Unterschrift BetreuerIn

TECHNISCHE UNIVERSITÄT WIEN

DIPLOMARBEIT

Development of an electric vehicle and battery pack simulator

ausgeführt am Institut für Angewandte Physik
der Technischen Universität Wien
und am
Austrian Institute of Technology GmbH

unter der Anleitung von
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Martin Gröschl
Dipl.-Ing. Daniel Stahleder
Stephan Ledinger

durch

Petra Reisz

[redacted]

July 3, 2024

Unterschrift StudentIn

Kurzfassung

Die Integration erneuerbarer Energiequellen ist eine Herausforderung für die Energiewirtschaft der Zukunft. Als zusätzlicher Puffer zwischen Stromangebot und -nachfrage sind Batterien ein Kandidat, der zunehmend an Bedeutung gewinnt. Dies wird durch ihre rasch sinkenden Kosten und ihre Flexibilität begünstigt. Sie werden eine wichtige Rolle bei der Unterstützung der Netzstabilität und -flexibilität spielen. Nicht nur die Energiegewinnung, sondern auch der Straßenverkehr wird stark elektrifiziert: Der Verkauf von Elektroautos nimmt rapide zu, so dass deren Aufladung die Stromnetze erheblich belasten kann.

Es besteht daher ein Bedarf an leistungsfähigen Batteriemodellen in Bezug auf Geschwindigkeit und Genauigkeit. Diese können für die Entwicklung, das Testen und die Simulation fortschrittlicher Ladealgorithmen für Elektrofahrzeuge, für die Planung moderner Energienetze mit kompletten Batteriespeichersystemen sowie für die Charakterisierung neu entwickelter Batterien und die Vorhersage ihres Verhaltens in verschiedenen Szenarien eingesetzt werden. Die vorliegende Arbeit soll einen Beitrag zur Lösung dieser Probleme leisten.

Die folgende Arbeit stellt eine komplette Simulationslösung eines Batteriesystems und die Grundlagen eines Elektrofahrzeug-Emulators basierend auf physikalischen Batteriemodellen vor. Insbesondere werden Simulationsmethoden zur Nachbildung von Schnellladealgorithmen für Elektrofahrzeuge entwickelt.

Zunächst werden die thermodynamischen Prinzipien der elektrochemischen Zellen, die die Grundlage dieser Technologien bilden, zusammengefasst. Es folgt eine Erläuterung der Prozesse, die bei der Degradation von Batterien ablaufen.

Der nächste theoretische Punkt der Arbeit ist ein Überblick über die wichtigsten internationalen Normen zum Laden von Elektrofahrzeugen. Es folgt eine Zusammenfassung der Grundlagen klassischer und neuer Ladealgorithmen für Elektrofahrzeuge. Anschließend werden moderne Batterietechnologien in Elektrofahrzeugen vorgestellt. Der letzte Teil des zweiten Kapitels ist der Aufgabe gewidmet, die bekanntesten Modellierungsansätze für Li-Ionen-Batterien vorzustellen und zu bewerten.

Aufbauend auf den theoretischen Grundlagen von Kapitel 2 wird in Kapitel 3 das Konzept der Simulationsansätze Schritt für Schritt entwickelt. Dies beginnt mit einer genauen Beschreibung des in dieser Arbeit verwendeten Modells.

Es wird eine Methode zur Parametrisierung dieses Modells entwickelt. Anschließend wird eine neue Methode vorgestellt, die auf adaptiver Polynominterpolation basiert, um mathematisch glatte funktionale Abhängigkeiten zwischen dem Ladezustand der Batterie und den Werten der Komponenten ihres Ersatzschaltbildes herzustellen.

Nach der Erstellung eines genauen Modells für die Simulation des transienten Verhaltens einzelner Zellen werden Ansätze für die Skalierung von Zellen auf der Ebene des Batteriepacks vorgestellt. Diese nutzen sowohl die Linearität der ohmsch-kapazitiven

Ersatzschaltbilder als auch die Differentialgleichungen, die diese beschreiben aus.

Da es nun möglich ist, nicht nur einzelne Batteriezellen, sondern auch ganze Batteriepacks zu simulieren, werden die von den (Schnell-)Ladealgorithmen vorgegebenen Bedingungen in die Simulation des Ladeverhaltens selbst integriert. Es sind mehrere verbreitete Ladealgorithmen implementiert.

Ein Vergleich von gemessenen und simulierten Spannungsprofilen von Batteriezellen zeigt, dass die entwickelten Modelle sehr genaue Ergebnisse liefern, sobald die entsprechenden Eingangsparameter verfügbar sind. Ein Vergleich mit traditionellen Methoden (nicht-adaptive Polynomanpassung und Lookup-Tabellen) wird ebenfalls durchgeführt und es zeigt sich, dass die in dieser Arbeit entwickelten Modelle den herkömmlichen Methoden überlegen sind.

Im zweiten Teil von Kapitel 4 werden die Modelle zur Simulation des Schnellladeverhaltens von Elektrofahrzeugen validiert. Zu diesem Zweck werden öffentlich verfügbare Ladekurven verwendet und nachsimuliert. Die Simulationsdaten werden nachbearbeitet und die resultierenden Daten werden mit den Herstellerangaben für die jeweiligen Fahrzeuge verglichen, um eine weitere Validierungsmethode zu erhalten.

Am Ende der Arbeit werden weitere Empfehlungen und mögliche Anwendungen für das entwickelte Tool gegeben.

Abstract

The integration of renewable energy sources is a challenge for the energy industry of the future. As an additional buffer between electricity supply and demand, batteries are an increasingly important candidate. This is facilitated by their rapidly decreasing cost and flexibility. They will play an important role in supporting grid stability and flexibility. In addition to power generation, road transport is becoming increasingly electrified: Sales of electric cars are growing rapidly, which means that charging them can put a significant strain on power grids.

As a result, there is a need for efficient battery models in terms of speed and accuracy. These can be used for the development, testing and simulation of advanced charging algorithms for electric vehicles, for the planning of modern energy grids that include complete battery storage systems, and for the characterization of newly developed batteries and the prediction of their behavior in different scenarios. This work is intended to contribute to the solution of these problems.

The following work presents a complete simulation solution of a battery system and the basics of an electric vehicle emulator based on physical battery models. In particular, simulation methods are developed to simulate fast charging algorithms for electric vehicles.

First, the thermodynamic principles of electrochemical cells that form the basis of these technologies are summarized. This is followed by an explanation of the processes involved in battery degradation.

The next theoretical point of the thesis is an overview of the most important international standards for charging electric vehicles. This is followed by a summary of the basics of classical and new charging algorithms for electric vehicles. Modern battery technologies in electric vehicles are then presented. The last part of the second chapter is dedicated to the task of presenting and evaluating the best known modeling approaches for Li-ion batteries.

Based on the theoretical foundations of Chapter 2, the concept of the simulation approaches is developed step by step in Chapter 3. This starts with a precise description of the model used in this thesis.

A method to parameterize this model is developed. Then a new method based on adaptive polynomial interpolation is presented to establish mathematically smooth functional dependencies between the state of charge of the battery and the values of the components of its equivalent circuit diagram.

After establishing an accurate model for simulations of the transient behavior of individual cells, approaches for scaling cells at the battery pack level are presented. These exploit both the linearity of the resistive-capacitive equivalent circuit diagrams and the differential equations that describe them.

Since it is now possible to simulate not only individual battery cells but also entire

battery packs, the conditions specified by the (fast) charging algorithms are integrated into the simulation of the charging behavior itself. Several commonly used charging algorithms are implemented.

A comparison of measured and simulated battery cell voltage profiles shows that the developed models deliver very accurate results as soon as the corresponding input parameters are available. A comparison with traditional methods (non-adaptive polynomial fitting and lookup tables) is also made and it is shown that the models developed in this thesis are superior to the conventional methods.

In the second part of Chapter 4, the models for simulating the fast charging behavior of electric vehicles are validated. For this purpose, publicly available charging curves are used and re-simulated. The simulation data are post-processed and the resulting data are compared with the manufacturer's specifications for the respective vehicles to provide a further validation method.

At the end of the thesis, further recommendations and possible applications for the developed tool are given.

Contents

Acronyms

SoC State of Charge

F Faraday's constant

e Elementary charge

n_e Number of electrons taking place in redox reaction

EV Electric vehicle

EVSE Electrical Vehicle Supply Equipment

LIB Lithium Ion Battery

SSB Solid State Battery

BMS Battery Management System

EEC Electric Equivalent Circuit (Model)

V_{oc} Open-circuit voltage

X Components of the first-order electric equivalent circuit ($\{V_{oc}, R_0, R_1, C_1\}$)

MCC Multiple stage Constant Current charging method

BC Boost Charging method

CV Continous Voltage charging method

CC Constant Current charging method

(x) Referencing equation x

1 Introduction

1.1 Electricity, renewables, battery storage systems and electric vehicles: trends and challenges

The global demand for electricity is on the rise. According to the International Energy Agency's (IEA) most recent report, the global electricity demand is expected to rise at a faster rate over the next three years, growing by an average of 3.4 % annually until 2026 [1]. It predicts a faster electricity demand growth both in advanced and emerging economies, strongly supported by the ongoing electrification of the residential-, transport- and the data centre sector. The latter is also caused due to the rising demand for the computing power of artificial intelligence. It is stated in the report that the share of electricity in final energy consumption was around 20 % in 2023, experiencing a significant rise compared to 2015.

The IEA's "Net Zero Emissions by 2050 Scenario" (NZE) is a normative scenario that shows a pathway for the global energy sector to achieve net zero CO_2 emissions by 2050 [2]. It meets United Nations key energy-related Sustainable Development Goals [3]. NZE is in consistency with emissions reduction statements assessed in the Intergovernmental Panel on Climate Change's "Sixth Assessment Report" whose measures are set to limit the global temperature rise to 1.5 °C with at least 50% probability [4]. According to the NZE scenario, the share of electricity in final energy consumption must reach about 30% by 2030. In addition, renewables are projected to account for more than one-third of total global electricity generation by early 2025, with growth largely driven by the deployment of increasingly affordable solar photovoltaic systems. However, expanding the renewable energy capacity requires significant investments in grid infrastructure and system flexibility to ensure a seamless integration [2].

These requirements become complex due to the variability of electricity generated from renewable sources, which undergoes daily and seasonal fluctuations. They are often misaligned with present electricity needs [5]. Moreover, renewables power generation is temporary. These issues can be addressed by the utilization of battery storage systems [6] and the integration of electric vehicles [7] into the energy grid. In the recent past, a rapidly increased deployment of photovoltaic and battery installations has been observed in the residential sector [8].

Therefore, it is assumed that the energy production will undergo a transition: The necessary stationary storage for energy from renewable sources will increase the demand for batteries. This will amount to about 400-500 GWh by 2030 [9]. For these purposes, lithium-ion technology is still the most widely used battery type. This is due to its high power- and energy density as well as excellent cycling capabilities [10]. Battery chemistries are also expected to continue to evolve and innovate technologically.

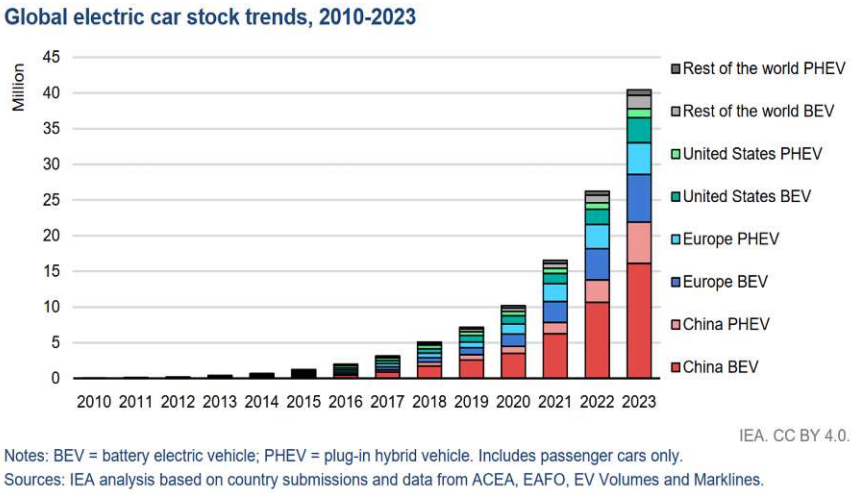


Figure 1.1: Development of EV selling numbers according to IEA [9].

Not only energy generation, but also the mobility sector undergoes a strong electrification: Electric car (EV) sales were reported to be around 14 million in 2023, 95% of which sold in China, Europe and the United States [9]. Looking at IEA's statistics (see Figure 1.1), it is clearly recognizable that the EV market is experiencing an exponential growth. The rapidly increasing number of simultaneously charging EVs poses a risk to electricity grids and charging infrastructure due to higher demand, especially during peak times and rush hours [9].

The electric vehicle driving range is constantly but moderately increasing. It still causes range anxiety among EV users and non-users. The concerns also relate to available charging infrastructure for long-distance journeys [11]. Still, according to IEA's forecast, the global EV fleet is set to grow twelve-fold by the year 2035, leading to a ten-fold battery demand for this market [9].

1.2 Motivation, problem statement and aims

Several trends in the energy and road transport sectors are presented in the previous section. Looking at them, it is clear that technology needs to catch up with the demand of the ongoing electrification. Battery technology plays a key role in all of these aspects. The design of the future's power grid and network requires extensive simulation, including modeling of transient effects and peak loads. In this context, there is a need for appropriate battery models.

These should enable the following methods:

- Realistically emulate electric vehicle charging processes.
- Simulate battery packs that are built up out of several single battery cells.
- Adequate charging algorithms for electric vehicles and their simulation for energy network planning and improved testing of charging infrastructure.

- Simulation of battery behavior with the ability to scale up to complete storage systems.
- Methods to characterize newly developed batteries and to predict their behavior under different scenarios.

All these issues are addressed in this thesis.

