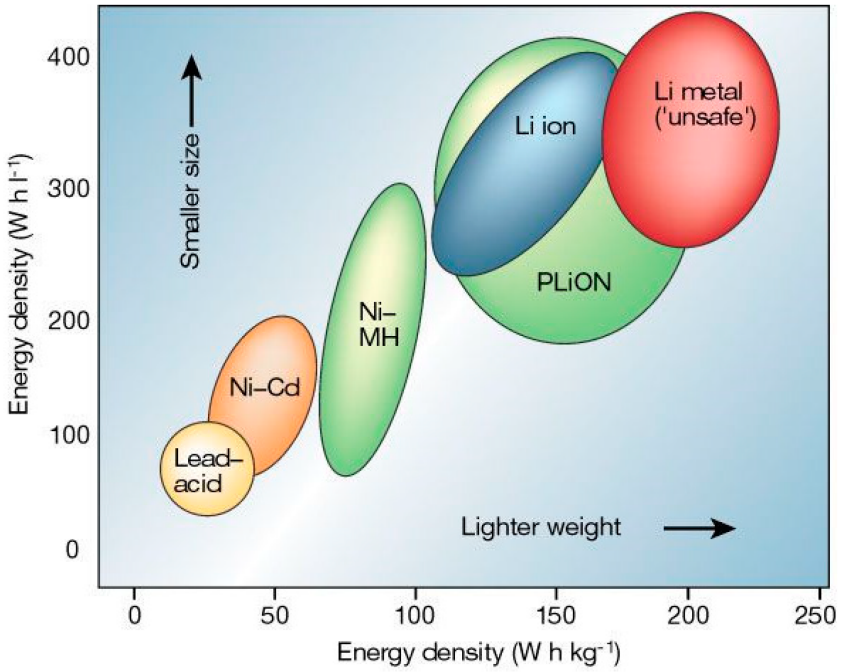


Figure 2.1: Ragone plot for a selection of batteries. From: Miao, Y., Hynan, P., von Jouanne, A., & Yokochi, A. (2019). Current Li-Ion Battery Technologies in Electric Vehicles and Opportunities for Advancements. Energies, 12(6), 1074. MDPI AG. Retrieved from http://dx.doi.org/10.3390/en12061074

Li-Ion batteries are categorized depending on which type of lithium metal oxide is used, which acts as the cathode or positive electrode in the battery [9]. The four most prevalent Li-Ion batteries for use within EVs are listed below.

1. **Lithium Cobalt Oxide**
   Lithium Cobalt Oxide (LCO) batteries are commonly used for handheld or small electronic devices [10]. LCO batteries have a high nominal voltage of 3.9V, but are thermally sensitive which means that they require proper thermal management systems [9].
2. **Lithium Manganese Oxide**
   Batteries of the Lithium Manganese Oxide (LMO) variety are commonplace due to the lower cost of manganese, as it abundant in nature while having a good nominal voltage of 3.7V [9]. LMO batteries have a lower capacity than their LCO counterparts but are more thermally stable while having equal to longer lifespans [9], [10].
3. **Lithium Nickel Manganese Cobalt Oxide**
   Lithium Nickel Manganese Cobalt Oxide (Li-NMC or NMC) batteries combines many of the advantageous factors of LCO- and LMO-batteries. The performance is slightly worse than LCO counterparts, having a lower nominal

voltage of 3.7V but a higher capacity [9], [10]. The thermal stability of NMC variety batteries land somewhere between LCO- and LMO-batteries, which makes them stable enough for more intensive and/or heat generating tasks [10].

4. **Lithium Iron Phosphate**
   Li-Ion batteries of the Lithium Iron Phosphate (LFP) type are stable and tolerant to abusive behaviours, such as overcharging and short-circuiting, and are very thermally stable [9]. This comes at the cost of middling capacity and a lower nominal voltage than other battery counterparts at 3.4V [9], [10].

The performance of Li-Ion batteries comes at a cost of additional complications. They are temperature-sensitive and can depending on the usage situation need either cooling or heating solutions. Short circuiting and overcharging are another set of problems that have to be handled when using Li-Ion battery solutions. The events mentioned above change the chemical structure of the anode within the battery cells, which can lead to thermal runaway occurring, during which the temperature of the cell increases rapidly due to the stored energy in the battery being released all at once [11], [12].

2.2 Battery Modeling

Battery behaviour is dynamic and non-linear, meaning that accurate mathematical modeling is hard. The hard to predict behaviour can lead to models that are to erroneous for practical use cases. Nevertheless, there are several models that can capture the different phenomenons happening within a battery, often at the cost of giving less accurate predictions or the models becoming increasingly complex, leading to longer computational times. One common approach is to use coupled differential equations derived from electrochemical relations, which describe how the concentrations of ions change within the cells, and equations describing the thermal energy change these concentration gradients cause. These equations are solved in many different ways, with Finite Element Methods being among the more prevalent. This gives accurate solutions, at the cost of computation time as well as the need for a large amount of accessible memory, since the solutions requires large numbers of constants and variables [13]. Another model that is widely used is the Equivalent Circuit Method (ECM), as it is easier to implement than differential equation approaches, while still offering good accuracy in relation to computation time [14].

2.2.1 ECM model

The Equivalent Circuit Method (ECM) is a battery model type that assumes the battery properties can be described as a circuit. The circuit contains a resistance coupled in series to a number of resistor-capacitor (RC) circuits, the number of RC-circuits is often either one or two. The resistance and RC-circuits are also coupled (in series) to a voltage source, the open circuit voltage (OCV). What the OCV entails is described in subsection 2.2.2. ECM models are divided into different orders, where the order of the model is determined by the number of energy storing components (capacitors) in the circuit. The RC-circuits are added to the circuit to capture the aforementioned non-linear behavioural aspects, specifically time-transient effects. In a first order ECM, all transient effects are captured within the single RC-circuit while second order ECM models can divide the behaviour into short- and long-term behaviours, represented by a separate RC-branch.

The accuracy of ECM models are determined by the circuit parameters used. To accurately simulate a battery, the parameters have to be extracted from data that describes how the battery behaves. Battery performance is tied to both the temperature and state of charge (SoC) of the battery, which means that the circuit parameters such as the OCV, resistance and capacitance values have to be extracted for a range of SoC-values for a few temperatures for optimal performance [14]–[16]. The OCV specifically has been shown to vary quite largely depending on the current state of the SoC and temperature of the battery [17]

2.2.1.1 First-Order ECM

The first-order ECM has one RC branch connected to the internal resistor which handles the whole battery dynamics. The figure 2.2 depicts a first-order ECM model where the OCV (V) varies non-linearly with respect to the SoC. R_Ω (Ω) is the ohmic resistance which predicts the resistance produced by the electrolyte and the RC chain that is connected in series with the internal resistance depicts the non-linear polarization behaviour of the lithium-ion battery where R_1 (Ω) and C_1 (F) is the polarization resistance and capacitance respectively. The I (A) gives the direction of the direction of the current with the applied load [18].

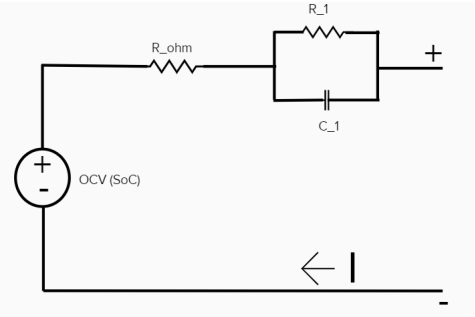


Figure 2.2: First-Order cell circuit

2.2.1.2 Second-Order ECM

In the second-order ECM model in comparison with the first-order has one more RC branch connected in series with internal resistor and RC branch. The figure [2.3] where the additional resistance and capacitance is represented by R_2 (Ω) and C_2 (F) respectively. The second RC branch gives higher levels of accuracy, with the direct consequence of being more computationally expensive due to the added complexity.

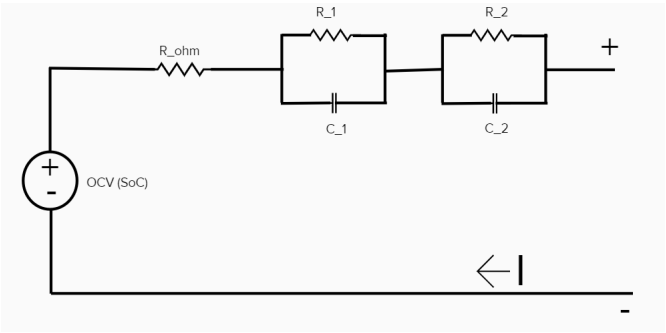


Figure 2.3: Second-Order cell circuit

2.2.2 Parameters Terminologies

The OCV is the voltage difference that is measured between the negative and positive terminal of a circuit when it is disconnected, i.e. no current that is not native to the circuit is flowing between the terminals. The SoC is a measure of how much of the total capacity the battery currently contains. Accurately tracking the SoC is hard, due to the earlier mentioned non-linearities of battery behaviour, but assuming that there are minimal current sinks and loses within the battery, the method of coulomb counting can be used [19]. The governing equation for coulomb counting is defined as

$$SoC(t) = SoC(t=0) + \int_0^t \frac{I(t)}{Q_n} dt \tag{2.3}$$

where $Soc(t=0)$ is the initial state of charge, $I(t)$ is the current curve as a function time in Amperes [A] and Q_n is the nominal capacity of the battery in Ampere-hours [Ah]. Equation 2.3 can be updated to account for discrete timesteps and this rewrite takes the form

$$SoC_i = SoC_{i-1} + \frac{I_i}{Q_n} \cdot \Delta t. \tag{2.4}$$

The discretizied formula allows for updates based on the timestep Δt, which is useful in simulation.

For a given current I, the potential change for a discrete first order ECM model is given by

$$U = U_{OCV} + IR_\Omega + \frac{I \cdot R_\tau}{1 + R_\tau \cdot C_\tau} \tag{2.5}$$

In equation 2.5, R_Ω corresponds to the instant voltage change when a current is applied to the cell, while R_τ and C_τ correspond to the transient response of the cell [20]. The sign of the current I determines if the cell is charging or discharging, with a positive sign corresponds to the former. The second order ECM model is similar, with the discrete voltage change expression being

$$U = U_{OCV} + IR_\Omega + \frac{I \cdot R_1}{1 + R_1 \cdot C_1} + \frac{I \cdot R_2}{1 + R_2 \cdot C_2} \tag{2.6}$$

In equation 2.6 above, the terms R_1 and C_1 now correspond to short term transient effects while R_2 and C_2 correspond to the longer term effects, as well as the diffusion happening within the cells [20], [21].

Equations 2.5 and 2.6 refer to discretized models, but continuous equations can also be formulated. For the first order, the new equation becomes

$$U(t) = U_{OCV} + I(t)R_\Omega + V_1 e^{\frac{-t}{\tau}} \tag{2.7}$$

where V_1 is the saturated capacitor voltage corresponding to capacitor C_τ in equation 2.5, t is time in seconds and τ is the relaxation time in seconds [22]. The relaxation time is an approximation of the time needed for the battery cell to reach a steady-state or chemical equilibrium. For the first order model, τ corresponds to all time transient effects.
Making equation 2.6 continuous we get

$$U(t) = U_{OCV} + I(t)R_\Omega + V_1 e^{\frac{-t}{\tau_1}} + V_2 e^{\frac{-t}{\tau_2}} \tag{2.8}$$

where τ_1 encapsulates short term time transient behaviour and τ_2 the long term behaviour.

2.2.3 ECM heat generation

As was previously mentioned, the batteries behaviour are vary depending on two primary factors, the SoC and the cell temperature. This means that accurate tracking and prediction of temperature changes will lead to a more accurate battery simulation. The simplest heat model for a battery is described in equation 2.9

$$Q = Q_i + Q_r \tag{2.9}$$

where Q is the total heat generation, Q_i is irreversible heat generation and Q_r is reversible heat generation [23]. This description is is severely lacking in detail, so equation 2.9 is expanded into

$$Q = Q_i + Q_r = R \cdot I^2 + T\Delta S \tag{2.10}$$

where the irreversible heat is described by the ohmic heat generation term $R \cdot I^2$ and the reversible heat is handled by the reactions entropic change in the term ΔS. As the cell temperature is the term that is of interest, the total heat term Q can be rewritten as $Q = m \cdot c_p \frac{dT_{cell}}{dt}$, where m is the mass of the cell and c_p is the heat capacity of the cell [24]. Assuming that the battery cells temperature is always uniform in its geometry the heat exchange between the cell and and a still environment, meaning no forced convection, can be written as

$$m \cdot c_p \frac{dT_{cell}}{dt} = RI^2 + T\Delta S + Ah(T_\infty - T_{cell}) \tag{2.11}$$

where A is the area of the cell exposed to environment in $[m^2]$, which can be assumed to be the cells surface area in rough estimation, h is the convection heat-transfer coefficient in $[\frac{W}{m^2 K}]$ and T_∞ is the surrounding environment temperature in $[K]$. Equation 2.11 can be further expanded to cover any heating or cooling efforts as a single Q_o term. By noting that the irreversible heat generation is the largest contribution factor in the total heat generation process, the $T\Delta S$ term can be removed without a large loss in accuracy [25]. Equation 2.11 can now be discretized and modified to yield

$$T_i = T_{i-1} + (RI^2 + Ah(T_\infty - T_{i-1}) + Q_o)\frac{\Delta t}{m \cdot c_p} \tag{2.12}$$

where T_i is the cell temperature at timestep i and $\Delta t = t_i - t_{i-1}$ is the timestep or time difference between steps i and $i-1$.