Some further requirements and problem statements are given in the following:
As the mentioned grid simulations can cover a large time scale with many components, and solve possibly back-coupled (differential) equations, their computational complexity can explode very quickly. To maintain reasonable memory and computational costs while ensuring high simulation accuracy, the models should be

- Scalable without big "cost penalties". This means, scaling up a bigger (storage) system out of smaller ones should not elevate necessary (computational) resources.
- Available for a large scale of time resolution
- Require as few computing operations per time step as possible
- Do not need to store big data or operate on big data per timestep.

Provided that these requirements are fulfilled, a huge amount of simulation-timesteps can be done within a few seconds. An explanation:
Floating Point Operations per Second "FLOPS" measures the number of floating point operations that can be performed in one second. It is one of the most important parameters to determine the computing power of a system. Simply put, it is the number of floating point operations performed per second. It gives an idea of how fast a computer or processor can perform mathematical operations such as addition, subtraction, multiplication, division, or assignment involving floating point numbers (i.e., decimals). Newer processors can reach are over 100 GFLOPS [12]. So if one can only use 1% of this performance (the full performance can basically never be reached), this yields about 10^9 operations per second on the respective data.

Another point is exploiting the so-called "temporal locality:" Temporal locality is the tendency of programs to reuse data elements during execution. Before looking to memory for a particular data item, the cache (very fast but small memory) is searched for a local copy [13]. When a variable or instruction is temporary local means that it is kept in the cache of the computer. As a consequence it can be easily retrieved and much less time is needed to search for it, resulting in faster performance. The only real limitation in exploiting this form of locality is cache storage size.

When considering the aforementioned points it becomes clear that by keeping the data size (i.e., using only a few variables to iterate over) and the number of operations per iteration step small, an enormous number of iteration steps can be done within a few seconds of real time. Therefore, the pathway of this work is stated in terms of the following modeling approach:

"Use numerical models that capture the physics of battery systems and describe EV behavior, but operate on small datasets with a limited number of steps"

Therefore, it is investigated which kind of equations can describe the battery and the charging process. One of the well-established approaches in the literature for this purpose are first-order ordinary differential equations in time (in which the spatial dependence of the system variables is not taken into account). For their numerical solution, time-stepping methods have proven to be efficient. When implemented in the right way, they fulfill the above-mentioned boundary condition on the computational aspect of modeling [14].

The goal is to develop a software that describes the behavior of the battery under current flow. It should be based on already existing and well proven numerical methods and physical considerations. This model should be applicable to real, commercially available batteries, such as those built into EVs. It is also a goal to be able to accurately and flexibly simulate any EV whose data is sufficiently publicly available. The model should be user friendly and leave room for future improvements in terms of more accurate numerical solver methods or model component describing equations and computational optimizations. Portability and platform-independence of the created simulation software is targeted as well. The discussed requirements regarding computational complexity are also an important boundary condition.

1.3 Methodology

First, the physical and thermodynamic principles of batteries are described. This is the subject of 2.1. The reader can learn how internal variables can capture the state of the battery and which processes enhance cell degradation at high currents.

This is followed by a section dedicated to helping the understanding of EV-EVSE interactions. For this purpose the most important international standards are evaluated in 2.2.

To address the cell degradation discussed earlier, fast charging algorithms designed to mitigate this issue are presented in the section 2.3. The focus is kept on the ones that are currently used in EVs. This selection is based on studying publicly available charging curves of electric cars.

In the section 2.4 an overview of modern EV battery technologies- and management systems (BMS) and their control variables is also given for better understanding.

Based on the above knowledge, physical models are developed and implemented in Chapter 3 to

1. Simulate the behavior of a single battery cell.
2. Based on 1., simulate any assembled battery pack whose cell is already parameterized.
3. Mimic the behavior of an electric vehicle with a certain battery pack under given charging conditions.
4. Post-process the simulation results to extract technically relevant parameters of the operating process.

In this work, the modeling of the temperature distribution and evolution of the cells is not explicitly considered due to its high complexity, geometry and design dependence. It would also require solving partial (instead of ordinary) differential equations, which can lead to huge system matrices in every time step if one is not interested in the steady-state solution. Solving (inverting) huge system matrices in every time step would violate the requirements stated in the previous section. Therefore, the thesis uses electric-circuit based models to simulate battery packs. Such approaches implicitly include the chemistry, kinetics, and thermodynamics of the batteries.

2 State of the Art

2.1 Thermodynamic and physical description of batteries

2.1.1 Working principle and types

In the following, an overview of battery types and physical working principles is given based on [15]. Batteries have the purpose of storing energy in chemical form, which can be extracted in electric form during discharging. They can be comprised of one or more electrochemical cells, which consist in general of a

- Cathode
- Anode
- Electrolyte

Anode is the electrode, that provides the electrons while discharging, meanwhile the cathode "receives" them. The electrolyte is a material between the electrodes that allows ion transport while being ideally completely insulating for electrons. To improve mechanical stability and electric safety, a separator is usually added to the system.

Extraction of the energy stored in an electrochemical cell is done by redox reactions, which can be schematically described as

$$Ox \rightleftharpoons^{n_e \cdot e^-} Red \tag{2.1}$$

Ox denotes the oxidized ("loses" electrons), Red the reduced ("gains" electrons) species. Figure 2.1 represents the schematics of battery operations during charge and discharge. The difference between the anodic and cathodic electrochemical potential determines the cell potential, i.e. the maximal theoretical voltage V_{th} that can be delivered by the cell.

$$V_{th} = V^0_{red} - V^0_{ox} \tag{2.2}$$

Note that the electrode potentials from (2.2) have to be defined with respect to some reference electrode, such as the standard hydrogen electrode.

Batteries can be distinguished by the property, whether their electrochemical reactions delivering energy are reversible or not. The former holds for the so-called *secondary* batteries, whose most common types are listed below.

Lead acid batteries They have an anode from lead, and a cathode out of lead dioxide. Typically, an aqueous sulfuric acid electrolyte is used. The overall cell reaction, that delivers a theoretical voltage of 2.0 V is

$$\mathrm{Pb_{(s)} + PbO_2 + 2H_2SO_{4(aq)} \rightleftharpoons^{2e^-} 2PbSO_{4(s)} + 2H_2O} \tag{2.3}$$

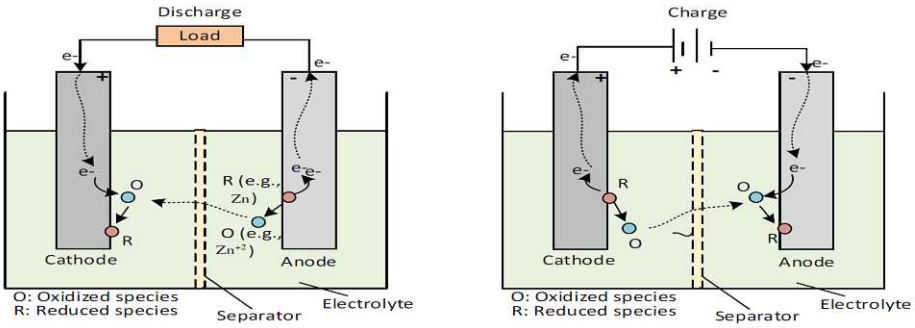


Figure 2.1: Generalized working principle of a battery with its main components. Figure taken from [15].

Lead-acid batteries have the advantage of low cost. On the other hand, they have a low specific energy (amount of energy a battery can deliver, typically in Wh normalized to its weight). Besides, the component lead is toxic. They are commonly used as starter batteries in vehicles with combustion engines. Another common application is in electrical storage technologies from small to medium scale storage applications.

Nickel-based batteries They use nickel hydroxide as cathode, and the anode material determines its sub-type. The cathode reaction can be summarized as

$$\beta\mathrm{NiOOH} + \mathrm{H_2O} \overset{e^-}{\rightleftharpoons} \beta\mathrm{Ni(OH)_2} + \mathrm{OH^-} \tag{2.4}$$

The stoichiometric coefficient β depends on the anode material, from which the following five main sub-types arise:

- Nickel cadmium battery, which uses cadmium plate as anode
- Nickel-iron battery, with Fe anode.
- Nickel-hydrogen battery, that uses H_2 as anode. Its development was inspired from both battery and fuel cell technologies.
- Nickel-metal hydride battery, with a complex metal-alloy as anode material.
- Nickel-zinc battery, using metallic Zn as anode.

All of these types have a theoretical voltage between 1.3 and 1.8 Volts under standard ambient conditions. This type is used in backup power applications, older (hybrid) electric vehicles and aircraft starting systems.

Sodium-beta batteries They use solid beta alumina ($\beta-\mathrm{Al_2O_3}$) as electrolyte. This allows the transport of sodium ions between electrodes. As anode, Na, as cathode either $S_{(l)}$ or solid metal chloride is used. They have a high operating temperature (350 °C) with a theoretical voltage between 1.78 and 2.08 V. The high energy density of this type makes it promising for automobile applications. However, until now, it could not substantially enter the market.

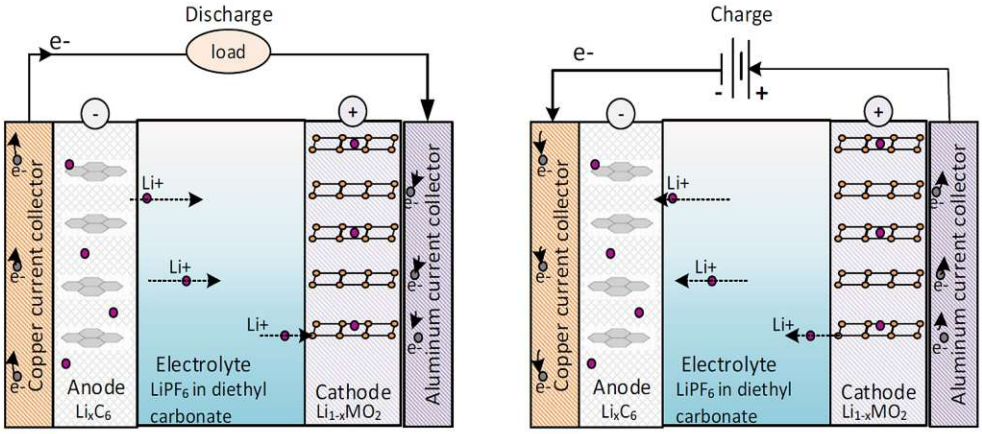


Figure 2.2: Operating principle of Li-ion batteries. Figure taken from [15].

Metal-air batteries They use (preferably atmospheric) oxygen as cathode and pure metal (Zn, Li) as anode. An aqueous solution serves as electrode. It is to be noted that it has the highest energy density and lowest energy storage costs among all existing battery technologies. However, their drawbacks such as capacity loss and enhanced degradation of its components due to exposure to the atmosphere has been prohibiting them conquering the market.

Lithium-Ion batteries (LIB) In them, Li^+ commutes between the electrodes. Therefore, they are often referred as "rocking-chair battery".

As cathode the following materials are typically used:

- Metal oxides
- $LiMPO_4$, where "M" stands for some metal, typically Fe, Mn, Co, Ni
- vanadium-oxide
- Lithium-manganese spinels

As anode, carbon, Li_4TiO_{12} and Si are most widely applied. The electrolyte is typically some organic material containing dissolved Li salt. For more details on these components refer to 2.4.1.

The working principle is shown in Figure 2.2. During discharging, Li is de-intercalated from the hexagonal C structure, while being oxidized to Li^+. The produced Li^+ ions are transferred through the electrolyte to the cathode, where they reduce some cathode components, which is traditionally cobalt. This leads to the following reactions:

$$\begin{aligned} &\text{Ox: } Li_xC_6 \Longrightarrow C_6 + x \cdot e^- + x \cdot Li^+ \\ &\text{Red: } Li_{1-x}MO_2 + x \cdot e^- + x \cdot Li^+ \Longrightarrow LiMO_2 \\ &\text{Overall: } Li_xC_6 + Li_{1-x}MO_2 \Longrightarrow C_6 + LiMO_2 \end{aligned} \tag{2.5}$$

They have low self-discharge rates of around 1.5%–2% monthly, high energy densities of 100–265 Wh/kg, and a cell voltage of around 3.6 V. Moreover, they stand out with their good cyclability. They are the market-dominating battery technology, see Figure 2.3. The most typical application areas of LIBs are:

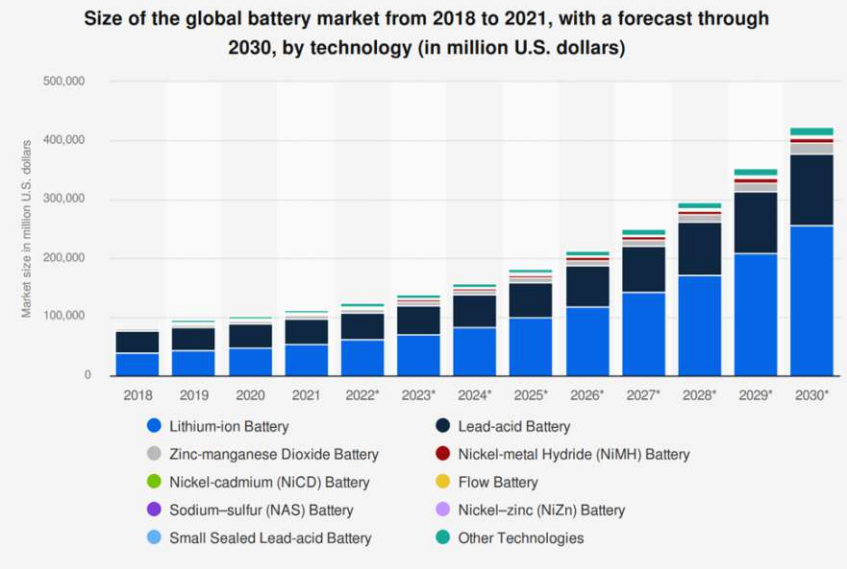


Figure 2.3: Market share of battery technologies with forecast. Figure taken from [16]

- Portable electronic devices
- Power tools
- Automotive applications such as power packs for electric vehicles
- Stationary power plants
- Renewable energy-based systems

Due to their outstanding dominance on the market and in the electric vehicle industry, this work focuses from this point on them, and with the term "battery", from now on, **Li-ion secondary batteries** are meant.