2.3 Parameter Fitting - Least Squares Method

Given a dataset (y_i, x_i), where y_i is data dependent on the in turn independent data x_i and i is the indexation of the dataset, it is often of interest to fit this data onto some model function $f(x_i, \beta) \approx y_i$, where β are the function parameters that are to be fitted onto the dataset. A common approach to this type of problem is to apply the method of least squares (LSM), which is a regression model that optimizes the function parameters β by minimizing the square-sum of the residuals (RSS),

$$RSS = \sum_{i}^{n} r_i^2 \tag{2.13}$$

where the residuals r_i are described by

$$r_i = y_i - f(x_i, \beta). \tag{2.14}$$

An example of this can be seen in figure 2.4, where a function $f(x, \beta_1, \beta_2) = x\beta_1 + \beta_2$ was fitted onto data that was generated along the line $y = x$ but then displaced with values drawn from a uniform distribution. This is an example of a *linear least squares* regression model, as the function f can be broken down into a linear combination of the optimization terms β according to $f(x, \beta) = \sum_j \beta_j \phi_j(x)$. Linear LSM problems can be solved explicitly by means of finding the minimum of equation 2.13. This is done by taking the partial derivatives of the expression with regards to the optimization parameters β_i and setting them to zero. The parameters can then be solved from the resulting equation system.

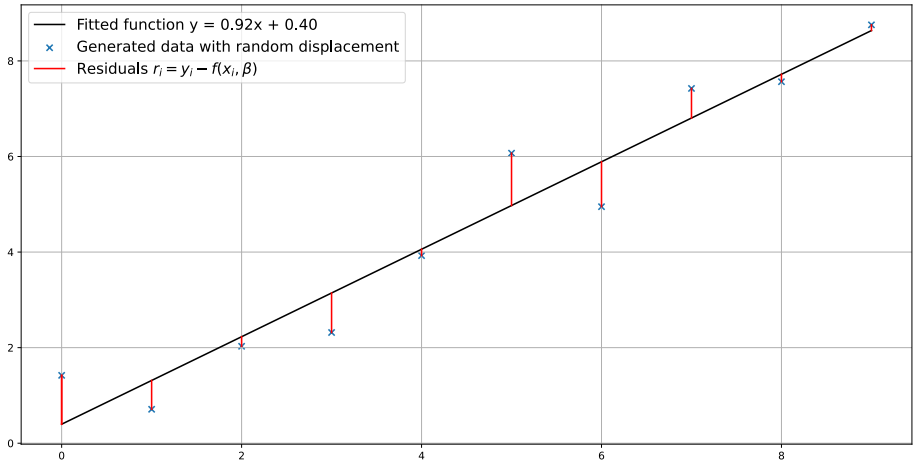


Figure 2.4: Plot of a linear line with fitted parameters using the least squares methods. The generated data was generated along the line $y = x$ and then randomly displace based on a uniform distribution.

When *non-linear* functions are used it often becomes necessary to find the optimal β values through the use of iteration. By setting $\beta^n = \beta^{n-1} + \delta\beta$, where the superscript n represents the iteration number, the parameters can be updated continuously, but this requires setting initial values for the parameters. As the β values have to be numerically found through iteration, there are many different algorithms for this. A few methods uses an approach similar to that of the linear-case, where the goal is to minimize the gradient of the RSS. One of the more prevalent algorithms for non-linear LSM parameter determination is the Levenberg–Marquardt algorithm, which uses the assumption

$$f(x_i, \beta + \delta\beta) \approx f(x_i, \beta) + \mathbf{J}_i\delta\beta \tag{2.15}$$

which is equivalent to approximating f by means of a first order Taylor expansion [26]. Here $\mathbf{J}_i$ is a row in the Jacobian matrix (or gradient row-vector) with respect to β, i.e

$$\mathbf{J}_i = \frac{\partial f(x_i, \beta)}{\partial \beta}. \tag{2.16}$$

Inserting the assumption defined in equation 2.15 into the RSS equation as defined in 2.13 the following expression is obtained

$$RSS = \sum_i^n (y_i - f(x_i, \beta) - \mathbf{J}_i\delta\beta. \tag{2.17}$$

Equation 2.17 can then be vectorized, becoming

$$RSS = ||\mathbf{y} - \mathbf{f}(\beta) - \mathbf{J}\delta\beta||^2.$$

Taking the gradient of the expression above with respect to $\delta\beta$ and setting the gradient to zero the following can be obtained after simplification

$$(\mathbf{J}^T\mathbf{J})\delta\beta = \mathbf{J}^T[\mathbf{y} - \mathbf{f}(\beta)].$$

This expression yields an equation consisting of n linear equations, from which $\delta\beta$ can be solved for. The process then repeats, until either some threshold is meet, such as having reached a number of iterations or the RSS is below some limit.

3
Methods

3.1 Software and hardware choice

The model was developed for usage within Volvo Cars Corporation, which limits the hardware that the model is supposed to run on to what was available from the company. The chosen embedded computer was a Raspberry Pi Model 4B (RPi4B), which in turn informed the choice of programming language. The RPi4B natively runs an operating system (OS) called Raspberry Pi OS (RPOS), which is a port/fork of Debian, which in turn is an open-source OS developed on the Linux Kernel. RPOS has built-in support for the Python programming language, which is one of the reasons that the language was chosen.

Python was also chosen for the fact that it is dynamically typed and memory safe, which makes the development process quicker. Python is written in the C-programming language, which has a rich history and is used for tasks that require efficiency, at the cost of being 'harder' to develop for due to the low-level granted to developers. Python is also widely supported, with libraries for static memory handling and computation with the NumPy library and data-analysis tools using the Pandas and Sci-Py libraries. The aforementioned NumPy library allows for the usage of static memory arrays, which are faster to access at the cost of being harder to expand, as the data has to be copied into a new allocation of memory if the need arises.

As Python is an Object-Oriented programming language (OOP), the main development focus is on creating and using Objects of different Classes. Classes act as a template of sorts for the objects which tells the program what each object must contain. Classes contain attributes, which in very general terms can be described as information of different types, and methods, which perform operations on or onto information. OOP is a very powerful tool for quickly developing frameworks for programs that require the usage of many similar 'packs' of information, while keeping the information for each instance of an object seperate.

3.2 ECM model parameters

3.2.1 Parameter extraction

In the equations describing the ECM models, 2.5 for the first order and 2.6 for the second order model, the equations are dependent on the circuit parameters and these parameters are in turn dependent on temperature and the SoC of the battery cell. To sufficiently use the ECM model, the accompanying parameters have to be extracted at different tem-

peratures for a range of SoC-levels. A common approach for extracting the required parameters is to use Hybrid Pulse Power Characterization (HPPC) [27]. During a HPPC test, the battery is kept in an environment with a fixed temperature and a current at a set C-rate is then applied to the battery for a limited time. The voltage is measured during the entire experiment, but the interval between measurements is much smaller when a current pulse is applied. Figure 3.1a presents how a HPPC current pulse will affect the voltage, for the dataset described in [28].

In figure 3.1a, the triangle marker represents the extracted OCV value, the cross marks where the instantaneous voltage drop ends and the circle marks where the current is turned off. The corresponding current pulse can be seen in figure 3.1b, where the markings correspond to the same events as noted in the previous figure. The figures are created based on data extracted from tests performed by Dr. Phillip Kollmeye at the University of Wisconsin-Madison [28], [29]. This dataset was also used to develop and verify the model.

To extract the parameters each pulse and its corresponding relaxation time, which is the time it takes for the battery to reach a steady-state with regards to the voltage, was found by finding all points that roughly have the same applied current as the C-rate which is being examined. This was to a C-rate of 1, which for the dataset being used corresponds to a current of 2.9 Amperes as the nominal capacity of the battery is 2.9 Ah. After the data has been divided into smaller 'chunks', which contain a single pulse each, as the extracted data is time-dependent, equations 2.7 and 2.8 will be used for data extraction, where the equations are fitted onto the data using least-squares. Taking the first order case as an example, we simplify the terms by combine some of the terms, see equation 3.1 below. The combination is done to make the fitting process easier.

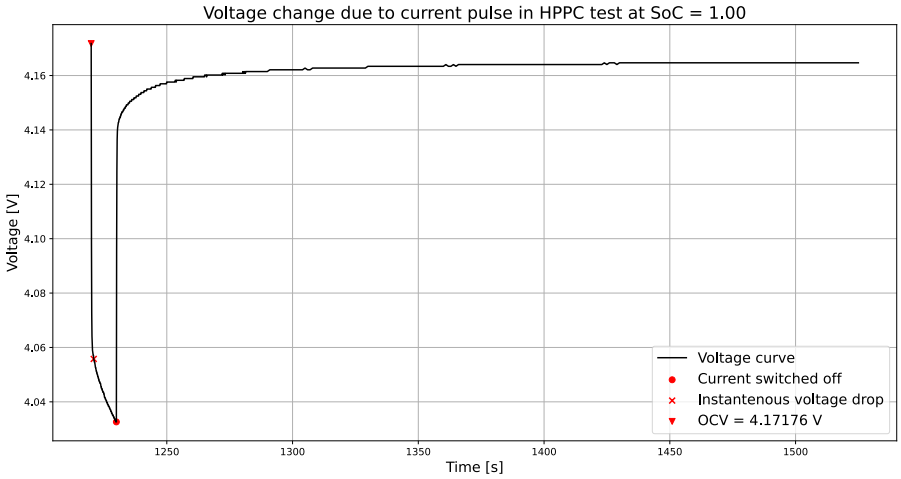
$$U(t) = d_0 + d_1 e^{-tA}, \tag{3.1}$$

where $U(t)$ is the voltage across the battery cell at time t, $d_0 = U_{OCV} + R_\Omega I$ represents the instantaneous voltage change due to application of a current to the cell, V_1 is changed into d_1, where $d_1 = R_1 I$ and finally the relaxation time τ_1 is rewritten as $A = \frac{1}{R_1 C_1}$. In the same way equation 2.8 also needs to be rewritten into equation 3.2.

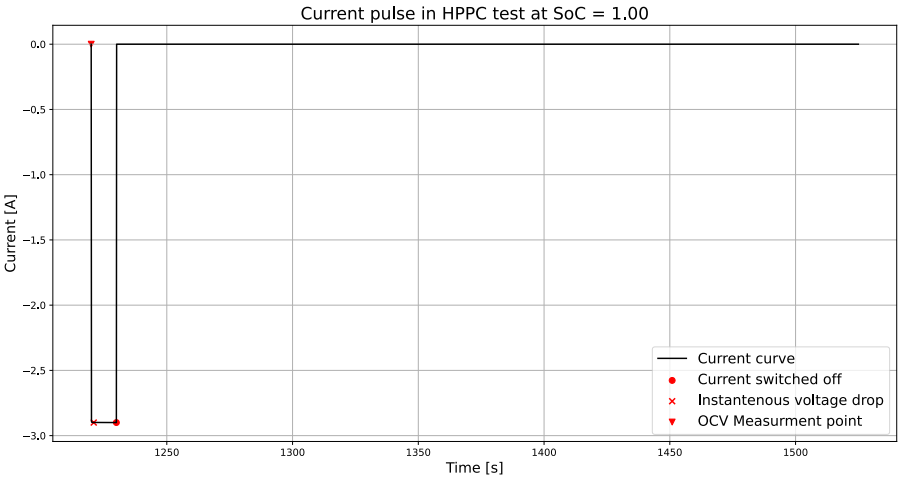
$$U = d_0 + d_1 e^{-tA} + d_2 e^{-tB} \tag{3.2}$$

where $d_2 = R_2 I$ and $B = \frac{1}{R_2 C_2}$.
Equation 3.1 or 3.2 is then fitted onto a pulse, such as in figure 3.1a, using the least-squares method. After the fitting process the circuit parameters can be extracted if the applied current I and the OCV at the current SoC and temperature is known. The current is obtainted through the known C-rate of the pulse. The OCV was obtained by assuming that the battery had reached a steady-state before each pulse, which means that the voltage that was measured just before the pulse is set to be the OCV, represented by the triangle in figure 3.1a.

The first order model parameters are fitted onto the HPPC data, as presented in figure 3.1a all at once, as the terms in equation 3.1 are different (one linear and one exponential term), which means they encapsulates different behaviour and are easy to fit at the same time.



(a) Voltage-time curve for a pulse during HPPC testing.



(b) Current pulse during HPPC testing.

Figure 3.1: Shows how a Panasonic 18650 batteries voltage will drop according to an applied current pulse during a HPPC test, as well as the required time for the cell to reach a steady-state.

The second order model requires another scheme for fitting the parameters. The two exponential terms in equation 3.2 makes it more prone to overfitting. Overfitting would mean that the model would work well only on the data from which the parameters are extracted. To generalize the model, we need to use the fact that one of the exponential terms encapsulates long term behaviour and the other short term.

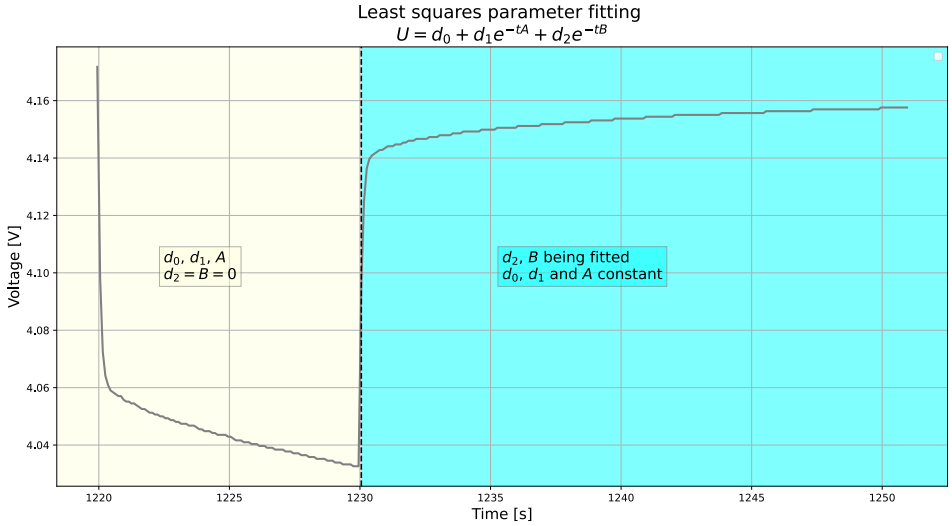
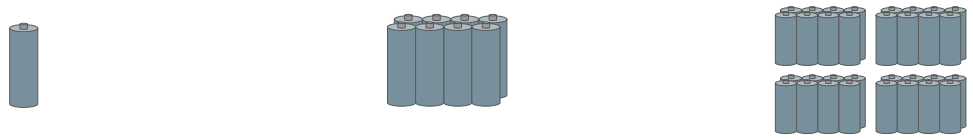


Figure 3.2: Displays the different regions of the HPPC curve that the parameters are fitted onto.

In figure 3.2 the yellow region represents the part of the HPPC curve from which the OCV, ohmic resistance and short-term transient effects are approximated. To achieve this, it is assumed that the long-term effects during this part of the curve are negligible, which is done by setting $d_2 = B = 0$ in this region. In the cyan region only the long term effects are considered, which means that d_0, d_1 and A are kept constant during this second region. The line between the short-term and long-term region is drawn immediately after the current is turned off.

3.2.2 Model parameter handling

The parameters were extracted for a set of different temperatures and for different SoC values for each of the temperature values. The data was organised by creating multi-dimensional arrays, where indexation was done by creating key-value pairs. Firstly, the model compares the current battery cell temperature to the temperatures at which the data was extracted and finds which one is closest to it. The model then repeats this process for the SoC, comparing the internal cell SoC and the values of SoC from where the parameters were extracted. After both the update temperature and SoC have been found, the cells parameters are updated. As each cell can have different values of SoC and temperature, this process is repeated for each cell.



(a) Battery Cell **(b)** Battery Module **(c)** Battery Pack

Figure 3.3: Generalized layout of a battery cell, module and pack. A module consists of cells and a pack consists of modules.

3.2.3 Panasonic 18650PF Li-ion Battery Thermal Properties

To simulate the battery cells temperature some physical properties are needed. In equation 2.12, the needed properties are the convection area A, the heat transfer coefficient h, the battery mass m and the specific heat capacity C_P. From Panasonic's datasheet for the 18650PF series of batteries, the diameter $d = 18.55$ mm, the height $h_l = 65.1$ mm and the approximate mass $m = 47$ g can be obtained [30]. It is assumed that the entire surface area of the cell is in contact with air, which means that the convection area can be calculated as $A = \pi d h_l + \pi \frac{d^2}{2}$ as the cell is assumed to by cylindrical. The thermal characteristic are approximated to be the following: $C_P = 960 \frac{J}{kg \cdot K}$ and $h = 22.46 \frac{W}{m^2 K}$. This was done by taking the mean value from a range of empirical values found in literature [31].

3.3 ECM model program

The program simulates the battery by grouping up a number of battery cells within a battery module. This approach of encapsulating the cells within the modules allows the program to minimize how often the circuit parameters have to be saved into memory, as they can now be saved and accessed from the module objects. The simulation uses standard SI units, with the largest change from the norm being that the nominal capacity Q_n in equation 2.3 is given in terms of Ampere-seconds [As] instead of the normative Ampere-hours [Ah]. The implemented model uses discrete time-steps defined in units of seconds, so all time dependent quantities are converted to reflect to utilize seconds. This eliminates the need for unit conversions after the initialization of the model.

3.3.1 Battery Cell class

Objects of the battery cell class are the base foundation of the simulation, as the actual simulation primarily occurs within these objects. The cells only contain data which is related to itself, which are either constant or dynamic data features. The constants consists of physical aspects which are not changed during the simulation, such as the area or mass of the cell, as well as ECM order. Data which is updated through out the simulation, such as the OCV, circuit parameters and temperature belong to the dynamic data type.

As mentioned earlier, the ECM order is set when the cell is initialized. The initial temperature, SoC and all the constant values are also set during the initialization. After the order has been set, the cell then looks up and stores the neccessary circuit parameters based on the order of the cell as well as SoC and temperature values. To make the simu-

lation easier to create while also removing a few repeated logic-checks (checking the ECM order of the cell for each simulation step in the runtime) the cell will then also set the voltage calculation function depending on the ECM order during the initialization. This is done by creating a reference to one of the two possible update functions depending on the cell order.

3.3.2 Battery Module class

The battery module class is written as an object that contains multiple Battery Cell class objects. The module creates and initializes the cells and stores references to all these cells within a list. Each module contains the data which is used to update the circuit parameters. The circuit parameters are stored as lists within a dictionary. A dictionary in Python is a data structure that uses key-value pairs for accessing the stored information, instead of the regular indexing. The key for the dictionaries is the temperature at which the parameters where extracted in units of degrees Celsius in a string format. The module looks at the cells temperature and identifies which of the temperature keys is closest to it. After this, the module is given access to an array which contains all the needed parameters for different SoC values. To access the correct list of parameters, the index is calculated by examining the cells SoC value and finding which extracted data point it is closest to. This process is repeated for each cell within the module, meaning that different cells can have very different temperature and SoC values, and therefore have wildly different voltage behaviour.

During the simulation, the module is sent a current that is to be applied to it and therefore the cells that it contains. The calculations are described in greater detail in 1 for a first order cell. The module uses the voltage change function defined within the cell, extracts the needed potentials and then calls on the cell to update its internal parameters. The potential change is summarized and after all cells have been updated, this potential is sent out as an output to the overhanging simulation program.

3.3.3 Battery Simulation and Update

During the update stage of the model a current is applied to all cells that being simulated. Based on this current, the appropriate voltage change is calculated and summed with the OCV value held by each cell. Each module keeps track of the voltages from each cell while also summing them together. After the voltages have been calculated, the cells dynamic parameters (SoC and temperature) are updated. The SoC update is done as described by equation 2.4 while the temperature update is done by a modified version of equation 2.12, where the irreversible heat is calculated by all the voltage changes within the model circuit.

$$T_i = T_{i-1} + (|U_\Omega I| + |U_1 I| + h_{th}(T_\infty - T_{i-1}) + Q_o)\frac{\Delta t}{C_{th}}. \tag{3.3}$$

As the outputs are directly linked to the dynamic parameters, the needed calculations are performed before updating the internal state. The circuit parameters are then updated as needed based on the newly calculated SoC and temperature values, according to the scheme described in subsection 3.2.2.
Equation 3.3 is the temperature change for a first order ECM cell. The absolute sign ensures that no matter what notation is used for the current (negative for discharge or

charge), the temperature increases due to either the exothermic nature of the discharge reaction. As equation 2.12 assumes that the ohmic resistance R is known, but the abstraction of the ECM models spread the effect of the ohmic resistance across the different battery parameters. This is why equation 3.3 is used instead.

Algorithm 1: Potential calculation and cell internals update

Data: $t_i > t_{i-1}$
Result: $\Delta U_{tot} = \sum_{cells} \Delta U_{cell}$

$\Delta t \leftarrow t_i - t_{i-1}$;
$\Delta U_{tot} \leftarrow 0$;
$I \leftarrow I_{input}$;
for *Cell in Module* **do**
  $Q_{Cell} \leftarrow A_{cell} h_{cell}(T_\infty - T_{Cell})$;
  $[U_\Omega, U_\tau] \leftarrow$ Equation 2.5;
  $\Delta U_{tot} \leftarrow U_{OCV}^{Cell} + U_\Omega + U_\tau$;
  if $I < 0$ **then**
    $Q_{Cell} \leftarrow Q_{Cell} + |U_\Omega I| + |U_\tau I|$;
  end
  $SoC_{Cell} \leftarrow$ Equation 2.4;
  $T_{Cell} \leftarrow T_{Cell} + \frac{Q_{Cell}}{m \cdot c_p} \Delta t$;
end

The process described in Algorithm 1 is used for each of the different modules within the simulation and is essential tool for getting knowledge of how the cells are performing. It also generates the two primary outputs, namely the total potential across the battery module as well as the temperatures of the cells. The module temperature is naively calculated to be the mean value of the internal cell temperatures. This is then repeated for each step in time. The update of the circuit is run less often, as the SoC and temperature don't change rapidly for the small timesteps used for simulation, as $\Delta t \in [0.1, 1]$ seconds for all of the simulations used in this paper.

4
Results

4.1 Extracted ECM Model Parameters

Based on the scheme laid out in subsection 3.2.1, the ECM parameters were extracted based on HPPC data from the dataset developed by Dr. Phillip Kollmeyer at the University of Wisconsin-Madison [28]. The first order parameters can be found in subsection A.1.1 and the second order parameters can be found in subsection A.1.2 within the Appendix. Parameters that could not be extracted at certain combinations of temperature and SoC are encoded as the shebang symbol, #!.

The extracted OCV values at different SoC are presented in table 4.1. The OCV-SoC relation was extracted for the temperature of 25 $°C$, as this was the most complete part of the data.

Table 4.1: Extracted OCV values at different SoC values at 25 $°C$.

SoC	OCV
1.0	4.17176
0.95	4.10356
0.9	4.05723
0.8	3.94528
0.7	3.86164
0.6	3.77092
0.5	3.66348
0.4	3.60236
0.3	3.55088
0.25	3.51228
0.2	3.45695
0.15	3.38875
0.1	3.34436
0.05	3.23112

4.2 Model Results

In figure 4.1 we see how the model compares to the experimental data for a driving cycle. As the the time spans for the driving cycles are large, in the range of 10000 seconds, in comparison to the time steps for the model, which are on the scale off being between 0.1 and 0.5 seconds, the rest of the results will be presented on a smaller timescale.

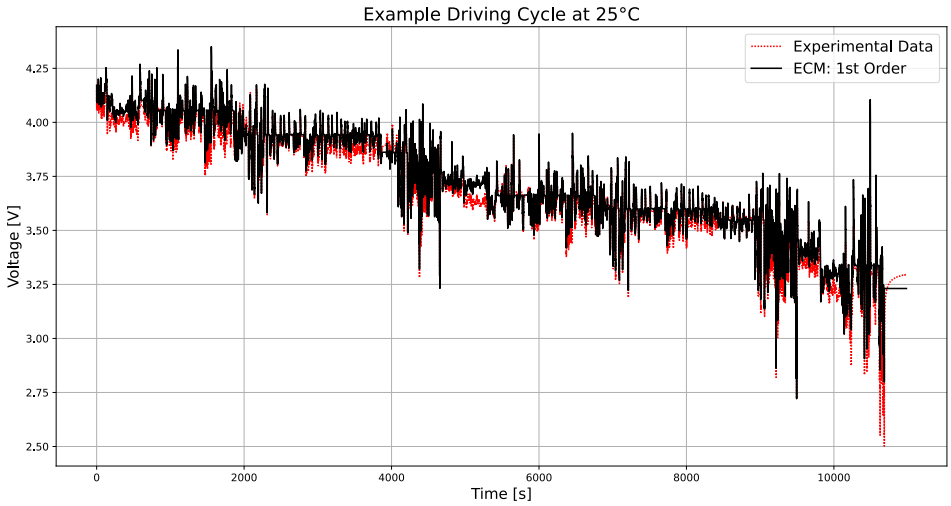


Figure 4.1: Comparison between model and data for a driving cycle. The red dashed line is the voltage data [28] and the black full line is the first order simulation for the same current pulse.

Using the extracted parameters, the model was examined in terms of accuracy over a range of tests. These tests were done for both the first and second order ECM models. The accuracy was measured in terms of the absolute difference between model output and data value, according to equation 4.1

$$E_i = \frac{|y_i - f(x_i)|}{y_i}. \tag{4.1}$$

In equation 4.1, y_i is the experimental data that arose due to some input data x_i. The input data is then fed into the approximate model function f. By taking the absolute difference and dividing by the expected output y_i the error, or divergence, from the expected output is found. This is then used as a measure of the accuracy of the model, where a lower E_i value means that the model lies closer to the experimental data.

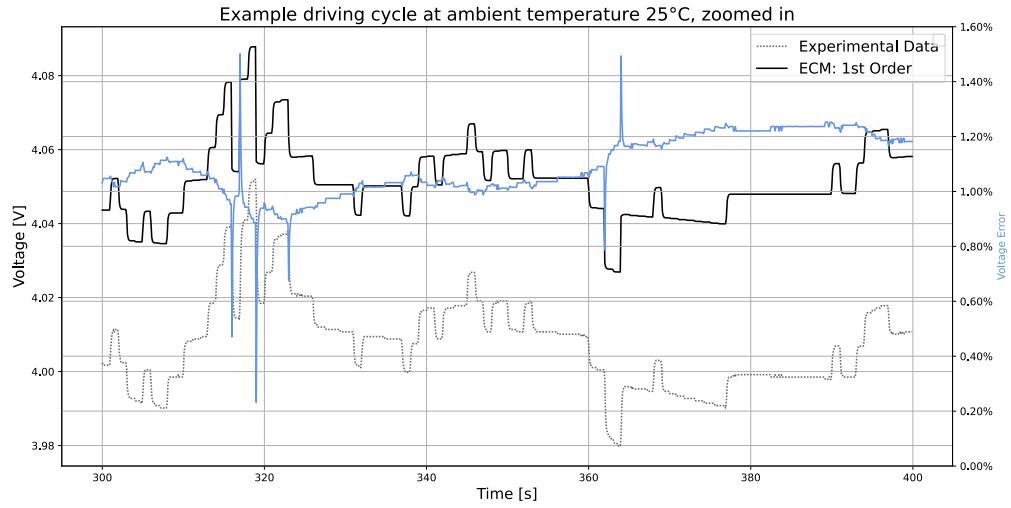


Figure 4.2: The figure shows how the models (black line) voltage simulation compares to the experimental voltage data (gray dotted line). The light blue line is the error between model and data.