Solid-state batteries (SSB) They represent a very promising and emerging technology on the market, which has not yet been able to make the breakthrough. A summary of their characteristics is given based on [17].

The first solid-state battery was in development even before the conventional liquid-electrolyte-based LIBs. The working principle of an SSB is the same as that of an LIB, based on the movement of lithium ions from the anode to the cathode. An important major difference is that this movement is through a **solid** electrolyte. Almost all cathode and anode active materials used in LIBs can be applied to SSBs. The property of ion conduction through a solid material leads to many unique properties of SSBs:

- Unlike their liquid or polymer counterparts, inorganic solid electrolytes are relatively difficult to ignite. This leads to a higher safety.
- SSBs have the potential to prevent dendrite formation. Details on dendrite formation, it is referred to 2.1.4.
- Thermal conductivity of the solid materials is typically higher than that of the liquids. This property may help eliminate the temperature hotspots inside solid state batteries, enhancing safety and mitigating degradation effects.

- Inorganic solid electrolytes can be more thermally stable at higher temperatures than liquid ones. Therefore they are able to work more reliably under high temperatures.
- Solid electrolytes allow to use Li metal as the anode. This leads to dramatic improvements in the energy density. By [17] it is stated as the main motivation behind their intensive development.
- Due to solid electrolytes with ionic conductivity even higher than liquid ones, very thin electrodes can be used. This leads to a high specific energy of the cell.
- Excellent anodic stability has been demonstrated, enabling the utilization of cathodes with high potentials. This is also a reason behind their high energy density.
- Excellent kinetic performance mainly because of high ionic conductivity.
- The higher lithium concentration provides sufficient lithium ions for charge transfer at fast rates. As a result, very fast charge transfer kinetics has been observed with small specific resistance.

The following questions remain open in order to achieve wider commercialization:

- Understanding changes in the nature of interfaces between electrodes and electrolytes
- Optimization of the interfaces between solid electrolytes and solid electrodes
- Understanding their electronic conductivity, electrochemical stability, mechanical properties and thermal stability
- Resolving issues of processibility and scalability

SSBs are expected to be an important technology in the future, although there are still many important challenges for a commercialization.

2.1.2 Thermodynamics and kinetics

The Gibbs free energy G is a thermodynamic potential which describes the amount of maximum reversible work done by a thermodynamic system. It is given as

$$G = G(\vec{n}, p, T) = H - T \cdot S \tag{2.6}$$

H denotes the enthalpy (in Joules J), T the absolute temperature (in Kelvin K) and S the entropy (in $\mathrm{J\,K^{-1}}$) of the system. Depending on how $\vec{n}$ is defined, this can be the absolute number of particles or normalized to weight/mol/volume.

By this consideration, the maximum amount of energy, that a battery cell can deliver is given by the difference of Gibbs potentials of the educts (compounds to be oxidized) and products (compounds to be reduced). Ideally, this chemical energy ΔG can be converted completely into electric work $W = q \cdot V$ (charge $\times$ voltage). This already delivers the formula for the maximum *theoretically achievable voltage* of the battery V_{th}:

$$V_{th} = \frac{\Delta G}{n_e \cdot F} \tag{2.7}$$

In (2.7) $\vec{n}$ is given in units of mol: $q = n_e \cdot N_A \cdot e = n_e \cdot F$, where

- N_A denotes the Avogadro-constant
- e the elementary charge
- n_e the number of electrons transferred in the Redox reaction
- ΔG the difference of Gibbs potentials of the oxidized and reduced side.

The measurable voltage under stationary, no load conditions is called *open circuit voltage* V_{oc}. This is usually insignificantly lower than V_{th} [15]. Therefore, from now on, in this thesis the following approximation is taken:

$$V_{oc} \approx V_{th} \tag{2.8}$$

In a real battery system, unwanted reactions occur in addition to the theoretical redox processes. Phase change and mixing effects also enter the H term of (2.6). The reactions together result in a generated entropy S_{gen}, that also contributes to G. Besides, the temperature of the cell is in general a function of space and time $T = f(\vec{x}, t)$. Therefore, by taking (2.6) and (2.7) a part of the deliverable voltage might be "lost" due to unwanted entropy generation. It is also important to consider, that there is an over-potential on electrode, cathode, and anode due to

1. Transport kinetic limits
2. Finite conductivity of single components

These points can be described by Butler-Volmer and Ohmic relations [15]. The above considerations lead to a deviation of the measurable battery voltage V_{bat} under current flow from V_{oc} and to heat generation. Section 2.5 deals with the former, and section 2.1.4 gives details on the latter.

2.1.3 Description of battery state

What can be observed on a battery is its voltage, current and temperature, and possibly its mechanical state, such as cracks and expansion. It is of interest to define from these observations a set of parameters that allow a statement to be made about "how the battery is doing". They are called the **internal state variables** of a battery and represent a state of the art abstract concept in the literature. Although they are not physical and measurable observables, they represent the input of many actions that need to be taken on the battery and model its behavior (see 2.4.3). They are also used as independent variables to parameterize battery models. They are evaluated in this section.

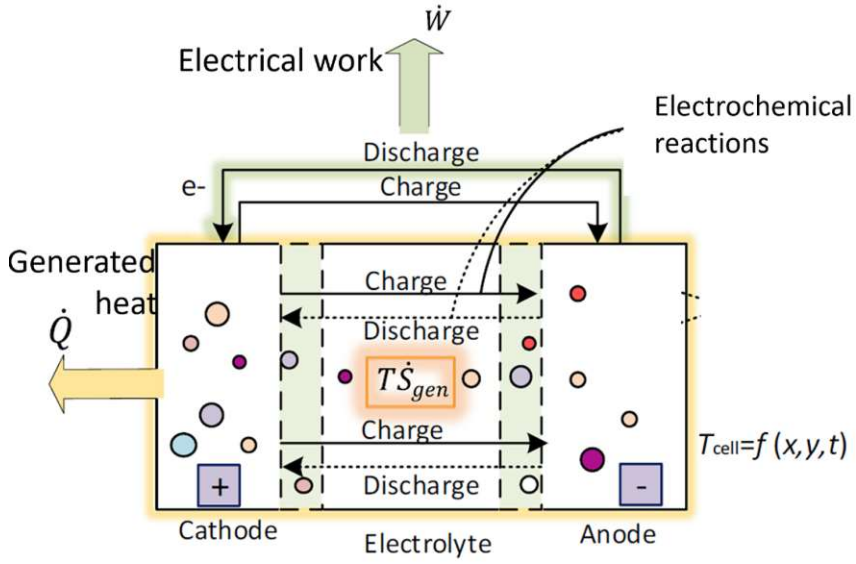


Figure 2.4: Battery system schematics. Figure is taken and modified from [15].

State of Health

Batteries irreversibly and gradually degrade with time. This process can be subdivided in two processes: cyclic and calendar aging.

The first one refers to the degradation of the battery due to charging and discharging. It depends on many parameters, such as the cycle conditions and temperature. There is no consensus in the scientific community how to exactly capture this process numerically. However, there are some protocols that define evaluation standards, such as the American "USABC" [18] and the Chinese "QC/T 743" [19].

Calendar aging should be analyzed independently of cyclic aging. Tests designed for this purpose involve studying the changes in the characteristics of a cell without charging or discharging it. Such tests take a long time, and the aging process depends on many environmental conditions, such as humidity and temperature. This adds to the complexity of the modeling process to capture calendar aging. To reduce the time and cost of testing, some researchers use the accelerated calendar aging method to develop mathematical models of calendar aging [20].

The mentioned processes result in altering of the cell parameters, which is described by the expression *State of Health* (SoH). There are varying definitions of this parameter, including considerations based on capacity, internal resistance, peak power of the battery or number of cyles [21]. The first two are the most common ones, so they are evaluated.

SoH definition by capacity According to this method, a numerical determination of the SoH is done by comparison of the aged and new capacity. This is again not well-defined, since it can be given in energy as well as in charge unit, and also depends on operating conditions, primarily on the C-rate. In this work, the "Capacity-related State of Health" SoH_C is defined as follows:

$$SoH_C = \frac{Cap}{Cap_{new}} \tag{2.9}$$

This means by the term capacity, the amount of charge in Ampere-hours Ah that can be stored in a battery is meant.

In (2.9) Cap denotes the maximum actual capacity and Cap_{new} the capacity of the new battery, both in Ah. The lower index C indicates that SoH is expressed in this capacity fade. Currently, electric vehicle batteries are considered to reach end of life when SoH_C drops to 70-80% [22].

SoH definition by internal resistance The consensus in the literature is that battery degradation results in an increase in internal resistance. However, there is no agreement on how to measure this "internal resistance". Especially since it plays a different role in different battery models. In this paper, due to the use of the so-called *Electric Equivalent Circuit Model* (see more in 2.5.2), it is defined by

$$SoH_R = \frac{R_{0,EOL} - R_0}{R_{0,EOL} - R_{0,new}} := \frac{k_a \cdot R_{0,new} - R_0}{k_a \cdot R_{0,new} - R_{0,new}} \tag{2.10}$$

Above, the internal resistance captures R_0 from Figure 2.14. Since SoH_R is by definition between 0 and 1, setting the value of the so-called "end of life resistance" R_{EOL} is necessary. It can be expressed by the internal resistance of the new battery with the help of a resistance-aging threshold factor k_a (this is a term introduced by the author) according to:

$$R_{0,EOL} = k_a \cdot R_{0,new} \tag{2.11}$$

In some literature, the value $k_a = 1.33$ is used. This factor is introduced by Guha et al. [23]. However, no proper explanation for this exact value was found.

State of Charge

This parameter is the most important one among all internal state variables, since most decision algorithms and models take it as an input parameter. It is given by

$$SoC = \frac{Q_B}{Cap} \tag{2.12}$$

The deviations of SoC's definition arise from the interpretation of nominator and denominator terms. Q_B denotes the dischargeable amount of energy or charge currently stored in the battery. Its amount is in general a function of operating conditions: There is no consensus on which ones to take and how to relate it to the current battery capacity Cap. As previously discussed, Cap is again, not well-defined, because it is a function of aging. In order to get a consequent method for its calculation, in this work

- In Equation (2.12) occurring variables have the units of Ampere-hours Ah.
- The dependence of Q_B on the operating conditions is neglected. This means, one assumes that under any operation conditions one can get out all the charge stored in the battery. This assumption can be justified with following two arguments:

1. SoC is an abstract concept that helps to relate parameters describing battery behavior, building some kind of "mapping". It is thereby only of importance that one uses a consequent calculation for its value that is used as an input for the following algorithms and models.
2. This study focuses on simulating fast charging. Hence, this method is conducted within an SoC range approximately between 0.2 and 0.8, making it not particularly relevant whether the battery is fully discharged or not.

- The stored amount of charge in the battery equals the integrated current $i(t)$ that flows into it, minus losses, which are described by the efficiency parameter η. This translates to the following equation:

$$Q_B = \int \eta i(t) dt \tag{2.13}$$

- Battery capacity Cap is governed by equation (2.9). Also $Q_B \leq Cap$ needs to be satisfied.

Further internal variables

In the literature, the internal variables listed above are also mentioned. They can be input parameters in some more advanced models. However, they are not currently widely used and will not be discussed further in this study.

- State of Power: For an EV, it can be defined by the amount of power that the battery pack can deliver to an electric load. In general, it is a function of SoC and the temperature.
- State of Energy: Depending on the narrower definition, it can either express the chemical energy currently stored in the battery, or the one that it can provide to an electric load.
- State of Function (SoF): for an EV, in can be defined as the power that can be delivered by a battery (pack) to some electric load, such as the motor. [21]

2.1.4 Degradation processes at high currents

Thermal effects

One of the main issues of fast charging is the heating of battery cells. In Figure 2.5 an overview of effects leading to battery heating in EVs are presented. In order to sufficiently cool battery packs and have a uniform temperature distribution in them, multiple approaches exist. An overview is given in [24].

The total heat generation Q_t can be divided into two main categories: reversible (Q_r) and irreversible (Q_i) heat.

The irreversible part of the heat generation rate in a cell during a charging process is due to the difference between the change in Gibbs free energy and the actual electrical work delivered by the cell.

- The change in Gibbs free energy $\Delta G = -nFV_{th}$ represents the maximum amount of chemical energy that can be converted into electrical energy
- The electrical work is $W_e = nFV_{bat}$

Here V_{bat} represents the cell voltage. The difference between the above two values corresponds to a dissipated heat rate. Such dissipation is an irreversible process.

The reversible heat rate is caused by the entropy change ΔS due to the chemical reactions inside the cell [25]. It is given as

$$\dot{Q}_r = \frac{d}{dt}(T_s \Delta S) \tag{2.14}$$

Therefore the total heat loss Q_t is given as

$$\begin{aligned} \dot{Q}_t &= \dot{Q}_i + \dot{Q}_r \\ \dot{Q}_i &= \frac{d}{dt}(\Delta G + W_e) = I(V_{th} - V_{bat}) \\ \dot{Q}_r &= \frac{d}{dt}(T_s \Delta S) \end{aligned} \tag{2.15}$$

T_s denotes the surface temperature of the cell. Assuming $V_{th} \approx V_{oc}$ that is usually a good assumption, the difference between cell and theoretical voltage is given because of the total resistance of the cell R_t [25].

$$R_t = \frac{\Delta U}{I} = \frac{V_{oc} - V_{bat}}{I} \tag{2.16}$$

Here ΔU includes both polarization losses of the electrodes and ohmic losses of both electrolyte and electrodes. The irreversible heat rate can be expressed as the product of battery current and voltage loss:

$$|\dot{Q}_i| = I(V_{oc} - V_{bat}) = I^2 R_i \tag{2.17}$$

The change of entropy due to the chemical reaction is given as

$$\Delta S = nF\frac{dV_{th}}{dT} \tag{2.18}$$

Using this relation with (2.14), the reversible heat generation is

$$Q_r = IT_s\frac{dV_{th}}{dT} \tag{2.19}$$

When obtaining (2.19) it is assumed that the surface temperature is constant and the theoretical voltage is not a function of time as well, i.e

$$\frac{dV_{th}}{dt} = \frac{dT_s}{dt} = 0 \tag{2.20}$$

The temperature evolution of the cell can be described via the energy balance. When neglecting expansion work, the change in the inner energy U of the cell at a constant pressure equals the amount of heat flowing into / coming out from the cell:

$$dU = m \cdot c_p \cdot dT = dQ \tag{2.21}$$

with

$$dQ = d(Q_t - Q_d) \tag{2.22}$$

Q_d is the heat delivered to the surrounding through the boundaries of the cells.

Describing the relation 2.21 in dependence of the time by inserting (2.15), (2.17) and (2.19) into that, one obtains the evolution of cell temperature in dependence of the charging current I

$$mc_p(T)\frac{dT}{dt} = (\dot{Q}_t - \dot{Q}_d) = I^2 R_t(T) + IT_s\frac{dV_{th}}{dT} - \dot{Q}_d \tag{2.23}$$

m and c_p are the mass and the specific heat capacity at constant pressure of the cell, respectively. The latter is in general a function of the temperature itself, so is the total cell resistance R_t.

The amount of heat that the cell "looses" Q_d is due to

- Conduction: linear dependence on the temperature
- Radiation: T^4 dependence, Boltzmann-law

Looking at (2.23) it can be seen that the temperature evolution is a non-linear equation. What is particularly important in the context of this work is the current dependence. At high temperatures and low currents, reversible heat generation dominates Q_t and radiation dominates Q_d. However, this is not the practical case for fast charging, where one needs high charging currents and wants to keep the temperature low. Therefore, the focus is on the I^2 term in Q_t and the linear (convective) term in Q_d.

Current densities are not equal inside the battery, which leads to different heat generation rates at different locations. Since the conductivity of the cells is finite (and usually inhomogeneous) this results in a non-uniform temperature distribution inside the cell. The local inhomogenities of temperature and current distribution can again lead to different reaction rates and therefore different local degradation rates [26].

Further overview on temperature behavior of the battery cells can be read in [27] and [28].

Temperature induced degradation

Many degradation mechanisms in Li-ion batteries show a temperature dependence, that can be represented by Arrhenius plots [30], or by empirical equations [31].

A key parameter for degradation effects is the growth of the so-called "solid electrolyte interface" (SEI). It is a passivization layer formed already in the very first cycle on the surface of lithium-ion battery anode, produced by electrolyte decomposition. It allows

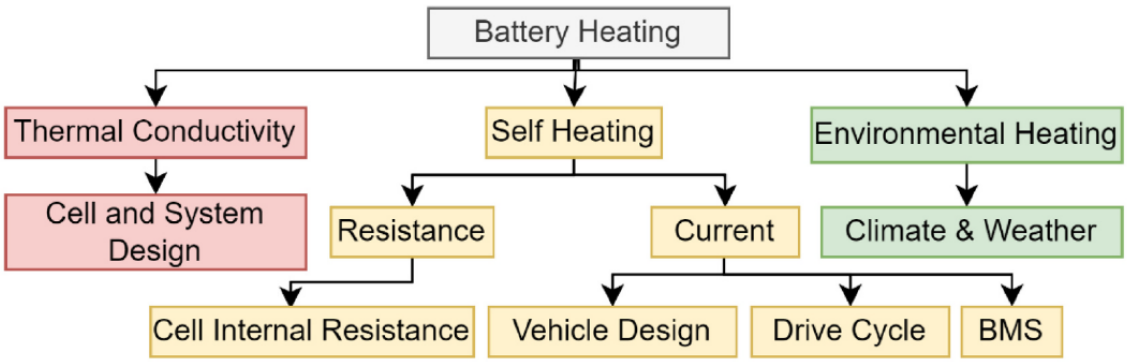


Figure 2.5: Overview of processes leading to battery cell heating. Graphic taken from [29]

Li^+ transport while preventing further electrolyte decomposition. The quality of the SEI plays a critical role in the cyclability, rate capacity, irreversible capacity loss and safety of lithium-ion batteries: a stable SEI layer can ensure good long term charge capacity retention stability [32]. Meanwhile, an unstable solid electrolyte interface will consume lithium ions thereby decreasing charge capacity while increasing electrical cell resistance. Unstable SEI layer formation is one of the primary reasons for poor capacity retention [33]. In the following it is to be evaluated, how temperature affects this parameter.

At high temperatures, the SEI thickness on the anode increases, causing irreversible capacity fade. SEI growth is considered the main degradation mechanism in Li cells because it consumes active lithium and is an irreversible process. The grown SEI film also increases the cell impedance. During its formation, gaseous by-products may be released, causing additional mechanical stress and thus degradation of the cell [26]. Above 25°C, in addition to the increased growth of the solid electrolyte interface, higher cathode degradation and an increase in cell resistance can be observed.

At lower temperatures, Li -plating on the anodes and subsequent reaction with the electrolyte is a dominant degradation mechanism. Li-Plating is a side reaction in which Li^+ ions form Li-metal on the anode instead of intercalating into it. This leads to loss of active material (cyclable Lithium). In this regime, the diffusion and intercalation of Li is slowed down leading to lower charging efficiency [30].

When at lower temperatures the lithium ion diffusion slow down, it can lead not only to lithium plating, but also to dendrite growth during fast charging. The dendrite growth means that lithium dendrites are formed when the Li^+ ions accumulate on the anode surface and cannot be sufficiently absorbed into the anode because of too slow kinetics. They can cause short circuits and leading to safety issues. Another consequence of this process capacity fade: The lithium dendrite reacts with the electrolyte, causing it to decompose and triggering the loss of active lithium inside the battery [34].

Figure 2.6 presents the qualitative behavior of the aging (degradation) rate dependent on the temperature based on the Arrhenius-relation.

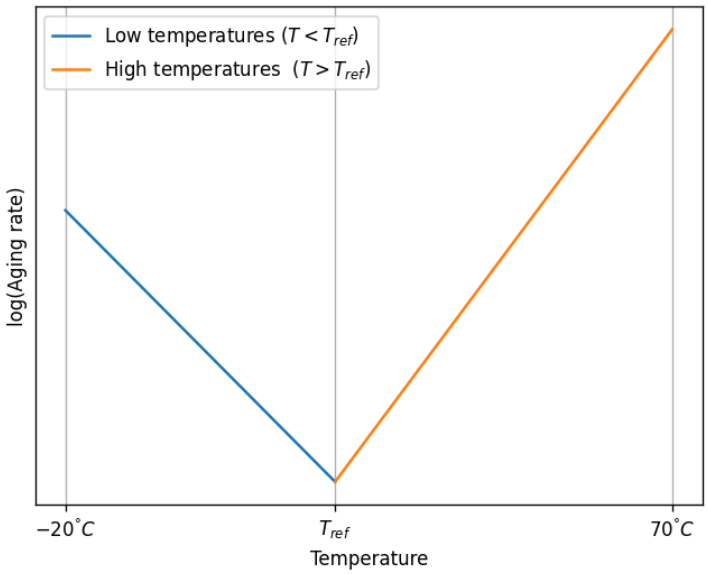


Figure 2.6: Qualitative dependence of battery degradation rate on the temperature. As T_{ref}, one usually takes 25 °C

Mechanical effects

Stress occurs during cycling due to Li (de-)intercalation: This process can lead to an increase/decrease of the electrode volume as ions are regularly inserted/extracted. Since the concentration of intercalated Li is also spatially non-uniform, the resulting gradient leads to different volume expansions and phase transitions. The intensity of the strain is strongly dependent on the current density, since at higher C-rates the concentration gradients of Li are higher, leading to higher strain and thus stress [35]. In addition, gas evolution and associated volume expansion due to side reactions lead to additional mechanical stresses. As discussed above, these side reactions are even more intense at higher current densities, because these lead to higher temperatures, which in turn facilitate side reactions (the reader may recall the Arrhenius equation for the temperature dependence of reaction rates).

The strain is not only dependent on the amount of charging current, but also varies at different SoCs [36]. The fast change in SoC during fast charging implies thereby higher variations in strain, which in turn, poses even more demand on the cell constituents.

Mechanical stress induces cracks, that result in higher reactivity with the electrolyte material. This lowers electrical conductivity [29].

One can classify the mechanically induced losses of the battery pack as following [29]:

- Loss of active material
- Impedance growth
- Loss of Lithium inventory

Effects on charging efficiency

Manufacturers publish the so-called "rated efficiency" or "efficiency" of a battery. It is given as the ratio of the energy retrieved from the battery, to the one provided to the battery, when coming back to the same State of Charge [37].

This quality, however does not take the specific charging and discharging characteristics into account [38]. As the (dis-)charge current increases, the (dis)charge efficiency decreases. This mainly due to potential losses that result in a lower output-voltage than the ideal open-circuit voltage. They will be evaluated above based on the work of Rosen [15].

1. One contribution to the overpotential caused by the resistance to the flow of current. This is called as the "ohmic" overpotential. Its main source is often the ionic resistance of the electrolyte.

2. The difference between the rate of forward and backward reactions corresponds to the net current density resulting from the anodic and cathodic reactions. For both, there is an energy barrier that must be crossed for the reaction to start and continue. This can be described using transition state theory. The Butler-Volmer equation describes the relation between the current density and overpotential related to the anodic and cathodic half-reactions. It is the result of reaction kinetics, and increases with the current density. The voltage drop due to these effects is also called as "surface overpotential".

3. When a battery is being charged or discharged, there is a transfer of mass, momentum, heat and work. These mechanisms result in the so-called "concentration overpotential": Transfer mechanisms lead to concentration gradients, especially when large amounts of current are drawn from the battery. This concentration difference, which is associated with limitations in the mass transfer of reactant or product species, can cause an electrical potential loss.

For discharge, the battery capacity as a function of discharge current is characterised by Peukert's equation [39]. This law, however, only considers discharging efficiency. Jeon et al. have proposed a method to measure the charging efficiency η_C of electric vehicles, which is governed by a polynomial of degree 3 depending on the charging power. In other words, it depends mainly on the charging current supplied. When charging at high power (current), η_C decreases [40].

2.2 International standards on EV charging

In order to establish a framework that allows to charge EVs produced by different manufacturers, international norms have been developed in the past. The charging equipment that is used to recharge an EV is called *electric vehicle supply equipment (EVSE)*. It needs some external supply itself to be able to provide the necessary power to the electric car. EVSEs can be classified into categories based on the following criteria according to the standard IEC 61851-1 [41]:

1. **Power supply input**: based on the grid supply system, it can be DC or AC.
2. **Power supply output**: Based on the type of current, that is delivered to the EV it can be either DC (direct current), AC (alternating current) or AC and DC. In case of "AC" the conversion to direct current (that actually arrives to the battery) takes place in the EV.
3. **Type of electric connection method**: it can be either plug and cable connected or permanently connected
4. **Environmental conditions**: Indoor or outdoor
5. **Access**: Locations with restricted or non-restricted access.
6. **Mounting method**: Stationary or non-stationary EVSE.
7. **Protection against electric shock**: Class 1 or class 2.
8. **Charging modes**: Mode 1-4 (see below)

There are also different types of connection sockets, both for AC and DC chargers. For AC, there are three main types [42]: Type 1 ("Yasaki"), Type 2 ("Mennekes") and Type 3 ("Grand") sockets.

For DC-charging, the most important standards are:

CHAdeMO. This is the Japanese electric vehicle charging standard.

NACS (North American Charging Standard) . The outdated name is "Tesla Super Charger". It is about to overtake the current CCS1 standard in North America. According to Tesla *"Superchargers make up three out of four fast chargers and have set the standard for EV charging in North America. This standard, known as the North American Charging Standard (NACS), has been adopted by all large automakers, with manufacturers and charge point operators transitioning to the NACS by 2025."* [43].

GB/T. Chinese electric vehicle charging standard.

Combined Charging System (CCS). This is the plug type used in Europe. The following will therefore concentrate on this standard. CCS also enables AC charging with Type 1 (North America) or Type 2 (Europe).

The standardization of electric mobility infrastructure requires specifications divided into the following sub-categories [42], [44]

Plugs, in- and outlets: They are subject of the international standard IEC 62196. They specify general requirements, dimensional compatibility for both AC and DC contacts, vehicle connectors and inlets as well as thermal management system requirements for DC chargers.

Safety: the IEC 61140 defines requirements on installation and electrical assemblies. Other standards dealing with these aspects are IEC 62040, IEC 60529, IEC 61851, IEC 60364-7-722, ISO 6469-3 and ISO 17409.

Security: Electric vehicles are intended to regularly connect to public infrastructure and exchange information. Hence, there are protocols to ensure secure digital communication exchange. These aspects are subject of ISO/IEC 27000, IEC 62351, IEC 61859 and ISO/IEC 15118.

Charging modes and topologies: They are specified by the IEC 61851-1. This standard classifies charging modes into 4 categories that are summarized in Table 2.1

- **Mode 1** establishes the connection for charging directly between the EV and the AC socket outlet. There is no controling pilot pin implemented in this mode, solely a protective earth conductor. Due to safety issues, this mode is prohibited in several countries.
- **Mode 2** uses a plug connection between AC socket and EV with an additional control pilot function inbetween. This mode covers the use of an in-cable-control device (see Figure 2.7).
- **Mode 3** establishes a connection between the EV and the AC-EVSE using a control pilot function. This enables in contrast to Mode 1 controlled charging. The EVSE itself is connected to a supply system (such as the supply grid).
- **Mode 4** uses a connection between the EV and a DC-EVSE with a control pilot function.

The schematics of a CCS2 connector, with its components

- Type 1 or 2 connector with CP and PP for communication, L1, L2, L3 for AC supply, protective earth conductor "PE" and neutral conductor "N"
- DC+ and DC- plugs for direct current supply

is shown in Figure 2.8. More on abbreviations "PP" and "CP" can be found below, under "Communication."