Figure 4.2 is zoomed in on an area of figure 4.1 and shows the voltages. This allows for a clearer view between the differences of the model and data. The black and gray-dotted lines are plotted against the left y-axis, representing the voltage outputs resulting from the current, which can be seen in figure 4.4. The light blue line is plotted against the right y-axis and is the error, as calculated by equation 4.1.

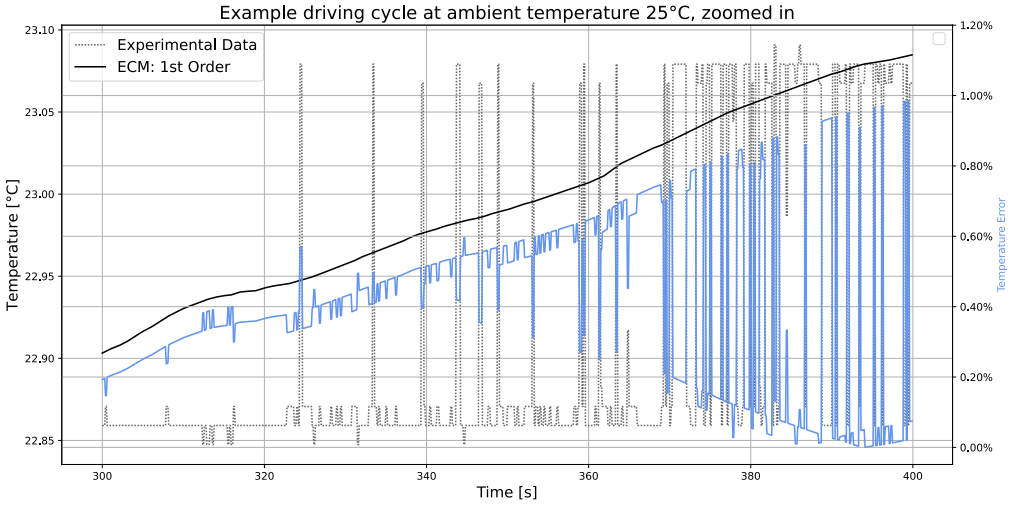


Figure 4.3: Shows how the model and data differ from each other, and are represented by the black line and gray dotted line respectively. The light blue line is the error between the model and data.

The same approach as for figure 4.2 was used for the temperature plot in figure 4.3. The black line and dotted gray line are plotted against the left y-axis, which represents the temperature in terms of degrees Celsius. The right y-axis is the error.

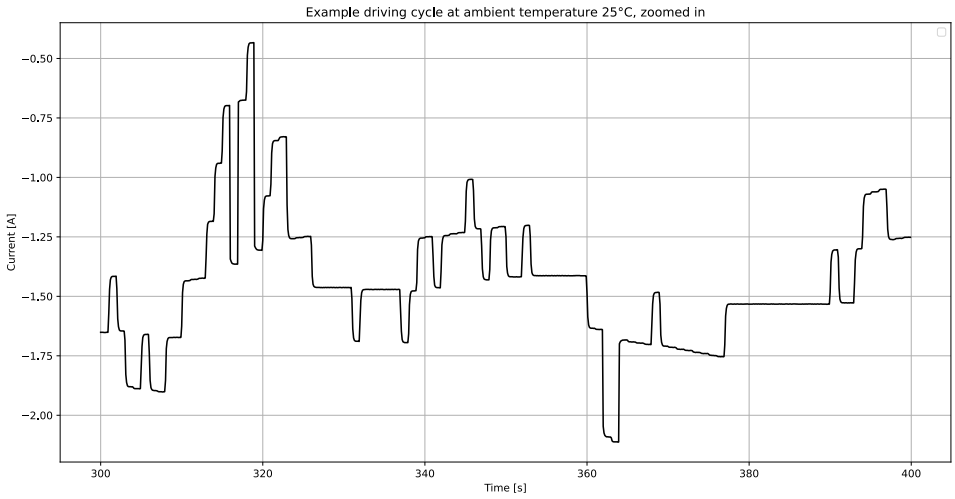


Figure 4.4: The applied current during a small time span of a driving cycle.

Figure 4.4 is the current that is applied during the time span of the driving cycle which is represented in figures 4.2 and 4.3. A negative current implies a discharge and a positive current results in the charging of the battery, according to the standard used for the model.

In figures 4.5 and 4.6 the first and second order ECM models are plotted against each other, along with the experimental data that corresponds to the correct output.

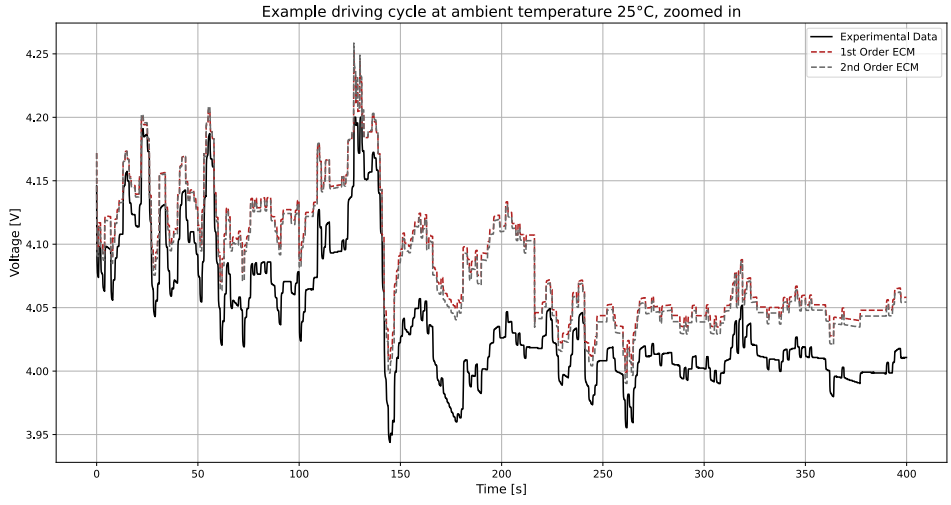


Figure 4.5: First Order (brown dashed line) and Second Order (gray dashed line) ECM models plotted with experimental voltage output.

Figure 4.5 shows how the two different ECM-models output voltages based on the current input shown in figure 4.4.

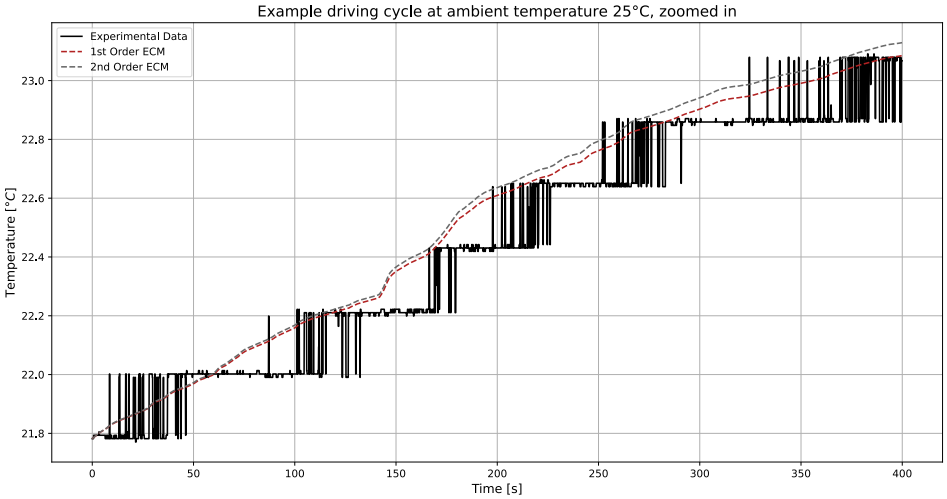


Figure 4.6: First Order (brown dashed line) and Second Order (gray dashed line) ECM models plotted with experimental temperature.

In figure 4.6 above the simulated temperatures for the first and second order ECM models are plotted along with the measured cell temperature. The second order model has slightly higher temperature values than the first order model.

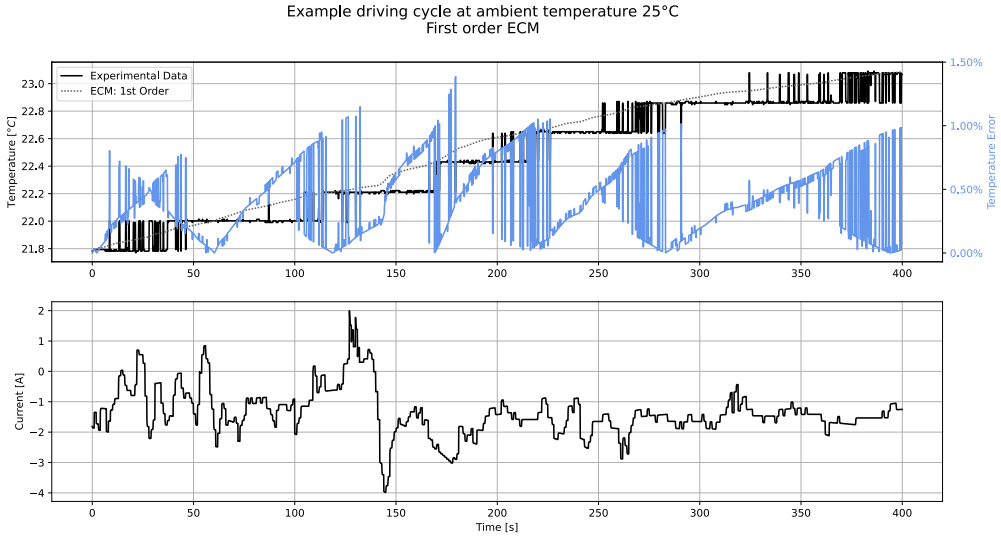


Figure 4.7: The upper figure is temperature from the model and temperature data along with the error between the two. The lower figure is the driving cycle current.

In figure 4.7 the upper figure shows how the battery temperature develops over time for the first order ECM model (brown dotted line) in comparison to the experimental data (black line), along with the error between these two (blue dashed line). This figure shows the output from the first order ECM model. The lower figure is the current for the plotted driving cycle.

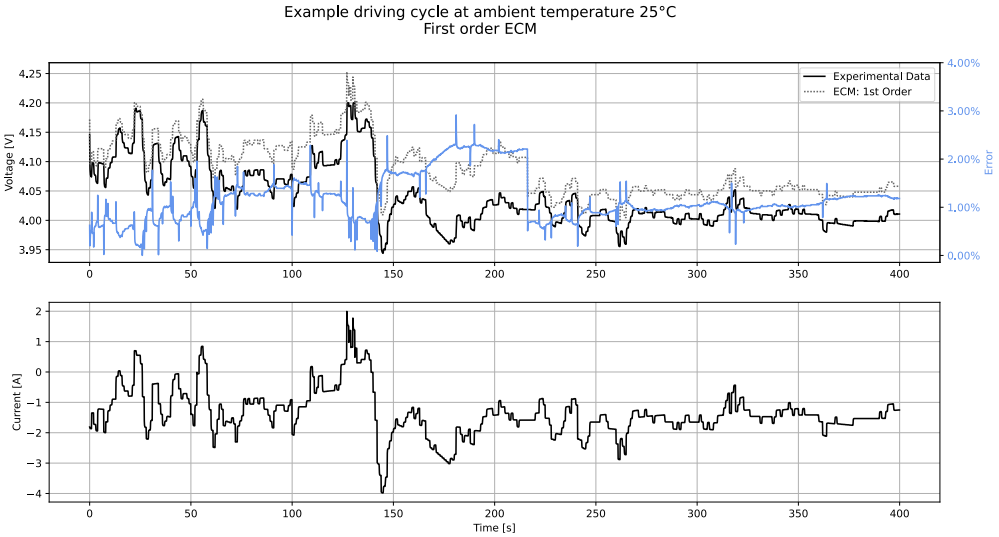


Figure 4.8: The upper figure is temperature from the model and voltage data temperature along with the error between the two. The lower figure is the driving cycle current.

Figure 4.8 is the voltage equivalent to figure 4.7, with the same description being true for this figure as well.

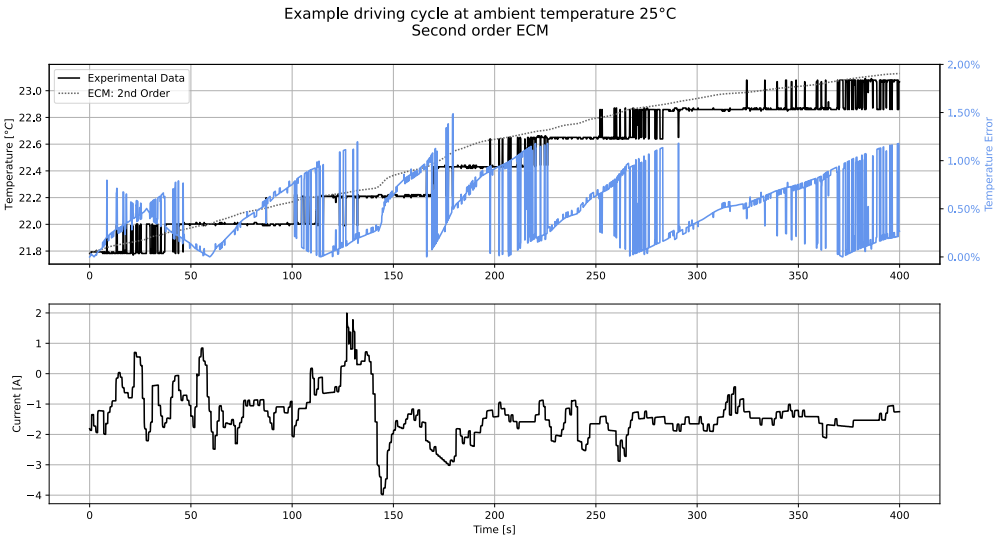


Figure 4.9: The upper figure is model and actual temperature values, plotted with the error between the two. The lower figure is the current that gave rise to the temperatures.

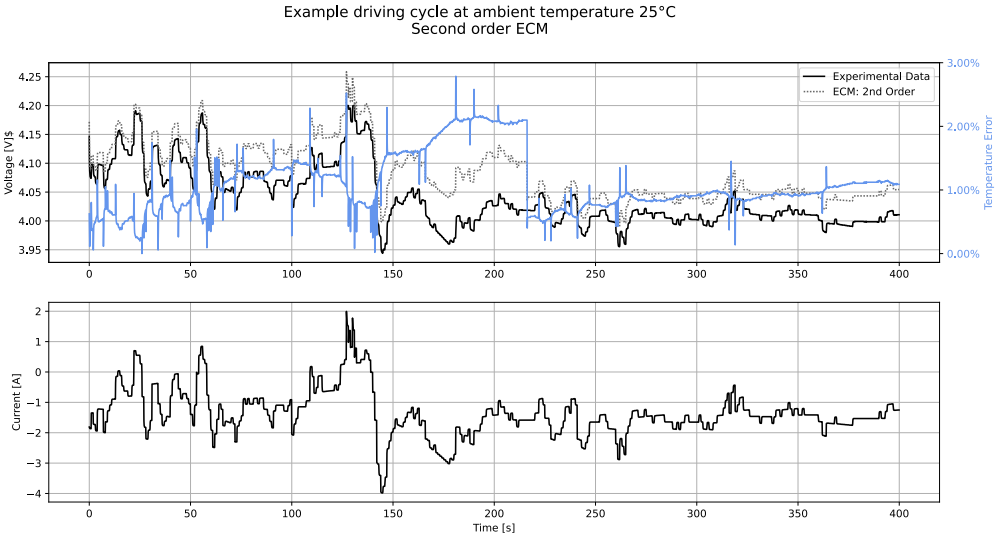


Figure 4.10: The upper figure shows how the model and experimental voltages vary, along with the error between these two. The lower figure is the current from the driving cycle used to plot.

Figures 4.9 and 4.10 are equivalent to 4.7 and 4.8 for the second order ECM. The first figure shows how the temperature varies for the model and the second figure how the voltage varies from the experimental data.

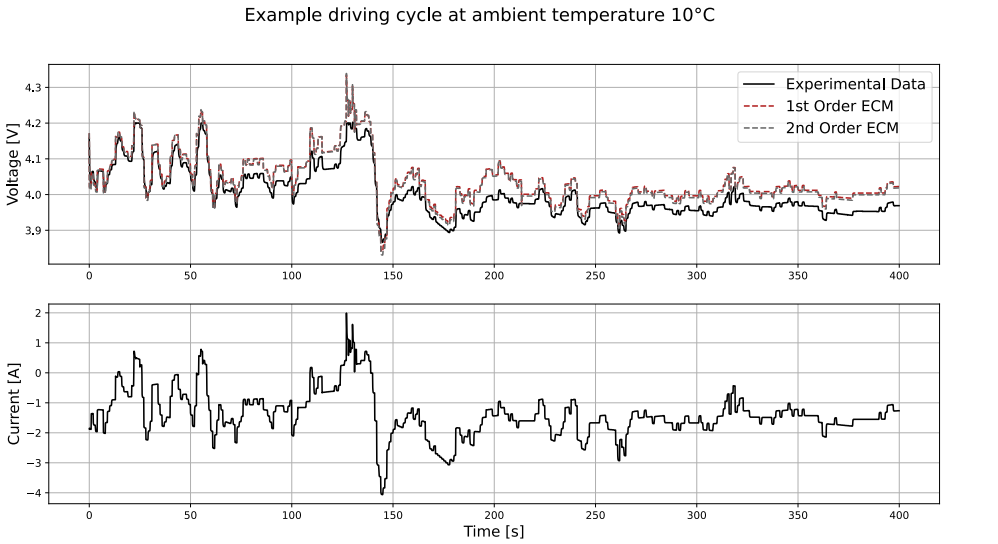


Figure 4.11: The upper figure shows how the actual voltage compares to the output from the first and second order ECM models for an ambient temperature of 10 $°C$.

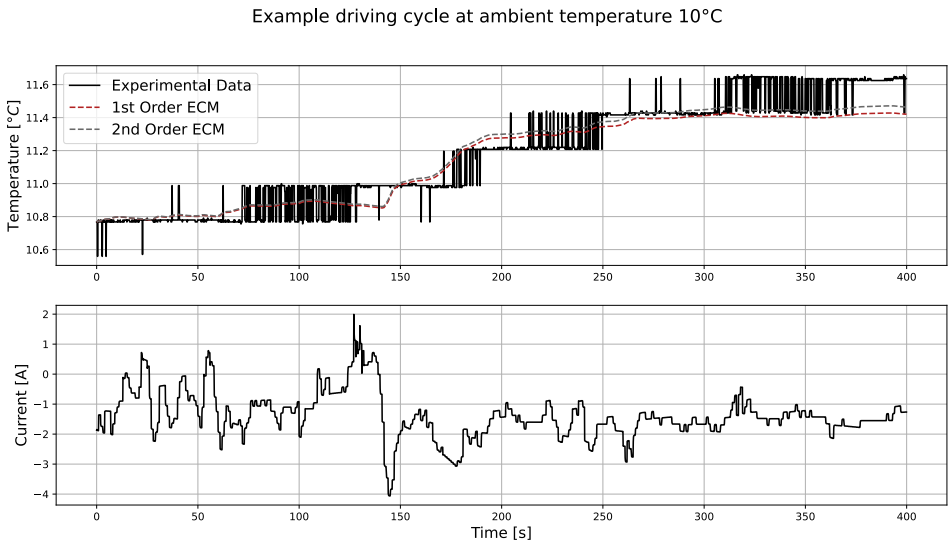


Figure 4.12: First and second order ECM model temperatures compared to the experimental temperature data for a driving cycle.

Figures 4.11 and 4.12 show how the two ECM models output (temperature and voltage) based on the current graph shown in the lower part of the figures. The black line shows the recorded voltages and temperatures from the dataset, while the brown dashed line is the first order and the gray dashed line is the second order ECM models, respectively. This description is true for all figures between and including figures 4.13 and 4.18.

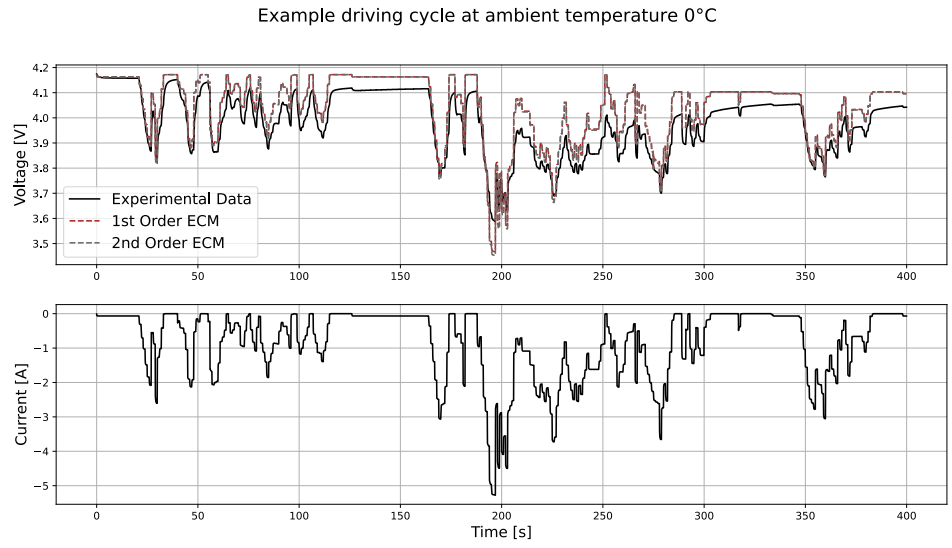


Figure 4.13: The upper figure shows how the actual voltage compares to the output from the first and second order ECM models for an ambient temperature of 0 $°C$.

In figure 4.13 both of the ECM models overlap each other during the entire presented timespan.

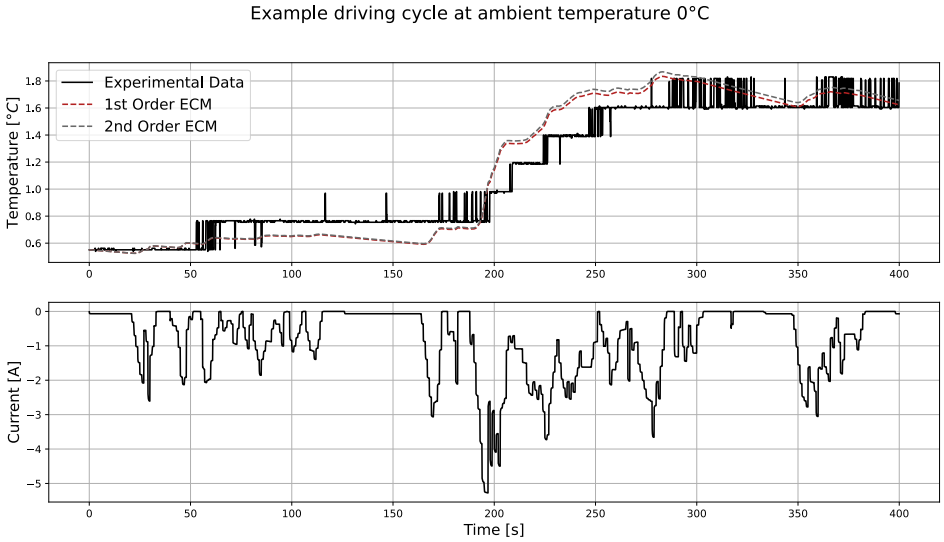


Figure 4.14: First and second order ECM model temperatures compared to the experimental temperature data for a driving cycle.

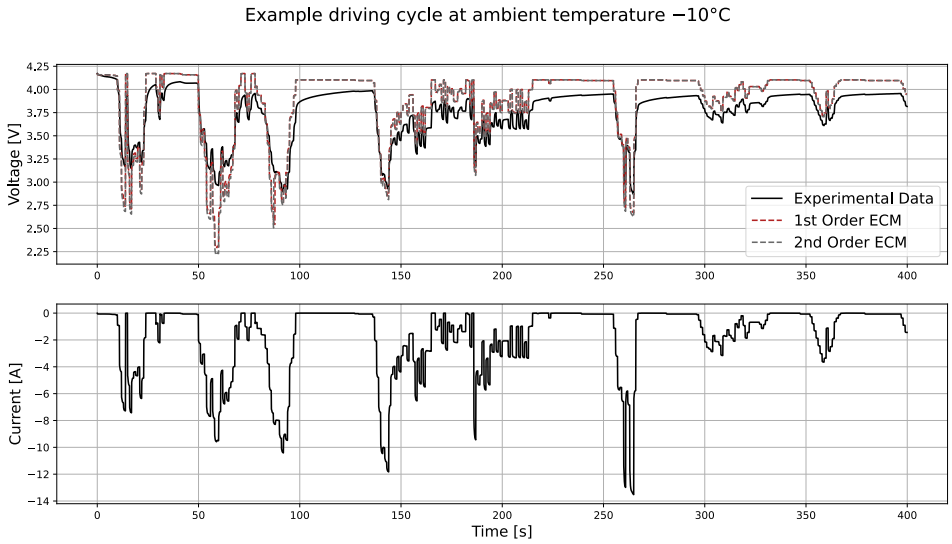


Figure 4.15: The upper figure shows how the actual voltage compares to the output from the first and second order ECM models for an ambient temperature of -10 $°C$.

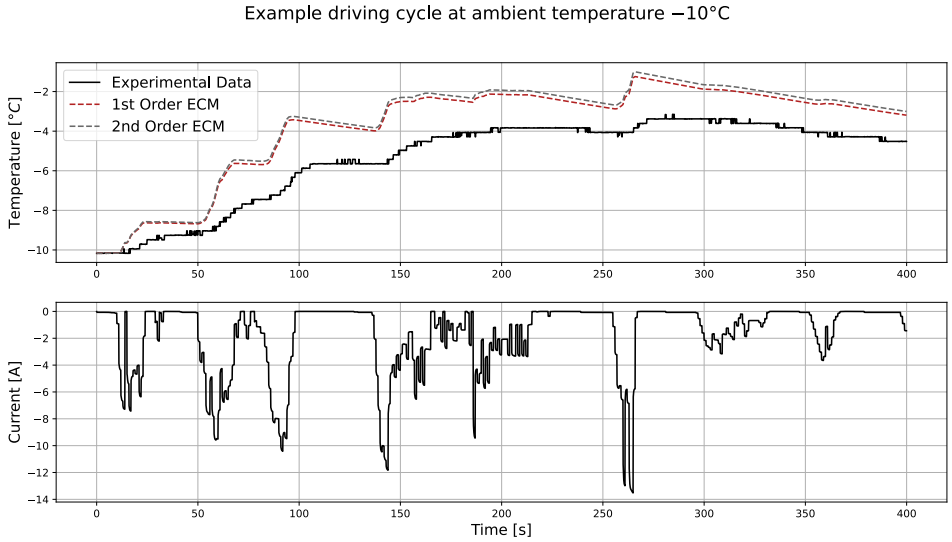


Figure 4.16: First and second order ECM model temperatures compared to the experimental temperature data for a driving cycle.

In figure 4.18 it can be seen that both of the models are displaced from the temperature data.