	Mode 1	Mode 2	Mode 3	Mode 4
Voltage [V]	480 (for 3-phase)	480 (for 3-phase)	480	DC 200-600
Current [A]	16	32	up to 250	up to 400
Power [kW]	13.3	26.6	up to 185	up to 250
Connector	Household	Household	Type 1-3	CCS/CHAdeMO/...

Table 2.1: Characteristics of the modes defined in IEC 61851. Table is based on [42].

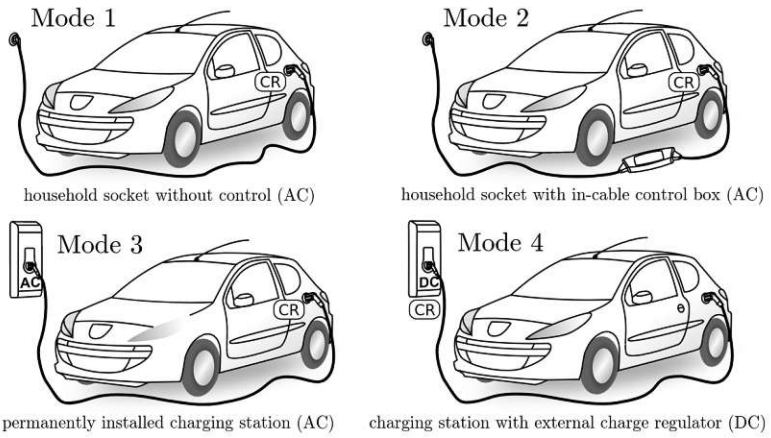


Figure 2.7: Charging modes according to IEC 61851. Graphics is taken from [45]. "CR" stands for "charging regulator".

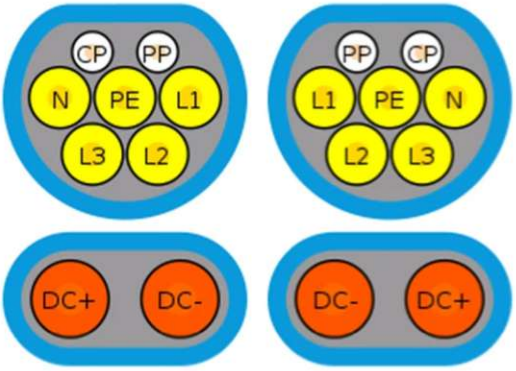
Communication The IEC 61851-1 establishes an analogue, low-level communication protocol based on pulse width modulation to control the charging process. It also describes the hardware requirements between EV and EVSE. Charging begins with an information exchange between the car and the charging station: By the Proximiy Pilot "PP" (implemented in Type 1, Type 2 as well as in CCS plugs) the maximum rated current of the connected cable is communicated towards the car. Via the Control Pilot "CP" pin, the EVSE sends the maximum charging current it can provide. The EV can still draw another, lower current while charging [44].

With the purpose to define a secure vehicle-to-grid communication interface, the ISO/IEC 15118 was established. It extends IEC 61851 with an extensive digital communication protocol for the transmission of energy-relevant information. The following points are defined in the scope of this standard [46]:

- Use cases, schematics of communication process, requirements both on EV and EVSE
- Technical protocols, physical and connection requirements
- Conformance tests
- Wireless communication
- Since the release of ISO 15118-20 a protocol (year 2019) on bidirectional charging is defined. This version of ISO 15118 describes the 2nd generation requirements for the network and application protocol.

A prerequisite for the usage ISO/IEC 15118 is the presence of a CP pin according to IEC 61851. It aims to create a "plug-and-charge" environment, intelligent integration of the EVs into the energy grid, by exchanging information about the car's State of Charge, schedules and electricity contract information [46].

The scope of ISO 15118 is limited to EV - EVSE communication. The Open Charge Point Protocol (OCPP) addresses this issue. It is an international open standard, designed to be vendor-independent, while allowing charging stations and central systems to



CP	Control Pilot
PP	Proximity Pilot
PE	Earth
N	Neutral
L1, L2, L3	3-phase AC supply
DC+	DC supply
DC-	DC return

Figure 2.8: CCS2 connector with Type 2 AC supply. Left: EVSE side connector; Right: EV side connector. Taken from [50]

communicate with each other, independent of vendor. [47].

The OCPP protocol enables EV charging stations and central backend systems to exchange information in form of messages and data with each other. The standard defines the role of [48]

1. "Charge Point" (client): This is the charging station which can have one or multiple connectors. It provides information about the availability, status of charging sessions and energy consumption.

2. "Central System" (server): By this term, the backend charge point management system is meant. It communicates with the charge point via OCPP to monitor and manage charging sessions and collect data on energy consumption and billing.

The most recent version is OCPP 2.0.1, introduced in 2019. It extends the earlier version (1.6) by many functionalities, incorporating 65 message and 129 data types [48]. The new version, OCPP 2.1 is currently being worked on. This will make bidirectional charging in accordance with ISO 15118-20 fully compatible with Charge Point Operator (CPO) backend systems [49].

2.3 Charging algorithms

A key aspect of batteries is how charging can be done. This process needs to fulfill multiple (sometimes concurrent) requirements:

1. Sufficiently short charging time: In order to motivate consuments to buy electric vehicles, it is a key aspect that they do not need to wait for long hours during trips to charge the car battery. This is the key idea behind fast charging. This however opens the door to new, earlier not relevant or known problems in real applications.

2. Safety: It is well known that Li-Ion batteries can be prone to thermal runaways and inflammations. High current densities are necessary to obtain a State of Charge around 1 in shorter time, since: $Q(t) = \int_{t_0}^{t} i(\tau)\eta(i(\tau))$. This means that the amount of current charged at time t, which determines SoC at that moment, is equal to the integrated current i between t and t_0, if one starts charging at t_0. The charge efficiency η also plays a role. This is generally a function of the charge current itself, and its definition is not entirely clear.

   What is also known is the quadratic relationship between generated joule heat q and current on some resistance: $q \propto i^2 \cdot R$. This places high demands on the battery thermal management system (such as cooling, monitoring).

3. Spare the battery: High current densities do not only pose safety issues. They also lead to enhanced degradation of the battery, see 2.1.4.

To charge a battery fast, high C-rates are required. However this has a major drawback: enhanced cell degradation by impacting the SoH and high capacity loss. The degradation effect is mainly due to thermal and mechanical effects [29]. They are discussed in 2.1.4.

Based on the previously discussed aspects, there is a need for new fast-charging algorithms that can mitigate the degradation effects of batteries. Meanwhile, they are supposed to reach a high SoC within a short time. It was verified by numerous studies, that simply applying higher currents in the conventional CC-CV algorithm does not satisfy these requirements. For this reason, many approaches have been proposed in the literature. Some of them are evaluated in the following, then compared with observable charging behavior of commercial EVs. This allows making assumptions which one might be already in use, and therefore should incorporated into the simulation framework of this work.

2.3.1 Conventional CC-CV charging

First, the traditional approach to charge batteries and electric vehicles is to be evaluated. It is called the **Constant-Current, Constant-Voltage (CC-CV)** algorithm. Its principle is the following:

1. After an initialization phase, the charging station delivers a constant current that has to be set during the communication process [41].

2. During the CC-phase the SoC of the battery increases, so does the open-circuit, and the measurable terminal voltage of the battery.

3. After a pre-set, battery-specific terminal voltage is reached, the charging current reduces exponentially to ensure that the terminal voltage stays in a specific boundary (ideally, constant, therefore the *constant voltage* naming). In this phase, also the charging power decreases exponentially. This CV mode is necessary for the battery's health, however it extremely elongates the charging time due to the exponential decrease of charging current. There is a trade - off between battery health and charging time / current. It has been found that higher C-rates in the CC arise a necessity for a longer CV - phase, making the net charging time not necessarily shorter than for lower C- rates [51].

The advantage of this method is its simplicity. However, its drawbacks do not make it suitable for fast charging: It results in higher charging losses and does not lead to a sufficiently low charging time even though applying high currents in the CC-phase. The high CC-phase currents result in addition in increased battery-degradation. .

2.3.2 Novel approaches

Many approaches have been developed and published claiming to be the "ideal" charging strategy. However, there is no consensus in the scientific community which one is ideal. It is in general a trade-off between the simplicity of implementing an algorithm and its accuracy/success. Many of the algorithms, even though being promising, could not find their way into real-world applications due their high complexity or demands towards sensoric or computational efforts.

One can cluster the optimization approaches into the following main categories: Heuristic studies and model supported studies on external or internal cell states.

The heuristic models are the most popular approach among the fast charging strategies due to their simplicity. Their principle is the following: Many cells are cycled with an individual, pre-determined cell cycling strategy that is based on some assumptions/motivation to optimize charge time to be as low as possible and/or keep the battery‘s health as good as possible. Then, cells are cycled until a pre-set number of cycles or capacity is achieved. The most optimal strategy "wins" and is proposed for application. The following methods are the ones mostly mentioned in the literature as part of this this category:
Boost Charging (BC) *Multi-stage Constant Current* (MCC), *Pulse Charging* (PC), *Constant Power* (CP) and *Negative Pulse Current*(NPC) charging [29], [51], [52].
Usually, after reaching a certain SoC, the aforementioned strategies switch to a CV-Phase. A short description of them follows.

In Figure 2.9 the most common strategies' voltage, current and State of Charge profiles are represented.

Boost Charging protocols. They apply high average current at the beginning of the charging process. This is followed by a CC-CV at lower currents. They are based on the following idea: Applying a very high, "boost" current at the beginning of charging is claimed to mitigate the possibility of overcharging. Applying high current and power densities in the low SoC-regime (and therefore reduced voltage) are less destructive for

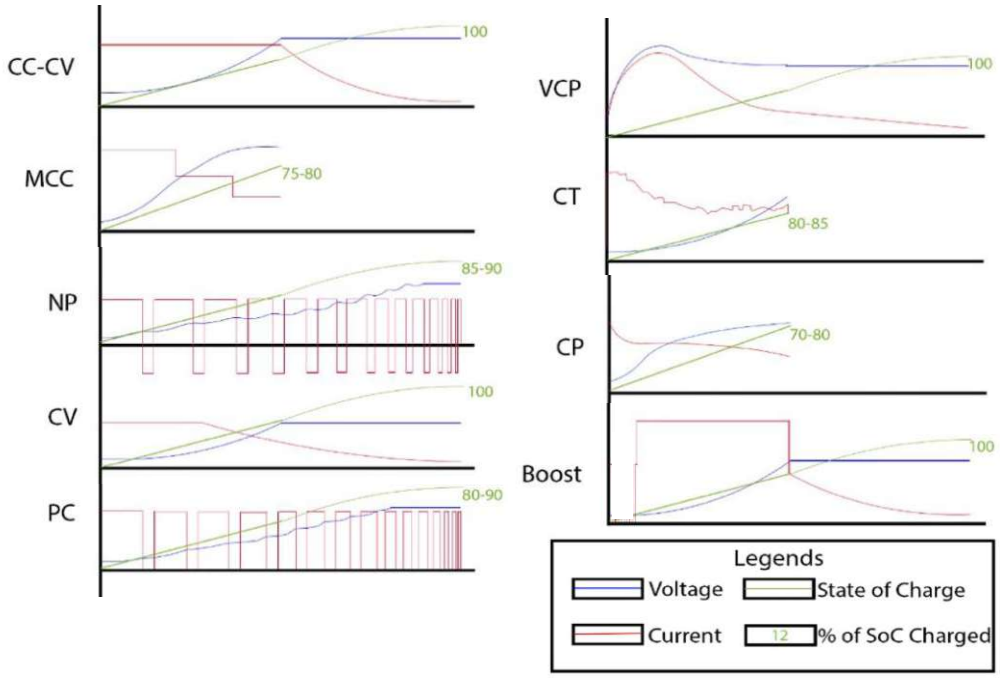


Figure 2.9: An illustrative representation of current an voltage profiles of different charging algorithms. Graphics is taken and modified from [29]

the cell. The initial boost charge stage typically consists of a high current CC or a CV in which the cell is immediately brought to the maximum voltage V_{boost} or a combination of the two in a CC-CV. Some researchers claim, that these protocols are only able to reduce charging time in exchange for higher capacity fade.

MCC protocols. The idea is to adjust the constant current levels during charging to mitigate cell degradation. The higher currents are usually applied at the early stages, but some authors proposed the opposite approach [53]. For transitioning from one CC-stage to the other, the "cut-off voltage-based" or SoC-based criteria are the most frequently used ones in literature. The current is reduced according to some formula as soon as the terminal voltage of the battery reaches the cut-off voltage during charging. In case of State of Charge based transition criterion jumping from one to another current level happens at certain SoC breakpoints. Note that even though using SoC-based criterion, in a real battery management system this is still based on battery terminal voltage management: The internal, abstract and directly non-measurable State of Charge is calculated according to some algorithms, based on observables such as measurable voltage, see 2.4.3. Some researchers also propose transition criterion based on a threshold-temperature [54].

CC charging continues (in the next stage, at another current level) until the cut-off voltage or next State of Charge breakpoint is reached again. This is repeated until the predefined number of stages is reached. Since the terminal voltage is a well-observable and measurable parameter, and the transition criterion is well-defined, it is an easily implementable algorithm. The same statement applies to the State of Charge based transition: in electric vehicles BMS, SoC calculation takes place anyway (charging state of the car is an indispensable information for users). For this reason it does not require great additional efforts to implement transition criteria based on reaching certain State

of Charge values. The MCC pattern was therefore not surprisingly observed in many charging curves of EVs, see following sub-section.

Khan et al. [55] developed a formula for the current levels based on an equivalent circuit model of a Li-ion battery to model its thermal characteristics. With this as boundary condition, the algorithm optimizes for the total charge time. In this paper, they used $N = 5$ stages, but other authors used the same analytical model with different, odd number of stages. The the nth-stage current I_n is given as arithmetic mean of the previous and next current level:

$$I_n = \sqrt{I_{n-1} \cdot I_{n+1}} \tag{2.24}$$

In the first and last stage, currents need to be predetermined.