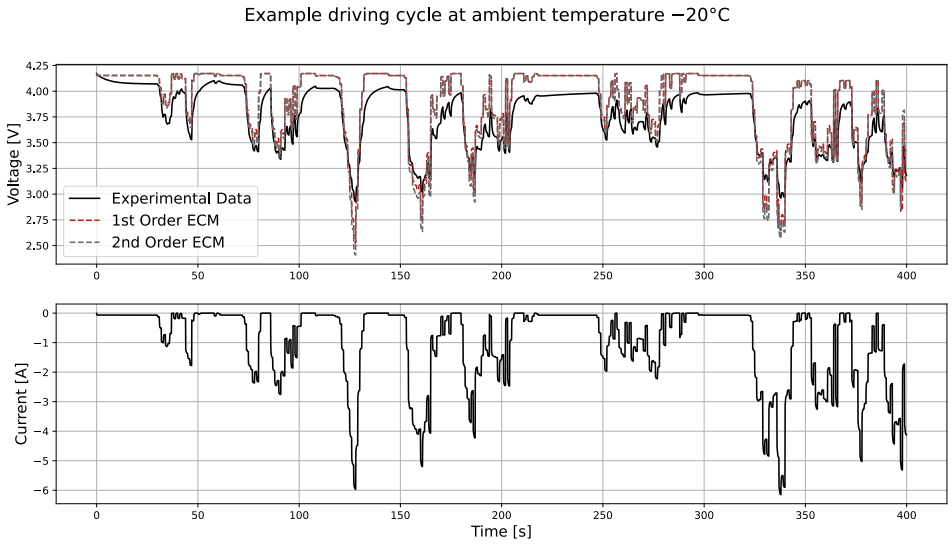


Figure 4.17: The upper figure shows how the actual voltage compares to the output from the first and second order ECM models for an ambient temperature of -20 $°C$.

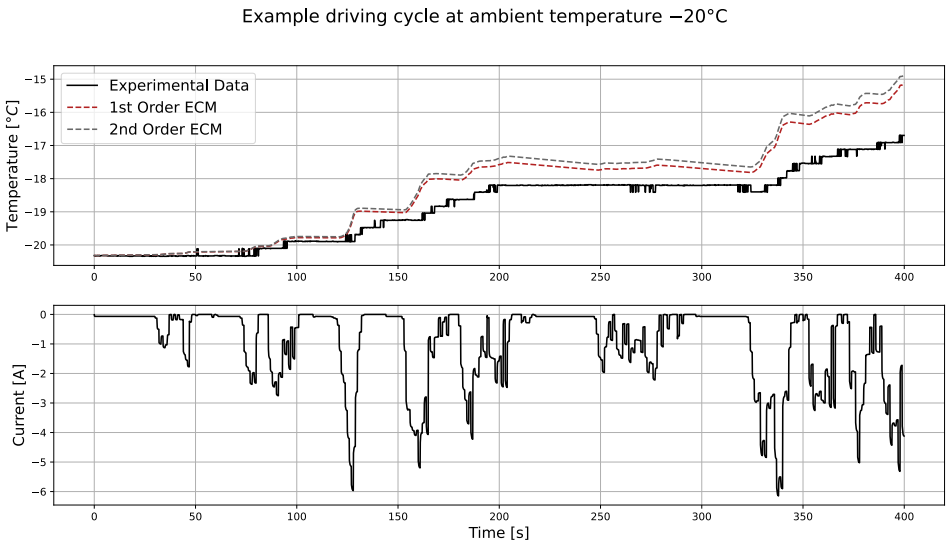


Figure 4.18: First and second order ECM model temperatures compared to the experimental temperature data for a driving cycle.

In figure 4.19 and 4.20 boxplots for a set of driving cycles at different room temperatures are plotted. The orange line within the boxes represents the median error accross all the simulated driving cycles at the corresponding temperature (shown by the x-axis), with the upper and lower bounds of the box are the third quartile (Q_3) and first quartile (Q_1) respectively. The whiskers (the black vertical lines) are represented by the interquartile range (IQR) which is formulated as $IQR = Q_3 - Q_1$. The upper whisker lies at $Q_3 + 1.5IQR$ and the lower whisker is at $Q_1 - 1.5IQR$. Points that lie outside this range (often called fliers) are removed, as the fliers can be extreme and make the figures hard to read. The temperature error for the ambient temperature was $-10°C$ was removed due to bad measurements.

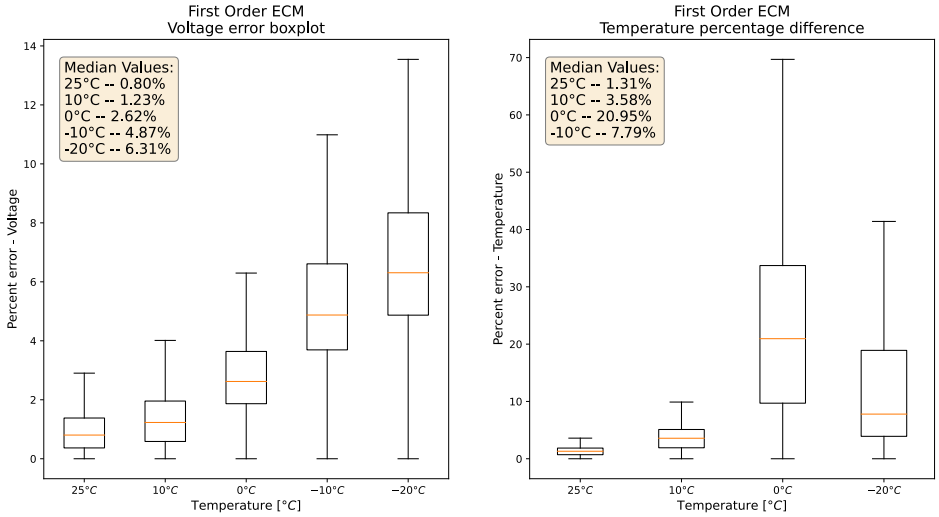


Figure 4.19: Voltage (left) and temperature (right) box-plots for the errors of a set of driving cycles at set room temperatures for the first order ECM model.

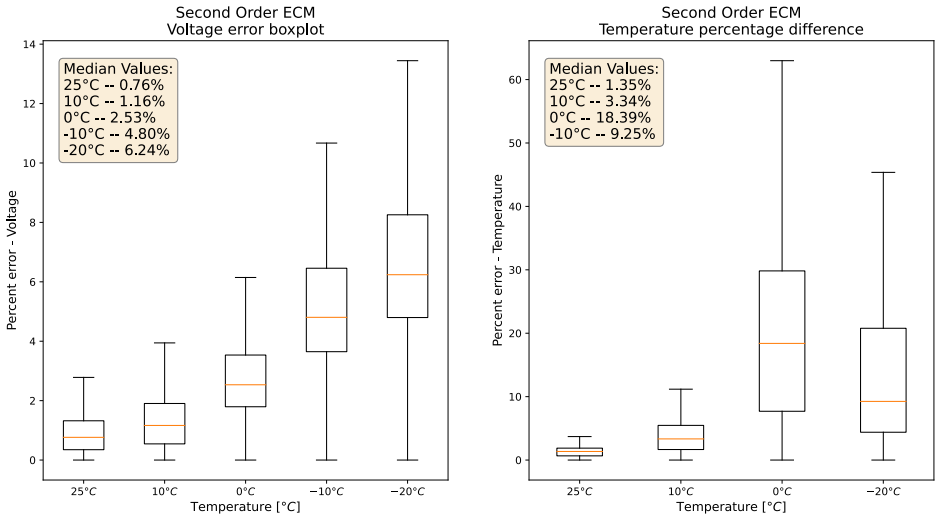


Figure 4.20: Voltage (left) and temperature (right) box-plots for the errors of a set of driving cycles at set room temperatures for the second order ECM model.

4.3 Model Performance

The performance of the two different orders of ECM models were examined by performing repeated simulations for different number of battery cells within a single battery module. The number of cells (n_c) varied between $[1, 135]$, where 135 was used as this is the number of cells within the battery used in modern Volvo personal vehicles.

The purpose of the test was to see how many updates the model can perform during a second as well as to see the average time it takes for the model to update for different n_c values. To do this, the model was feed a constant current $I = 4A$ with a fixed timestep $\Delta t = 0.1$ for a single second. It is important to note that Δt does not correspond to real-time, but instead simulation time. The model was initialized anew before each test to ensure the same starting conditions and therefore a fair test. The test was repeated 10 times for each value of n_c and the results from these 10 runs where averaged. The data that was recorded during these tests where the number of updates achieved during the second and the time each update took.

4.3.1 Raspberry Pi4B - 8GB RAM

These tests were performed on a Raspberry Pi Single Chip computer of model 4B with 8GB of RAM. The Pi4B has a quad-core CPU with a clockspeed of 1.5 GHz. The computer was run without graphical output and accessed remotely using Secure Shell (SSH).

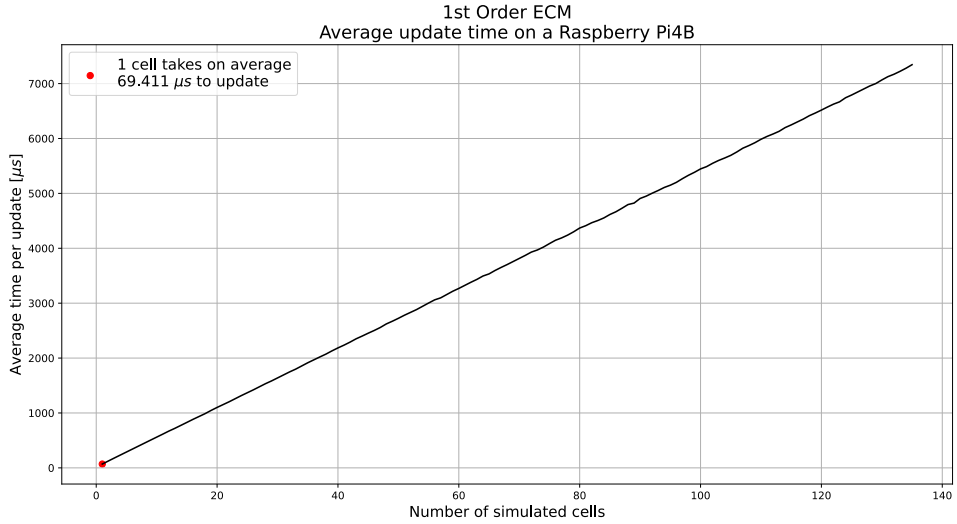


Figure 4.21: Average time for an update of all the cells within the module for different numbers of cells when using the first order ECM.

In figure 4.21 the average update time of the entire battery module is plotted against the number of cells within the module. The red point is marked, as this is the update time for a module consisting of a single cell.

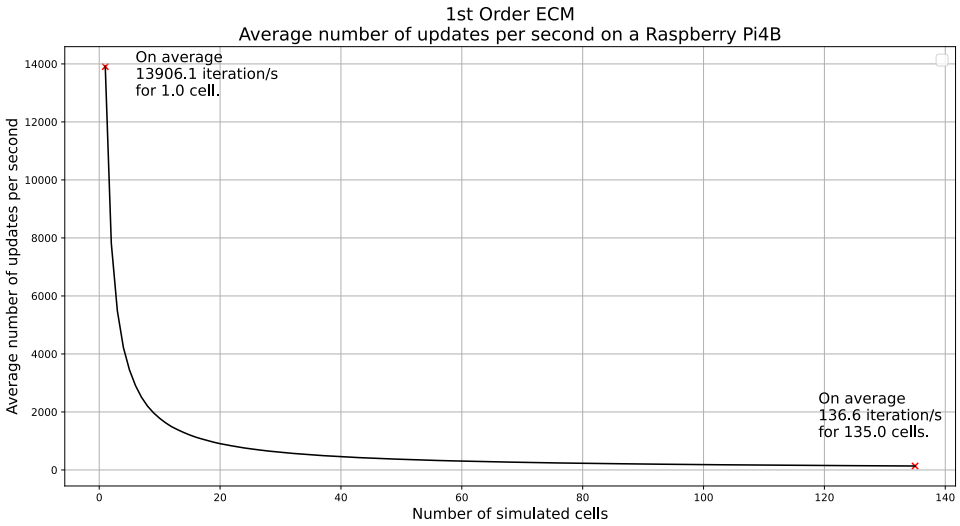


Figure 4.22: Updates per second depending on the number of battery cells within the module, for the first order ECM.

Figure 4.22 plots the average number of updates per second for a battery module consisting of a varying number of battery cells, represented by the x-axis. The marked points show the average updates per second for modules consisting of 1 cell and 135 cells respectively.

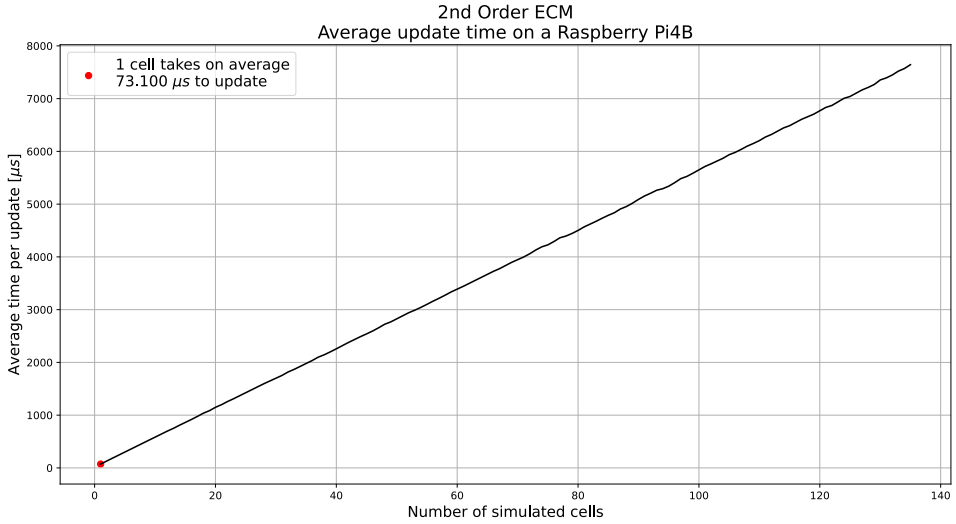


Figure 4.23: Average time an update takes for a battery module consisting of the number of cells represented by the x-axis. This figure is for the second order ECM.