The benefits of the MCC charging strategy have been studied by many researchers, with a varying number of stages and distinct transition conditions. The battery performance is determined by the following four parameters: charging time- and efficiency, discharge capacity and temperature rise during charging. In the literature, there is often a concentration solely on one parameter.

Thermal Management Protocols. The battery temperature is considered as a key degradation metric. Therefore, fast-charging methods based on the idea to sustain constant temperature (CT) were presented by multiple researchers [56], [57]. The problem with this method is that it requires complex equipment and charging algorithms to monitor and maintain temperature. This makes it infeasible to implement on the commercial level [29].

Constant Power (CP) Protocols The idea of this strategy is to keep the charging power constant. This allows high currents at lower SoC because then the battery voltage is in general lower for small SoC. it automatically results in charging current decrease at higher State of Charge regimes due to a higher battery voltage in this case. The researchers in [58] claim that this method results in a better thermal behavior as compared to CC-CV. Due to the low energy efficiency of this method, it is less adaptable for fast charging strategies [29].

Pulsed Charging (PC) Protocols. In this method, the battery is charged using current pulses spanned for a short time interval. The peak value of pulse current is much higher than that of CC, but with the same mean rate [29]. The evaluation of this protocol among researchers is ambiguous. Some claim that it can result in significantly improved battery charge and energy efficiencies, longer battery cycle life, and decreased charge time [59]. In the other hand, some scientist found that the method has bad cycling performance due to the pulverization of electrode particles arising from the larger strain during cycling [60]. "Trickle Charging" is a sub-category of the CP method, being claimed to help extending the battery's life. However, a high charging time makes it unsuitable for fast charging strategies [29].

Negative Pulse (NP) Protocols . A short negative pulse is applied after each positive current pulse. The idea is that theoretically no lithium plating should take place when the anode potential is $\geq 0\,\mathrm{V}$. Ideally, discharge (negative) pulse currents would help to recover

ions from plated lithium [61]. Therefore, charging protocols involving short negative pulses have been proposed [61]. It is claimed that these protocols allow minimization of degradation by lithium stripping effects without significantly increasing the charge time [61]. Others claim that the NP strategy shows higher degradation and slower charging, making it unsuitable [29].

Other Protocols. There are some other, more complicated charging strategies such as "Pulse Amplitude-" or "Pulse Width Modulation" [62], [63], "Continuous Direct Current" [64], "Varying Current Strategy" (VCP) [65] and optimization methods in terms of loss and charging time [66]. The problem with such protocols is their complexity and high cost of implementation. These issues make them unsuitable for mass production [29].

2.3.3 Comparison of literature methods with observed charging curves

In this section, curves are represented, that have been observed on publicly available websites. Based on these observations, a subset of the represented novel charging approaches will be implemented in the simulation, see Chapter 3. It can be also observed by looking at the publicly available charging curves that car vendors stick to one pattern in their products in general. This observation is based on charging curves provided by the website Fastned [67].

CC-CV method. The pattern of this mode was observed for many cars, mainly those with a smaller maximum charging power.

MCC, MCC-CV method. The pattern of this mode was observed for many cars, probably due to its simplicity and the fact that it is based on the conventional CC-CV mode. Some of the cars that were observed to charge using this mode, according to publicly available data on the website Fastned [67]:

- BMW ix models
- Hyundai IONIQ models
- Mercedes-Benz EQV
- Porsche Taycan models
- Toyota ProAce models
- Opel models
- Fiat models

Boost charge method. The pattern of having a high charging power at or shortly after the beginning of the charging process was also observed in case of multiple cars [67]. They were identified to use the Boost algorithm. A collection of vehicles having this property [67]:

- Tesla models

- Skoda Enyaq models
- AUDI etron GT
- Mercedes-Benz EQV
- Maxus e-deliver models
- JAC iev7s (followed by an MCC-CV phase)

Not clear. Some cars did not show up the very typical pattern of Boost/MCC, or some combinations of them, However, they mostly still resemble them. A collection of vehicles having this property [67]:

- Lexus UX-300e
- Nissan e-NV200 and Leaf
- Seres 3
- Volkswagen ID models

2.4 Battery technology in modern EVs

2.4.1 Typical cell chemistries

In the electric vehicle industry, the focus is on the utilization of Li-Ion batteries. There is hardly any non-hybrid battery electric vehicle using a different technology [68]. In the following section, state of the art technologies for anodes, cathodes and electrolytes for Li-Ion batteries used in EVs are presented. Since there is an emerging demand on high performance and low cost, a short outlook on potential future technologies is given as well. One of the main driving motives behind them is, that around 50-70% of battery costs arise from the materials, with Cobalt being especially relevant [69].

Before discussing the properties of single technologies, some terminology is presented for better understanding of the text below:

- **Battery chemistry**: When using this term, one refers to the cathode material of the battery.
- **Nominal voltage** U_N: It is the average voltage of the cell (measured between anode and the cathode) when fully charged. The actual voltage varies around this value. When referring to the "nominal voltage" of a specific battery chemistry x, it typically indicates the average voltage of the cathode material x measured against the anode, which is typically graphene.
- **Specific capacity** Cap_s: This is the amount of charge that can be stored in the battery divided by its weight (when referring to the *gravimetric* value) or by its volume (when referring to the *volumetric* value). In electric vehicle battery technology, it is common to refer to the gravimetric value when specifying certain properties. So Cap_s is given in Ah/kg .
- **Specific energy density** E_s: It is the amount of average *energy* the battery cell can store divided by its weight. It is calculated by $E_s = Cap_s \cdot U_N$.
- **Specific power density** P_s: Altough it is not a discussed property below, for the sake of completeness it should be also defined. It measures, how fast a battery can deliver power, in proportion to its weight.

Anode

The anode is the negative electrode in a battery. Graphene serves as the anode material in most commercially available EV batteries. In its hexagonal structure, Li ions can be (de-)intercalated during charging/discharging. The graphene structure has the great advantage of being very cheap and environmentally friendly. However, these anodes have the problem of low capacity and safety concerns. Although it is the state of the art in commercial EV batteries today, it is believed that breakthroughs in lithium-ion battery technology would require innovative anode components such as alloy materials, conversion-type transition metals, silicon-based and carbon-based compounds. For more details on these listed next generation technologies, see [70].

As an exception, the so-called "LTO" batteries use the compound $Li_4Ti_5O_{12}$ instead of graphite [71]. As an anode, the nominal voltage is 1.5 V with a Li cathode (standard

lithium potential). With commercial cathodes such as LMO, NMC, NCA it is between 2.1 and 2.5 V with a specific capacity of about 175 mAh/g. Although the specific energy of LTO battery is low, it is very safe and has excellent performance when cycling [71].

Cathode

The most typically used anode materials for Li-ion EV batteries are lithium iron phosphate (LFP), lithium nickel manganese cobalt oxide (NMC), lithium nickel cobalt aluminum oxide (NCA) and spinal lithium manganese oxide (LMO).

The LFP technology stands out for its enhanced safety, electrochemical and thermal stability and low costs. Its major drawback is its low energy density and conductivity: 190 Wh/kg and approximately 1×10^{-9} S, respectively. Despite cheap materials (iron and fluorine) this leads to a high cost per energy unit. To mitigate the problem of poor conductivity, carbon coating or metal doping is being used [72].

LMO was designed with the main purpose of replacing Cobalt and thereby mitigate costs. It has a three dimensional spinel structure which improves the diffusion of lithium ions compared to the other mentioned technologies. Similar to LFP, this technology is also relatively cheap with a low specific capacity of around $150 - 160$ mAh/g. A severe concern is that it also suffers from fast capacity fade when cycling. As an advantage one can mention its low resistance due to the spinel structure, which enables an excellent performance at high current rate in applications [71].

In NCA cathodes, layered structure of $LiNi_{1-x-y}Co_xAl_yO_2$ is used. The idea is to replace Cobalt by Nickel in the composition, thereby lowering the costs and increasing energy density [69]. Using more Ni however, results in worse thermal stability and cycling performance. These type of architecture is more expensive than LFP and LMO, but has a higher voltage and about 30% more capacity. Also, it is more safe in case of overcharging than the other of the mentioned technologies. NCA has promising trends in improving energy density [69], also having good capability at high current rate [71].

NMC is a layered $LiNi_{1-x-y}Mn_xCo_yO_2$ structure. It has a low internal resistance (due to the use of Manganese) and high capacity (because of Nickel). Its costs are lowered since the amount of Cobalt is reduced in the composition by using Nickel and Manganese. The main design parameter for this type of cathode are the stoichiometric factors x, y. As a disadvantage, the low specific energy has to be pointed out.

The qualities of the mentioned cathode materials are summarized in Figure 2.10. Additionally, a comparison of LTO with standard graphene anode technology is listed.

Electrolyte

Electrolyte materials should have the following properties: safety, electrochemical and thermal stability, withstanding dendrite formation and electrolyte decomposition and good ionic, but at the same time very poor electric conductivity.

Electrolytes can be subdivided in the following categories: liquid, polymer, and solid.

Cathode Type	Energy density	Life span	Cost	Safety level	Nominal voltage
LMO	Medium	Low	Low	Moderate	3.7–3.75 V
NMC	High	High	High	Moderate	3.6–3.7 V
NCA	High	Medium	High	Low	3.65 V
LFP	Low	High	Low	High	3.2 V
LTO	Very low	Very high	Very high	Very high	2.2–2.4 V

Figure 2.10: Comparison of commercial cathode materials to each other. In the last row the LTO cathode technology is compared to the traditional graphene cathode. The properties and values are taken from [71].

Liquid electrolytes are the most studied and used ones in lithium ion batteries. They consist typically of Lithium salt (mostly $LiFP_6$) dissolved in some organic solvent mixture. Their major drawback is the limited thermal stability [73]. Polymer electrolytes are based on the capability of some polymers being ion-conductors. A brief theoretical description of this process can be found in [73]. Solid electrolytes are gaining attention because they do not require (flammable) solvents, thereby providing higher safety. There is a big focus nowadays on their development. For more details, it is referred to the literature [69], [73] and 2.1.

2.4.2 Battery packaging

Since a single cell would not be able to deliver the power needed to move an electric vehicle, multiple cells are organized in strings and modules, forming a *battery pack*. The principle is the following: having n_s voltage sources, each delivering U Volts in series, their total voltage will be ideally $n_s \cdot U$. Having n_p current sources in parallel (each of them delivering I Amperes), they yield together $n_p \cdot I$ Amperes.

When packaging, m single cells are first associated into smaller units, the so-called modules. These M modules then are ordered in the package, that has $M \cdot m$ cells. How this is exactly done, is a unique feature of the respective car. However, at certain cases, there is some public data on the number of battery modules M in the battery pack, number of cells in the module m, eventually also including cell configuration (n_s serial, n_p parallel). A website that provides such data on almost all commercially available electric vehicles is the *"Electric Vehicle Database" (EVDB)* [74]. This website was extensively used by the author to study patterns and trends for electric vehicles.

As an example, the Renault Zoe ZE50 R135 can be taken [75]:

- EV-Battery architecture voltage: 400 V (350 V nominal voltage)
- Full energy capacity E_{pack}: 54.7 kilowatt-hours
- Number of battery modules: $M = 12$
- Configuration: 96 cells serial, 2 parallel.

Category			Power Cell		Energy Cell		
Model			P41	JP3	E101A	N2.2	E79
Chemistry			NCM/ Graphite	NCM/ Graphite	NCM/ Graphite	NCM/ Graphite	NCMA/ Graphite
Performance	Capacity (Min, 25℃, 0.3C)	Ah	40.8	62.4 (Min, 25℃, 0.5C)	101.8	64.8	78
	Nominal Voltage	Vdc	3.63	4	3.67	3.634	3.69
	Energy	Wh	148	229.6 Wh (Min.)	374	235	287
	Energy Density (Min)	Wh/L	486	389 Wh/L (Min.)	637	556	591
		Wh/kg	226	184 Wh/kg (Min.)	287	266	267
	Pulse Charge Max Current(A)* (10sec, SoC 50%, 25℃, BOL)		380	192	395	146	202
	Pulse Discharge Max Current(A)* (10sec, SoC 50%, 25℃, BOL)		380	192	590	146	546
	Continuous Discharge Performance* (SoC 100%→0%)		5C, 12min, 25℃	-	-	-	3C, 17.3min, 45℃ 3C, 12.4min, 25℃

Figure 2.11: Snippet of the manufacturer's datasheet for LG Energy Solutions cells. The cell *E79* was identified to be used in Renault Zoe ZE50 R135.

It is also provided at [75] that the car has a battery "NCM712". After doing some research, it can be found out that in Renault Zoe models the cells from the manufacturer LG are being used [76]. From the above information the following properties of the used cells from the company LG can be deduced:

- Cell nominal voltage: $U_{N,cell} = \frac{U_{N,pack}}{n_s} = \frac{350V}{96} = 3.646\,\text{V}$
- Pack capacity: $Cap_{pack} = \frac{E_{pack}}{U_{N,pack}} = \frac{54.7kWh}{350V} = 156.286\,\text{Ah}$
- It follows for th coulombic cell capacity: $Cap_{cell} = \frac{Cap_{pack}}{n_p} = \frac{156.286Ah}{2} = 78.143\,\text{Ah}$
- Cell energy capacity: $E_{cell} = U_{N,cell} \cdot Cap_{cell} = 3.646\,\text{V} \cdot 78.143\,\text{Ah} = 285.22\,\text{Wh}$

After calculating these parameters with the additional provided information of battery chemistry being "NCM712", one can go to the manufacturer's website and find the cell-type that is used in the car by comparing the datasheet values with the above calculated parameters. In this case, when comparing the official website and cell datasheets of LG Energy Solutions [77], it can be assumed that the cell *LG Energy Solution's E79* is the used one in this car. A part of its datasheet is shown in Figure 2.11.

For further calculations, one can then extract the parameters from the datasheets if needed. It should be noted that there is a slight difference between the parameters calculated and those supplied by the manufacturer. The reason may be that the way of packaging may affect the parameters. In Chapter 3, a method to deal with this issue is derived.