Figure 4.23 is the same plot as 4.21 but for the second order ECM model instead.

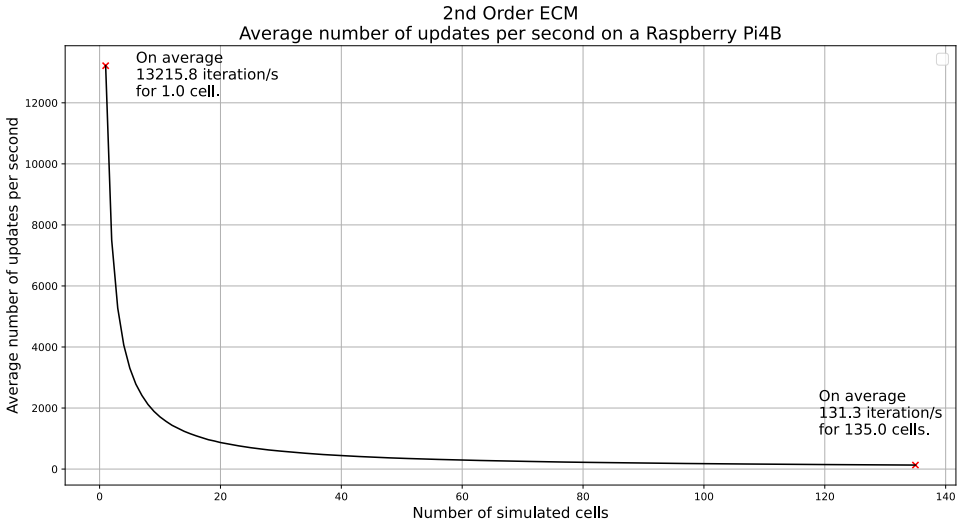


Figure 4.24: Average number of updates per second for a second order ECM battery module containing the number of cells as represented by the x-axis.

Figure 4.24 is the as figure 4.22 above, but once again for the second order ECM instead of the first.

5
Discussion

The goal of the thesis was to develop a battery model that is able to run on embedded computers which are often lacking in 'horsepower' in comparison to other types of computers, while being accurate enough in its predictions as to be able to use it for development of a BMS.

To evaluate this goal two primary questions have to be asked. Firstly, is it accurate enough? If the model is not accurate enough or if the model responds in ways that break the 'logic' of a battery, it cannot be used to 'fool' a BMS. Secondly, is the model quick enough to be able to run the simulations in real time? It is paramount for the model to be able to simulate quickly enough to fool the BMS into thinking that it is being fed actual sensor data.

5.1 Model Accuracy

The verify the accuracy of the model, the error of the model compared to the experimental data was used (see equation 4.1) as the primary way of confirming the accuracy. By looking at figures 4.19 and 4.20 we can see that both the first and second order models simulate the voltage response based on some applied current pull well for the cases where the ambient temperature is above or at 0 $°C$, but the performance starts to degrade below these temperatures. The second order model also has a slight advantage in median error for all temperature ranges, which is to be expected due to the second exponential term, as it allows the model to more accurately capture the non-linear behaviours of a battery.

In figure 4.5 we can see that first and second order models almost completely overlap, with there being small differences along the way. This is to be expected, as the difference between the two boxplots of the errors (see figures 4.19 and 4.20), as the results for the ambient temperature of 25 °C are very similar. If we look at figure 4.2, we see that the model seems to mimic the behaviour of the experimental data very well. The major difference (and source of error) seems to be some displacement on the voltage values. This can be explained by the extraction of the OCV-values being imperfect, as the assumption that battery has reached a chemical steady-state before each pulse seems to be faulty. This can be rectified by performing experiments that specifically extract the OCV-values, instead of the OCV values being extracted in tandem with the model parameters. Another possible problem is that the models only use OCV-values extracted at the ambient temperature 25 $°C$. This is due to the fact that the OCV has been shown to vary with both SoC and temperature, which means that the model might be able to encapsulate the right behaviour, but the output is wrong to due to the OCV-data [17].

When looking at figure 4.2, it can be seen that at around the 310 second mark there is a spike in error. This spike in error seems to be related to how quickly the model reacts to the current change as compared to the actual battery. Seemingly, the model reacts quicker by some margin, leading to a spike which then rapidly disappears. This might be connected to the intrinsic resistance (R_Ω), as the instantaneous voltage change is connected to this term. Another possibility is that some type of delay has to be implemented into the model, if this spike is to be removed.

On average, the models perform admirably well in regards to the voltage simulations but seem to lack the same precision for the temperature simulation. This is possibly due to the used parameters in combination with a naive solution for the heat model. For example, in figure 4.18 it can be seen that the temperature plot mimics the behaviour of the experimentally found data to some degree, while being displaced in the y-axis. It can also be seen from this figure that while the simulation mimics the behaviour, it is also more 'extreme', as the applied current gives rise to higher increases in temperature within the model as compared to the experimental data. The first order model has a slightly lower temperature, which implies that $\Delta Q_1 \leq \Delta Q_2$, where

$$\Delta Q_1 = |U_\Omega I| + |U_\tau I|$$

for the first order model and

$$\Delta Q_2 = |U_\Omega I| + |U_1 I| + |U_2 I|$$

for the second order model. This implies that the sum total of the potentials generated for the second order model is slightly higher at points compared to the first order model. As the model seems to be running hot in comparison to the data, the exclusion of the reversible heat generation term might come into question. This could theoretically cool down the output temperatures from the model, which would present better values. The thermal parameters were also rather naively extracted as the mean of a range of possible values. The convection term h is also dependent on temperature, which is not taken into account within the model.

Due to problems with the emulation software at Volvo, the model could not be tested in conjunction with the BMS, which means that any discussion about if the obtained results are accurate enough for BMS development will be speculation instead of verifiable. The voltage outputs obtained and presented in the previous chapter shows that the general 'logic' or 'behaviour' of the battery is encapsulated, with the differences lying in the amplitude of change due to the applied current and in displacement due to the imperfect OCV value extraction. Without reference to the input data the obtained results seem fully reasonable, as the simulation does not display erratic or volatile behaviour. From this, it can be speculated that the model portrays the behaviour accurately enough (even if the actual values are wrong) that the model can fed the BMS with data that will not trigger any of the fail safes, unless that was the purpose of the data.

5.2 Model Performance

From figures 4.21 and 4.23 it can be seen that the update time increases linearly with the number of cells within the module. From this we can reason that the simulation model is

not being bottle-necked for performance on the Raspberry Pi 4B. If the behaviour had not been linear, this would imply that the model does not scale properly in the performance domain, i.e that the model needs a more computationally powerful computer.

As expected, the slightly more complex second order model is more computationally expensive. Each second order cell is on average 4 μs or 5.5% slower per update. When simulating the maximum number of cells used within Volvo car batteries (135 cells total, with 9 modules with 15 cells), the first and second order models reach 136.6 and 131.3 updates per second on average, respectively. As the performance tests were performed on a single module containing all the cells, the actual performance will be slightly slower due to the fact that more memory locations will have to be accessed.

To determine if the simulations have achieved real time simulation, the meaning of what 'real time simulation' means has to be defined. One way of doing this is by examining the purpose and use cases from the situations the simulation was developed. For this project, the simulation was developed for BMS testing. The BMS is connected to a set of emulators, which emulate the computer chips that measure the voltage, current and temperature. The simulation will be feeding these chip emulators with the needed data. According to the engineers and developers of Volvo, the chips update at 50 and 100 ms, leading to 20 and 10 updates per second, respectively. As the simulation will be giving information to both chips at the same time, irrespective of when they need the information, the lower bound (20 updates per second) will be used as a reference.

The update frequencies that were obtained from the naive testing show that both of the models surpass the lower bound of 20 updates per second easily, with both orders of the model being over 6 times faster than the needed speed. From this, we can state that the model runs at or faster than real time for the use case presented in this thesis.

5.3 Future Work and Model Expansions

There are a few key points that could be improved or expanded upon. Firstly, performing separate experiments to determine the SoC and temperature dependence of the parameters and OCV, with additional measuring points for the OCV especially. This could possibly eliminate the displacement seen from the performed simulations, while allowing for better accuracy due to the added granularity in terms of SoC and temperature.

Secondly, the thermal model can be expanded. For car battery simulation, there are cooling/heating systems install that add or remove heat to battery cells based on the environment the car finds it self in. This would mean further investigation into how the heating systems work, with heat transport happening through cooling fluid or heating through electrical engines in cold environments.

The effects that adding the reversible heating back into the temperature modeling could also be examined, especially in terms of how the error would change.

As the model runs above what is needed, another possibility would be to downscale the equipment that is being used, i.e. use a less powerful and expensive embedded computer.

The embedded computer will need to have some type of OS (some Linux distro possibly) due to the fact that Python code is not compiled and translated into machine code, but requires the use of a Python interpreter, which adds both CPU and RAM requirements. Other than this, it is entirely possible that the model is able to run at real time on a significantly less powerful computer. One possibility is that the model will not scale linearly for these weaker machines, but this would have to be tested.

5.4 Ethical Concerns

Due to the nature of batteries, as they use sometimes rare or hard to extract materials, it is important to make sure that the materials are ethically sourced. Battery performance will degrade over time and with use, but the use of a BMS ensure that the battery's lifetime is as long as possible, while ensuring maximum performance. This means that the materials are used for a much bigger part of their total potential, which in turn means that there will be less potential material waste.

The battery management systems that are used within EVs are not only for better performance and efficiency of the employed battery solution, but also for the safety of the user. Li-Ion batteries are more thermally volatile than other types of batteries, and there have been many cases of the batteries exploding or frying the electronics attached to them. To minimize and manage the risks of these types of situations, safety measures and systems have to be thoughtfully implemented, as the consequences for these types of failures can in the context of EVs be lethal.

6
Conclusion

The accuracy of the model varies based on which circumstances that are being simulated, with the best performance being for ambient temperatures between 10°C and 25°C. The second order model performs slightly better in voltage accuracy as it is able to capture non-linear behaviours with the secondary exponential term, but it also has slightly worse behaviour in the temperature simulation as total sum of the simulated voltage is higher for the second order model in comparison to the first.

The behaviour of both the first and second order models seem very similar to that of the experimental data, with there being displacements in the voltage curve and that the amplitude of the current response varies from the found data. No erratic or illogical voltage spikes have been observed during testing, which lends us to conclude that the model can be used for BMS development, even if this cannot be verified due to problems with the emulated software at Volvo.

The parameters that have been extracted work well, with there being a slight deviation in the OCV-values. The assumption that the battery reaches chemical-equilibrium before each pulse in the HPPC test seems to be not entirely true, as the voltage values are slightly shifted in comparison to the experimental data when simulated.

The computational performance of the model is good, as it is able to update at frequencies higher than 100 Hz for both types of the model running on a Raspberry Pi 4B. There were no detected bottlenecks, neither in computational performance or in memory on the device and with the high update speed, the model is able to run in real time within BMS development context, as both of the models update quicker than the required 50 ms.

Based on the obtained accuracy and performance, the first order ECM is the better choice for BMS development, when running on an embedded computer. The performance gained from using the first order model is much larger than the accuracy increases that is achieved by using the second order model. The additional performance could allow for a more complex thermal model, while ensuring that the model can still update in real time.

Bibliography

[1] *Greenhouse gas emissions from a typical passenger vehicle*, https://www.epa.gov/greenvehicles/greenhouse-gas-emissions-typical-passenger-vehicle, Accessed: 2022-03-08.

[2] *Co2 emissions from cars: Facts and figures*, https://www.europarl.europa.eu/news/en/headlines/society/20190313STO31218/co2-emissions-from-cars-facts-and-figures-infographics, Accessed: 2022-03-08.

[3] M. S. Ziegler and J. E. Trancik, "Re-examining rates of lithium-ion battery technology improvement and cost decline," *Energy & Environmental Science*, vol. 14, no. 4, pp. 1635–1651, 2021.

[4] Y. Ding, Z. P. Cano, A. Yu, J. Lu, and Z. Chen, "Automotive li-ion batteries: Current status and future perspectives," *Electrochemical Energy Reviews*, vol. 2, no. 1, pp. 1–28, 2019.

[5] A. J. Bard and L. R. Faulkner, *Electrochemical methods: fundamentals and applications, 2nd Ed.* John Wiley & Sons, Inc, 2001.

[6] B. Scrosati, J. Hassoun, and Y.-K. Sun, "Lithium-ion batteries. a look into the future," *Energy & Environmental Science*, vol. 4, no. 9, pp. 3287–3295, 2011.

[7] *Lithium-ion battery*, https://www.cei.washington.edu/education/science-of-solar/battery-technology/, Accessed: 2022-03-14.

[8] Y. Miao, P. Hynan, A. von Jouanne, and A. Yokochi, "Current li-ion battery technologies in electric vehicles and opportunities for advancements," *Energies*, vol. 12, no. 6, 2019, ISSN: 1996-1073. DOI: 10.3390/en12061074. [Online]. Available: https://www.mdpi.com/1996-1073/12/6/1074.

[9] X. Chen, W. Shen, T. T. Vo, Z. Cao, and A. Kapoor, "An overview of lithium-ion batteries for electric vehicles," in *2012 10th International Power & Energy Conference (IPEC)*, IEEE, 2012, pp. 230–235.

[10] G. E. Blomgren, "The development and future of lithium ion batteries," *Journal of The Electrochemical Society*, vol. 164, no. 1, A5019, 2016.