2.4.3 Battery management system

The battery management system (BMS) is an essential part not only of an electric vehicle, but also of any battery-based storage system. It should ensure optimal and safe usage of

the energy stored in the battery and preventing (or at least mitigating) deterioration and the risk of damage.

The following features should be included by a BMS design [21]:

- (Dis-) charge control
- Battery capacity and efficiency calculations (such as State of Charge and Health)
- Thermal management
- Prediction of battery failure
- Safety and alarm indications when necessary, such as in case of beyond operating conditions.

One can see that the BMS actually governs the behavior of an EV in multiple situations. It is a very complex system, that has some kind of communication channel. This means, in order to simulate charging behavior in an efficient way, one needs to understand how a BMS works and "mimic" it an efficient way. For this reason, in the following, its workflow is evaluated.

The block diagram shown in Figure 2.12, taken from [21] sketches the role of a BMS in an electric car. The BMS monitors real-time, measurable parameters, such as voltage, current and temperature. Each of these can be done for the entire pack or a number of single cells. Temperature monitoring can also include the environment. In addition, some history may also be taken into account: This could be the number of cycles or current profiles the battery pack has experienced in the past.

Out of the aforementioned parameters, a battery state analysis is performed according to some implemented algorithms. This includes the calculation of $SoC, SoH, SoF \ldots$, being SoC the most important among them. Based on the internal state, the BMS decides over the actions to take. That is, in the case of charging, requesting the current, power and voltage accordingly.

Battery state analysis

Let us have a closer look first at the *'Battery state analysis'* step from Figure 2.12. This procedure represents the first step of all BMS actions. It mainly includes SoC estimation, in more advanced systems, SoH and SoF estimation as well. In recent research, the additional parameters of State of Power, State of Life and State of Energy are also proposed as internal parameters. This work focuses on the more common SoC evaluation.

The battery performance irreversibly declines with time and cycling, being a highly complex process. It depends on many parameters and the battery's history. Currently, intense research is being conducted to model this parameter, hoping that quantifiable indicators can be found. The main candidates are DC internal resistance and capacity loss, see equations (2.9), (2.10), (2.12). Battery degradation diagnosis in EVs is based on the calculation of State of Health. There is, however, no standard protocol on that and this method is being currently under development. It can be done either online or offline,

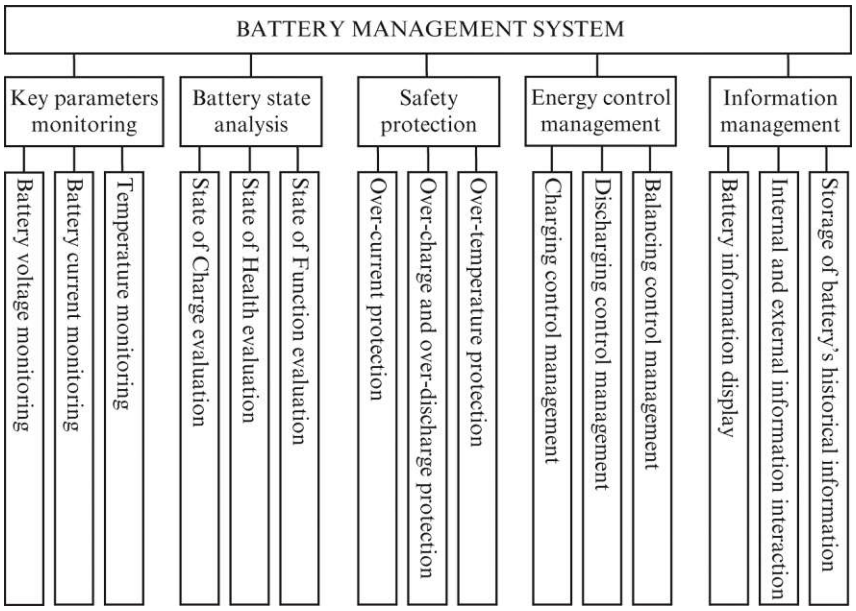


Figure 2.12: Role of a BMS in an EV. Graphics taken from [21].

based on physical, data-driven, machine learning and regression models [21], [78], [79].

State of Charge estimation in BMS systems is one of the most important tasks in battery engineering. The most widely mentioned methods in the literature are listed in the following.

Coulomb counting: This method refers to the calculation of the current SoC from the previous one. Thereby, Q_B from (2.12) at a given time t calculates as

$$Q_B(t) = \int_{t_0}^{t} i_B(\tau)d\tau + Q_B(t_0)$$

i_B denotes the battery current, that actually flows into the battery. It is the charging current provided by the EVSE/socket i minus the losses: $i_B = \eta \cdot i$. η denotes here the charging efficiency related to the *charging equipment*, such as cables etc.

Ideally, considering the conservation of charge the above formula would yield the correct value for Q_B. However, in a real - world BMS system the following issues arise:

1. The initial value $Q_B(t_0)$ has to be known. In a BMS, this requires appropriate logging and consideration of self-discharge effects, while the vehicle is shut down.

2. The integration is done numerically, with some step size Δt. The accuracy is limited on the one hand by the numerical method used, which is a compromise between computational complexity and accuracy. On the other hand it is limited by the step size Δt. A smaller Δt means a higher sampling rate, which imposes higher requirements on the sensor system.

3. Measuring currents comes with a certain noise. This also goes into the integral, meaning, errors accumulate with time.

4. For the sake of completeness, it is to be mentioned, that self discharge of the battery while the charging process is also not captured by this method. However, concentrating on (fast) charging, the used currents and relatively short time this effect is negligible.

Open Circuit Voltage Method: the BMS refers to the SoC by measuring the open circuit voltage V_{oc}. A previously established functional dependency or lookup table of $V_{oc}(SoC)$ is required for this purpose. Moreover, it is assumed that:

1. The mapping: $SoC \mapsto V_{oc}$ is bijective
2. Assume that at the time instant when the battery voltage is being measured, there is no current flowing in there
3. Aging does not affect the open circuit voltage. To put it another way: Degradation is reflected only in capacity fade and in changes in the **passive** components in Figure 2.14.

The second condition makes this method inapplicable in most cases of EV operation, i.e. when driving or charging, because then the battery needs to supply or receive current.

Compromised method: The strategy can be summarized as follows: to exploit the advantages of both previous methods, use Coulomb counting when operating, and use OCV-method when the battery is out of work or at start. This eliminates the accumulated errors from the previous operating cycle and the problem of knowing the initial parameter $Q_B(t_0)$.

The following factors also decrease the accuracy of the real-world SoC estimation in a BMS:

1. Electromagnetic interference between sensors and other electric components of the car. This adds an additional error to the measured value.
2. Difference between the battery cell being measured and the "reference" cell. There may be differences between individual battery cells and the one used as the reference for building the models. This phenomenon has also been observed in 3.3.2. This effect may be due to different ageing conditions of individual cells within a pack compared to the "reference" ageing model as well. Also new cells may have slightly different characteristics even though their behaviour is captured by the same model.
3. Differences between cells in the pack: Non-uniform degradation within the pack (see also 2.1.4) due to position - dependent heat absorption, dissipation and eventually mechanical requirements lead to different ageing and therefore different characteristics within the pack, even if all its cells were completely identical initially.
4. To solve or at least mitigate the above problems, each battery in the pack should be monitored and its history recorded and processed accordingly. This requires a computational, memory and hardware effort that is not feasible for a real application EV management system. Typically, some cells are monitored out of all the battery cells.

There have been methods developed to cope with the aforementioned issues and approve the accuracy of SoC estimation.

The *fuzzy logic control* (FLC) can help improve the inconsistency of series-connected lithium-ion batteries in the pack [80]. Its principle is that instead of using Boolean algebra, where decision values can be either 1 or 0, the algorithms takes "shades" of truth between 0 and 1. To obtain this "value of truth", fuzzy logic uses certain mathematical models.

Kalman filter (KF) algorithm is designed to solve the problem of obtaining accurate information from inaccurate data. In the considered case it can be posed as follows:
How to select the most appropriate SoC values when we only an inaccurate value of the measured voltage/current is available?
The Kalman-Filter algorithm takes as input the model with process noise source(s) and known inputs. The resulting estimator has inputs $[u; y]$ and outputs $[\tilde{y}; \tilde{x}]$. KF takes the system input u (such as voltage) and the noisy model output y (such as SoC) as inputs, and produces the estimated noise-free output $\tilde{y}$ as outputs. It can be a "current estimator" that produces an estimate at the current time step t_n using all available measurements up to t_n. The "delayed estimator" calculates the $\tilde{y}(t_{n-1})$ when data points up to t_n are available [81]. This algorithm is already implemented in commercial software packages such as MATLAB and Python [81], [82].

An important remark is to be left at this point:
In this work, the **simulation** of EV batteries is considered. Therefore, one does not need to bother about real-time noise and its removal. Therefore, Kalman filter and fuzzy logic control are not further discussed in this work. They are mentioned, because this work makes the foundations of an EVSE **emulator**, where measuring real physical quantities is of importance. These methods will become relevant at that stage.

2.5 Standard battery modeling approaches

2.5.1 Models in literature

An overview on the state-of-the art approaches to model battery behavior under different operating conditions is given in [83]–[85]. A summary is given below. The chosen modeling approach *Electric Equivalent Circuit* will be evaluated in a separate subsection 2.5.2.

Depending on how much "physical insight" is used, the battery models can be subdivided into three categories. From the most to less physical understanding they are "white", "grey" and "black box" models [85].

White box, electrochemical model Such models actually use physical laws to model battery system behavior. As sketched in section 2.1, there are multiple chemical reactions going on in the battery. They generate heat which determines the temperature field. The latter is a function of spatial coordinates and time. It once again interacts with the cell reactions. This leads to a system of coupled partially differential equations, whose boundary conditions are determined by ambient factors. This system has no analytical solution, but can be solved with iterative non-linear or finite element- combined with time stepping methods. The electrochemical approach stands out with its accuracy and transparency of outputs. Since these models incorporate temperature changes resulting from reaction processes, they can predict the development in the temperature field with time in a natural way [86]. Such models usually relate mass transport phenomena (such a diffusion) and current laws [85]. The main drawback of these models is that they are very complicated and computationally intensive. Therefore, they are claimed to be rather unsuitable for the use on real-time BMS of electric vehicles. For this reason, they are not used in this work.

Grey box models A sub-category of grey box models are the so-called mathematical models. They implement empirical equations and stochastic models. Their computational effort of solving algorithms are simplified compared to the electrochemical models. Their use is very low, since it is claimed that the relation of current-voltage characteristics with model parameters is not sufficient.

Another approach are the circuit oriented grey box models. They can again be subdivided into the categories simple-, Thevenin-, impedance-, runtime based-, generic- and combined models.

- Simple models use an ideal voltage source with solely one ohmic resistance to model the battery. They do not take into account the charging or discharging rates and nonlinear effects [85].
- Impedance model. The structure has an impedance for both anode and cathode connected in series. Electrolyte, separator, and current collector are represented by a pure ohmic resistor R_0 between anode and cathode. At high frequency, battery current collectors and cables introduce inductance, represented by an inductor L which appears as a positive imaginary part in the Electrochemical Impedance Spectroscopy (EIS) diagram. For the representation see Figure 2.13. Such models can be

characterized with time consuming Electrochemical Impedance Spectroscopy measurements [87]. The proposed impedance model in [87] has 16 parameters, whose values have to be found in the frequency domain. The conversion to time domain might become complex.

- The runtime-based approach uses a complex circuit network to simulate battery runtime and DC voltage response. However, it is thought only for constant discharge current in SPICE-compatible simulators. They are not able to predict a voltage response for varying load currents accurately [88].
- Combined models consist of the combination of different circuit oriented ones, to exploit their advantages simultaneously [89].
- Generic models are claimed to be suitable to model batteries of different technologies undergoing different charging and discharging conditions [89].
- Thevenin models model the transient responses of the battery by using Resistor-Capacitor (RC) networks. This approach has been demonstrated to be accurate and effective in capturing the dynamic behavior of batteries [83], [90], [91]. Due to this property, and being computationally cheap as well as having not too many parameters, this approach is used in this work. In the following, Thevenin models will be referred also as "electric equivalent circuit (EEC)" models.

A schematic of some circuit oriented approaches is shown in Figure 2.13.

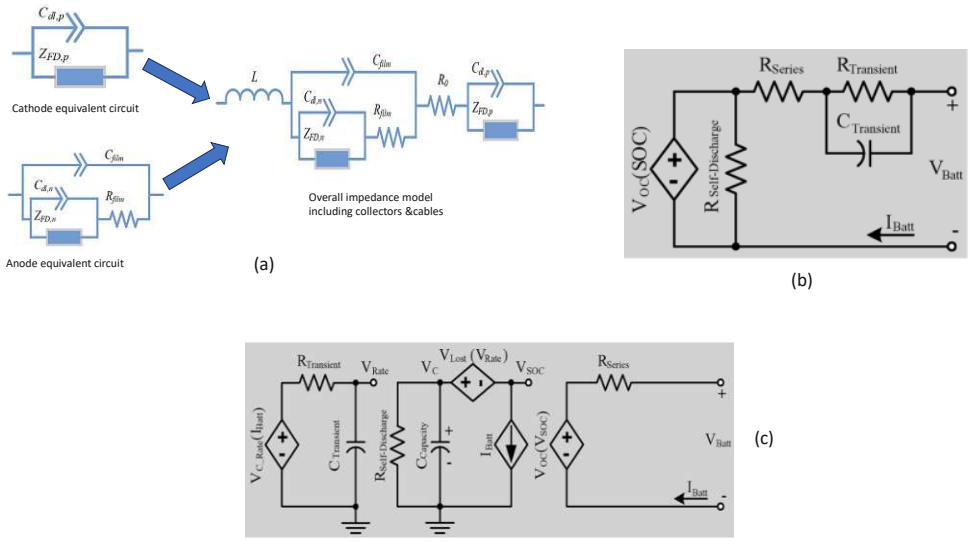


Figure 2.13: Grey box model approaches. (a): Impedance model. (b): Thevenin model, that boils down to the simple model if $R_{\text{Transient}}, C_{\text{Transient}} = 0$. Here it is shown with one RC-element, but it can contain multiple ones. (c): Runtime-based models. Graphics is based on figures from [87], [88]

Black box models They use for battery parameter estimation data-driven approaches which include methods such as fuzzy-based estimation or neural network, artificial neural networks, support vector machines etc [85]. In particular the data-driven approach of

machine learning is considered as very promising. It is highly adaptable to changes in the respective system under study and does not need much resources [85]. Building and applying a black box model usually consists of three phases:

1. Data pre-processing: First, one needs to acquits an accordingly amount and quality of data. It is followed by cleaning, normalizing and transforming it in a form that is suitable for the respective machine learning algorithm.

2. Model training: By using specific algorithms, find the best model describing system behavior based on acquitted data.

3. Estimation: Applies the "knowledge" gained in the previous step. With this knowledge, the required battery parameters can be predicted.

Since the default capabilities required for performing machine learning algorithms are generally included in battery management systems, machine learning techniques are a candidate for modeling charging/discharging profiles in electric vehicles in the future [85].

2.5.2 Electric Equivalent Circuit model and its components

Since the relevant charging algorithms have the battery terminal voltage as control parameter, there is a need to simulate that. Over the years many approaches were developed for this purpose. Due to their simplicity, easy understanding and applicability, the *Electric Equivalent Circuit* models are the most widespread ones. Their generic principle combined with the consideration of charging infrastructure is illustrated in Figure 2.14.

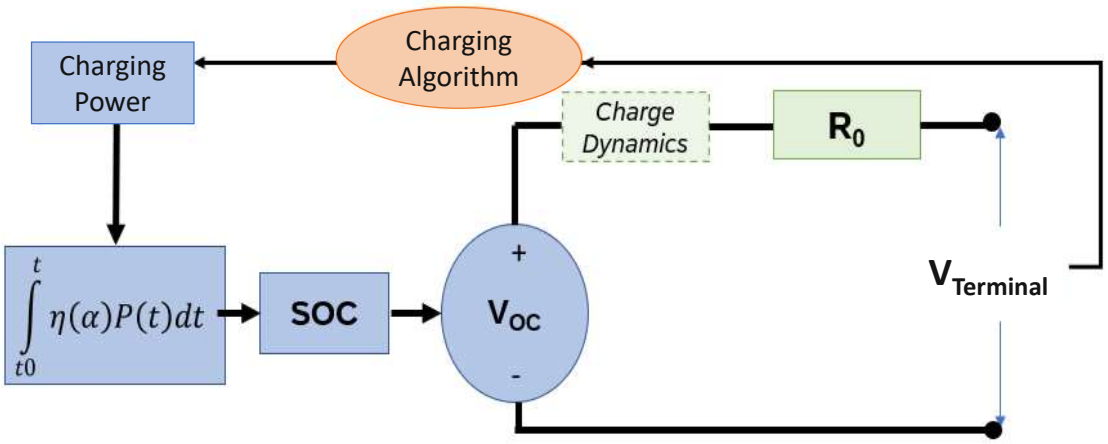


Figure 2.14: Modeling approach of the battery terminal voltage V_{bat}. Note that V_{bat} is referred as $V_{Terminal}$ on this figure.

The idea is the following: One considers the battery as an ideal voltage source, that delivers the *open circuit voltage* V_{oc}. It is dependent on the internal state of the battery, in the first order of its State of Charge, SoC. The measured, no-load (terminal) voltage V_{bat}

($V_{Terminal}$ on Figure 2.14) differs from V_{oc} due to internal losses. The losses are based on the inner resistance of the cell and time/current-dependent characteristics (batteries do not react immediately to currents in general). The aforementioned dynamics effects are modeled via RC-components, whose number can vary. They are situated in the *Charge Dynamics* box on Figure 2.14. The literature standard uses one or two RC components. The finite conductance of battery components leads to a voltage drop as well. This is modeled by the internal, ohmic resistance R_0 on Figure 2.14.

The terminal voltage is the control parameter of most of the currently implementable charging algorithms, due to its easy accessibility. The algorithms again, calculate the charging power $P(t)$. By integrating its current part ($P = U \cdot I$) one obtains the State of Charge of the battery. The SoC determines the open-circuit voltage and in general the value of the other passive components of the equivalent-circuit model, thereby V_{bat}. The charging power can be in general multiplied with some efficiency or filter function $\eta(\vec{\alpha})$). η can again depend on a set of parameters $\vec{\alpha}$, such as the temperature. This results in a loop. In the following, the modeling approaches of the single components shown in Figure 2.14 will be discussed.

Passive parameters

Passive parameters are the internal resistance R_0 and the RC elements "hidden" in the *Charge Dynamics* box in Figure 2.14. R_0 describes the ohmic (internal) resistance of the battery, representing conduction losses. This is in series with one or multiple RC elements, usually one or two are used in the literature. These RC parts characterize the *transient* response of the battery to currents.

EEC circuits can be modeled with the differential equations

$$\begin{aligned} \dot{v}_i &= \frac{-1}{R_i C_i} v_i + \frac{1}{C_i} I(t) \\ V_{bat} &= V_{oc} - R_0 I - \sum_{i_1}^{N_e} v_i \end{aligned} \tag{2.25}$$

v_i is the voltage across the ith RC element. R_i, C_i are its resistance and capacitance, respectively. I is the battery current. In the literature it is usually positive during discharge. This means that care must be taken that the sign is always correct. If N_e RC parts are used to characterize the transient behavior, it is called an N_e order EEC (or Thevenin) model [92]. How to obtain these values is described in 3.2.2.

Open circuit voltage

It is important to note that V_{oc} can only be measured when no current is flowing (otherwise, there is a voltage drop on the passive components). For this reason, the $V_{oc}(SoC)$ curve can be determined by (dis)charging the battery with short and small current pulses, leaving enough time between them. This method allows the transient processes to equilibrate. This results in discrete values of $V_{oc}(SoC_i) := V_{oc,i}$. To obtain an analytic expression for this relationship, based on real-word data, multiple model equations are represented in the literature:

- Nernst model:

$$V_{oc}(SoC) = c_0 + c_1 ln(SoC) + c_2 ln(1 - SoC) \tag{2.26}$$

[93]. The Nernst model is based on the thermodynamic characteristics of the battery. This means that (2.26) incorporates the fact that the thermodynamic voltage V_{th} is given as the difference of the electrostatic potentials $\tilde{\mu}$ between the electrodes. They are governed by Nernst's equation:

$$E = E_0 - \frac{RT}{zF} \cdot \ln(\frac{a_{red}}{a_{ox}}) \tag{2.27}$$

E_0 is the standard cell potential, R is the universal ideal gas constant, T is the temperature in Kelvins, z is the number of electrons transferred in the cell reaction, F is the Faraday constant. a_{red} is the activity of the reduced chemical species and a_{ox} is the activity of the oxidized form.

What is important for the numerical, modeling case in this thesis is the dependence of the activities of the reduced/oxidized species a_{red}, a_{ox}. The activities are dependent on the concentration. In the "battery world" this is accessed by the State of Charge, resulting in the custom equation (2.26).

- Zheng model [94]

$$V_{oc}(SoC) = a_0 + a_1 ln(SoC) + a_2 ln(1 - SoC) + a_3 \frac{1}{SoC} + a_4 SoC \tag{2.28}$$

- Shepherd Model [83]

$$V_{oc}(SoC) = E_0 - \frac{c_1}{SoC} \tag{2.29}$$

that is in its modified version [93]

$$V_{oc}(SoC) = E_0 - \frac{a_1}{SoC} + a_2 \exp\big(-B(1 - SoC)\big) \tag{2.30}$$

- Lam Model [95]

$$V_{oc}(SoC) = a_1 e^{-a_2 SoC} + a_3 + a_4 SoC + a_5 e^{-\frac{a_6}{1-SoC}} \tag{2.31}$$

- Chen and Rincon-Mora model [88]

$$V_{oc}(\mathrm{SoC}) = a_0 \cdot e^{-a_1 \mathrm{SoC}} + a_2 + a_3 \cdot \mathrm{SoC} - a_4 \cdot \mathrm{SoC}^2 + a_5 \cdot \mathrm{SoC}^3 \tag{2.32}$$

These modeling equations shall be called *physical equations (PhEs)* in the following.

3 Simulation of fast charging behavior of EVs

The previous chapter introduced the theory of battery technology. After reading it, the reader has a basic knowledge of the physical/thermodynamic principles of batteries, including their main degradation effects. There is also a review of charging standards and algorithms in EV technology, and modeling approaches of battery behavior under load.

Having established the basics, the next step is to build a model to capture the behavior of the battery (packs) and electric vehicle charging. For this purpose, the well-established Thevenin model with one RC element is used as a basis. As a next step, a novel approach is represented to obtain the EEC's components State of Charge dependency. The latter is done by setting up a numerical model based on adaptive order polynomial fitting, as well as presenting a measurement procedure that provides the raw data for the subsequent numerical modeling. Then, exploiting the linearity of electrical circuits (those with only voltage sources, resistors and capacitors) a method is presented to scale up single battery models to package models. To simulate vehicle charging or voltage profiles under load currents, differential equations are set up with boundary conditions appropriate to the particular application. They are solved numerically using time-stepping methods.

These approaches can achieve the following objectives:

- Simulation of individual battery cells when subjected to an arbitrary load current
- Simulation of arbitrary assembled battery packs
- Simulation of the most common fast charging algorithms, taking into account EVSE limitations

3.1 Summary of approach

The problem of simulating EV or battery pack charging is subdivided as following:

1. Simulate and capture the behaviour of a single battery cell.
2. Take this cell, and build up a pack out of it.
3. Scale the pack using the linearity of electric circuits.
4. Consider operational EVSE limitations.
5. Define the charging algorithm.
6. Run the simulation.

A graphics connecting these ideas is shown in Figure 3.1. How single steps on this figure are done is the subject of the following sections.

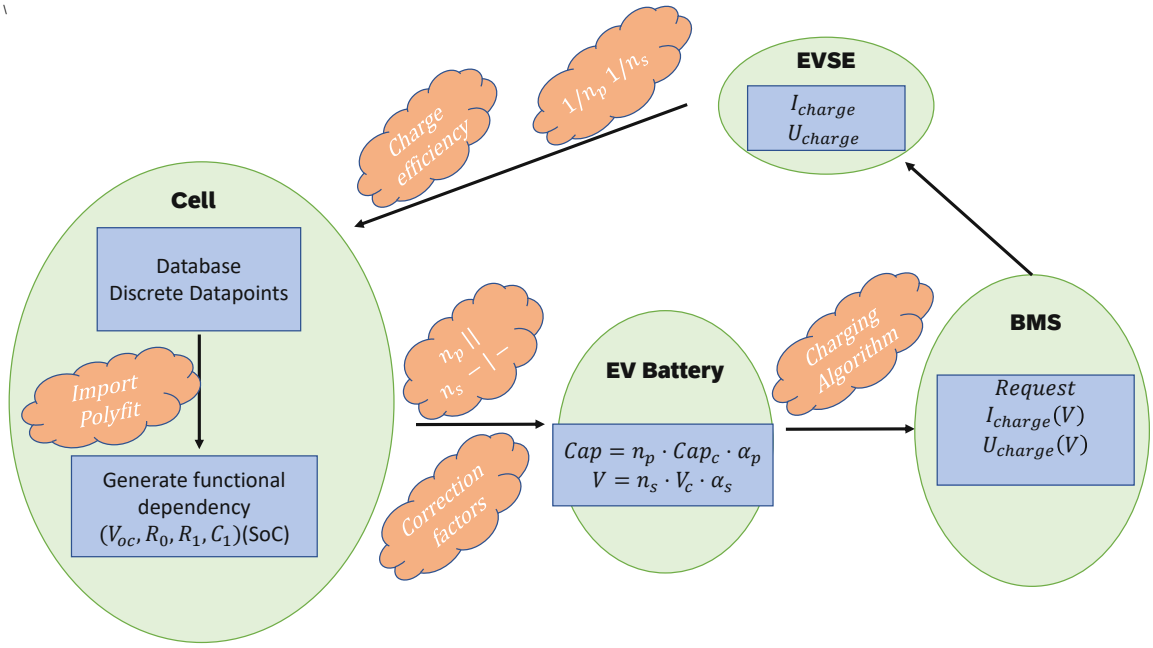


Figure 3.1: An approach that will be followed in the following sections to develop a simulation software for EV and battery (dis-) charging.

3.2 Implementation of the electric equivalent circuit model

In this section, the framework for parameterizing the State of Charge dependency of EEC model components is established. The presented method delivers the X values at discrete SoC points. This is based on the examination of transient voltage profiles after current pulses that are applied to the battery whose parameters are of interest.

3.2.1 Target model

The principle of Thevenin-based models is described in 2.5. A common subcategory in literature is the case where only one RC element is used to model transient battery behavior. This means that the value of N_e from (2.25) is $N_e = 1$. Therefore one can imaginally replace the battery by the electric circuit shown in Figure 3.2. This means:

- The open-circuit voltage of the battery is modeled by an ideal voltage source, V_{oc} from Figure 3.2.
- Finite conductance of battery components are modeled by a standalone ohmic resistance, R_0 from Figure 3.2.
- Transient response of battery voltage on currents are modeled with an RC-network consisting of solely one capacitor C_1 and resistor R_1 from Figure 3.2.

This approach was reported to be effective in capturing the dynamic behavior of Li-ion batteries [83], [90]–[92], [95]. Moreover, it is computationally cheap and requires solely 5 parameters. Therefore it is used in this work to further develop battery models.

An important, left-over issue is now to get the exact values of V_{oc}, R_0, R_1, C_1. It has already been discussed that properties of a chemical system depend on the activities of the reacting species. The activity, depending on the chemical concentration is implicitly incorporated in the term of State of Charge. Therefore, it is expected that V_{oc}, R_0, R_1, C_1 depend on the SoC. It is shown in the literature as well as in the following that this expectation holds.

As a next step in establishing the battery model, it is presented how to obtain the values of V_{oc}, R_0, R_1, C_1 at a certain State of Charge.