[11] M. Lowe, S. Tokuoka, T. Trigg, and G. Gereffi, "Lithium-ion batteries for electric vehicles," *The US Value Chain, Contributing CGGC researcher: Ansam Abayechi*, 2010.

[12] *Lithium-ion batteries: How can thermal runaway be prevented?* https://www.rutronik.com/article/detail/News/lithium-ion-batteries-how-can-thermal-runaway-be-prevented/, Accessed: 2022-03-14.

[13] W. He, M. Pecht, D. Flynn, and F. Dinmohammadi, "A physics-based electrochemical model for lithium-ion battery state-of-charge estimation solved by an optimised projection-based method and moving-window filtering," *Energies*, vol. 11, no. 8, p. 2120, 2018.

[14] J. Meng, M. Boukhnifer, and D. Diallo, "Comparative study of lithium-ion battery open-circuit-voltage online estimation methods," *IET Electrical Systems in Transportation*, vol. 10, no. 2, pp. 162–169, 2020.

[15] S. Yang, C. Deng, Y. Zhang, and Y. He, "State of charge estimation for lithium-ion battery with a temperature-compensated model," *Energies*, vol. 10, no. 10, p. 1560, 2017.

[16] Z. Lu, X. Yu, L. Wei, F. Cao, L. Zhang, X. Meng, and L. Jin, "A comprehensive experimental study on temperature-dependent performance of lithium-ion battery," *Applied Thermal Engineering*, vol. 158, p. 113 800, 2019.

[17] A. Farmann and D. U. Sauer, "A study on the dependency of the open-circuit voltage on temperature and actual aging state of lithium-ion batteries," *Journal of Power Sources*, vol. 347, pp. 1–13, 2017.

[18] L. Zhang, H. Peng, Z. Ning, Z. Mu, and C. Sun, "Comparative research on rc equivalent circuit models for lithium-ion batteries of electric vehicles," *Applied Sciences*, vol. 7, no. 10, p. 1002, 2017.

[19] W.-Y. Chang, "The state of charge estimating methods for battery: A review," *International Scholarly Research Notices*, vol. 2013, 2013.

[20] M. Ruba, R. Nemeș, S. Ciornei, and C. Marțiș, "Parameter identification, modeling and testing of li-ion batteries used in electric vehicles," in *Applied electromechanical devices and machines for electric mobility solutions*, IntechOpen, 2020.

[21] S. S. Madani, E. Schaltz, and S. Knudsen Kær, "An electrical equivalent circuit model of a lithium titanate oxide battery," *Batteries*, vol. 5, no. 1, p. 31, 2019.

[22] S.-T. Ko, J.-H. Ahn, and B. K. Lee, "Enhanced equivalent circuit modeling for li-ion battery using recursive parameter correction," *Journal of Electrical Engineering and Technology*, vol. 13, no. 3, pp. 1147–1155, 2018.

[23] C. Lin, C. Cui, and X. Xu, "Lithium-ion battery electro-thermal model and its application in the numerical simulation of short circuit experiment," *World Electric Vehicle Journal*, vol. 6, no. 3, pp. 603–610, 2013.

[24] N. H. F. Ismail, S. F. Toha, N. A. M. Azubir, N. H. M. Ishak, M. K. Hassan, and B. S. K. Ibrahim, "Simplified heat generation model for lithium ion battery used in electric vehicle," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 53, 2013, p. 012 014.

[25] Q. Zhang, F. Wei, P. Zhang, R. Dong, J. Li, P. Li, Q. Jia, Y. Liu, J. Mao, and G. Shao, "Research on the reversible and irreversible heat generation of lini1- x-ycoxmnyo2-based lithium-ion batteries," *Fire Technology*, pp. 1–21, 2022.

[26] J. J. Moré, "The levenberg-marquardt algorithm: Implementation and theory," in *Numerical analysis*, Springer, 1978, pp. 105–116.

[27] T. T. Lou, W. G. Zhang, H. Y. Guo, and J. S. Wang, "The internal resistance characteristics of lithium-ion battery based on hppc method," in *Advanced materials research*, Trans Tech Publ, vol. 455, 2012, pp. 246–251.

[28] P. Kollmeyer, "Panasonic 18650pf li-ion battery data," *Mendeley Data*, vol. 1, no. 2018, 2018.

[29] G. Dos Reis, C. Strange, M. Yadav, and S. Li, "Lithium-ion battery data and where to find it," *Energy and AI*, vol. 5, p. 100 081, 2021.

[30] *Ncr18650pf*, `https://na.industrial.panasonic.com/products/batteries/rechargeable-batteries/lineup/lithium-ion/series/90729/model/90730`, Accessed: 2022-05-02.

[31] P. Keil, K. Rumpf, and A. Jossen, "Thermal impedance spectroscopy for li-ion batteries with an ir temperature sensor system," in *2013 World Electric Vehicle Symposium and Exhibition (EVS27)*, IEEE, 2013, pp. 1–11.

A

Appendix 1

A.1 Extracted ECM model parameters

This part of the appendix contains all the parameters that were extracted from the battery data that was used for the creation and testing of the model [28]. Parameters which could not be extracted are (due to the lack of measured data) are symbolized by the shebang symbol #!.

A.1.1 First Order

Table A.1: Extracted values for the R_Ω parameter in the first order ECM.

R_Ω						
SoC	° C	25	10	0	-10	-20
1.0		0.040	0.081	0.133	0.195	0.272
0.95		0.036	0.063	0.105	0.160	0.234
0.9		0.034	0.055	0.091	0.146	0.220
0.8		0.033	0.049	0.077	0.127	0.195
0.7		0.032	0.048	0.074	0.118	#!
0.6		0.032	0.045	0.070	0.114	0.187
0.5		0.030	0.043	0.069	0.115	0.190
0.4		0.031	0.045	0.074	0.122	0.201
0.3		0.032	0.050	0.086	0.141	0.220
0.25		0.034	0.056	0.100	0.156	0.228
0.2		0.037	0.073	0.121	0.168	#!
0.15		0.046	0.098	0.135	#!	#!
0.1		0.068	0.114	#!	#!	#!

Table A.2: C_τ for ECM first order.

C_τ						
SoC	° C	25	10	0	-10	-20
1.0		1252.909	1157.572	868.526	442.292	221.151
0.95		1254.288	1100.527	961.104	601.517	313.225
0.9		1157.803	939.777	820.834	618.103	358.046
0.8		1049.284	867.400	764.421	644.337	388.778
0.7		1025.603	867.400	778.229	673.325	#!
0.6		1075.103	1026.114	884.584	673.034	392.141
0.5		1457.127	1156.533	960.529	693.864	369.769
0.4		1456.108	1156.648	940.058	601.337	273.389
0.3		1410.000	1074.781	716.450	317.711	107.356
0.25		1328.404	884.760	421.646	146.007	39.638
0.2		1187.518	518.664	149.396	52.730	#!
0.15		885.379	149.396	51.661	#!	#!
0.1		313.509	55.750	#!	#!	#!

Table A.3: R_τ for the first order ECM.

R_τ						
SoC	°C	25	10	0	-10	-20
1.0		0.008	0.009	0.012	0.023	0.045
0.95		0.008	0.009	0.010	0.017	0.032
0.9		0.009	0.011	0.012	0.016	0.028
0.8		0.010	0.012	0.013	0.016	0.026
0.7		0.010	0.012	0.013	0.015	#!
0.6		0.009	0.010	0.011	0.015	0.026
0.5		0.007	0.009	0.010	0.014	0.027
0.4		0.007	0.009	0.011	0.017	0.037
0.3		0.007	0.009	0.014	0.032	0.093
0.25		0.008	0.011	0.024	0.069	0.097
0.2		0.008	0.019	0.067	0.148	#!
0.15		0.011	0.067	0.161	#!	#!
0.1		0.032	0.171	#!	#!	#!

A.1.2 Second Order

Table A.4: Extracted R_0 values for second order ECM.

R_0						
SoC	°C	25	10	0	-10	-20
1.0		0.044	0.084	0.136	0.204	0.295
0.95		0.039	0.067	0.109	0.167	0.250
0.9		0.038	0.060	0.097	0.153	0.233
0.8		0.037	0.054	0.083	0.133	0.207
0.7		0.037	0.053	0.080	0.124	#!
0.6		0.037	0.050	0.075	0.120	0.198
0.5		0.034	0.047	0.073	0.121	0.202
0.4		0.034	0.049	0.078	0.129	0.218
0.3		0.035	0.054	0.092	0.157	0.244
0.25		0.037	0.061	0.112	0.188	0.238
0.2		0.041	0.082	0.155	0.198	#!
0.15		0.052	0.132	0.169	#!	#!
0.1		0.084	0.152	#!	#!	#!

Table A.5: R_1 for second order ECM model.

R_1						
SoC	C	25	10	0	-10	-20
1.0		1.63e-08	1.86e-08	2.56e-08	7.34e-08	3.66e-08
0.95		1.61e-08	1.98e-08	2.38e-08	4.21e-08	4.25e-08
0.9		1.76e-08	2.27e-08	2.71e-08	4.09e-08	3.88e-08
0.8		1.94e-08	2.55e-08	3.11e-08	3.98e-08	3.33e-07
0.7		2.03e-08	2.68e-08	3.16e-08	3.86e-08	#!
0.6		1.98e-08	2.16e-08	2.57e-08	4.06e-08	4.01e-08
0.5		1.37e-08	1.83e-08	2.28e-08	3.94e-08	5.35e-04
0.4		1.40e-08	1.82e-08	2.37e-08	6.73e-08	1.32e-02
0.3		1.45e-08	2.10e-08	4.34e-08	1.37e-07	2.38e-01
0.25		1.60e-08	2.89e-08	2.95e-08	5.05e-02	1.36e-01
0.2		1.85e-08	2.22e-08	2.20e-02	3.71e-01	#!
0.15		3.50e-08	1.48e-02	3.95e-01	#!	#!
0.1		1.03e-04	4.13e-01	#!	#!	#!

Table A.6: R_2 for second order ECM.

R_2						
SoC	C	25	10	0	-10	-20
1.0		5.05e-15	1.29e-14	3.59e-13	4.21e-13	2.16e-12
0.95		2.28e-15	1.26e-14	1.54e-13	7.84e-13	2.05e-12
0.9		1.78e-15	6.53e-15	1.77e-14	2.32e-13	2.21e-12
0.8		1.33e-14	1.37e-14	2.15e-13	5.37e-13	6.21e-12
0.7		1.36e-13	1.70e-13	4.40e-13	7.31e-13	#!
0.6		4.04e-13	4.61e-13	2.93e-13	3.16e-13	2.06e-12
0.5		5.86e-11	6.10e-15	5.54e-16	3.86e-13	1.68e-12
0.4		1.12e-14	1.04e-14	2.29e-13	6.32e-13	2.86e-12
0.3		6.69e-11	8.96e-14	1.23e-13	6.78e-13	6.68e-12
0.25		1.38e-10	1.14e-13	8.23e-14	1.09e-12	1.09e-11
0.2		5.56e-15	1.80e-14	5.40e-13	6.96e-10	#!
0.15		9.15e-11	1.80e-13	1.19e-09	#!	#!
0.1		1.48e-13	2.35e-09	#!	#!	#!

Table A.7: Extracted C_1 values for the second order ECM model.

C_1						
SoC	C	25	10	0	-10	-20
1.0		6.76e+12	7.64e+12	6.56e+12	1.53e+12	1.66e+13
0.95		6.50e+12	5.91e+12	5.87e+12	3.15e+12	5.58e+12
0.9		5.83e+12	4.71e+12	4.44e+12	3.07e+12	6.03e+12
0.8		5.11e+12	3.80e+12	3.34e+12	2.92e+12	1.22e+11
0.7		4.72e+12	3.50e+12	3.13e+12	2.88e+12	#!
0.6		4.90e+12	4.57e+12	4.21e+12	2.62e+12	4.98e+12
0.5		7.66e+12	5.73e+12	5.00e+12	2.79e+12	9.03e+21
0.4		7.48e+12	5.81e+12	4.86e+12	1.23e+12	1.03e+19
0.3		7.28e+12	4.88e+12	2.10e+12	1.50e+12	4.36e+15
0.25		6.33e+12	3.24e+12	6.77e+12	1.03e+17	2.61e+13
0.2		5.43e+12	9.04e+12	2.67e+17	2.78e+15	#!
0.15		2.26e+12	4.11e+17	1.21e+15	#!	#!
0.1		1.73e+18	4.31e+14	#!	#!	#!

Table A.8: Extracted C_2 values for the second order ECM model.

C_2						
SoC	C	25	10	0	-10	-20
1.0		8.51e+14	3.33e+14	1.25e+13	1.06e+13	2.07e+12
0.95		1.89e+15	3.41e+14	2.89e+13	5.70e+12	2.18e+12
0.9		2.42e+15	6.58e+14	2.43e+14	1.93e+13	2.02e+12
0.8		3.24e+14	3.13e+14	2.08e+13	8.32e+12	7.21e+11
0.7		3.28e+13	2.63e+13	1.02e+13	6.11e+12	#!
0.6		1.11e+13	9.68e+12	1.52e+13	1.42e+13	2.17e+12
0.5		8.37e+10	7.05e+14	6.91e+15	1.16e+13	2.65e+12
0.4		3.85e+14	4.12e+14	1.95e+13	7.07e+12	1.56e+12
0.3		7.33e+10	4.99e+13	3.64e+13	6.59e+12	6.69e+11
0.25		3.55e+10	3.93e+13	5.43e+13	4.10e+12	4.49e+11
0.2		7.73e+14	2.39e+14	8.27e+12	8.23e+09	#!
0.15		5.36e+10	2.47e+13	4.82e+09	#!	#!
0.1		3.01e+13	2.44e+09	#!	#!	#!

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY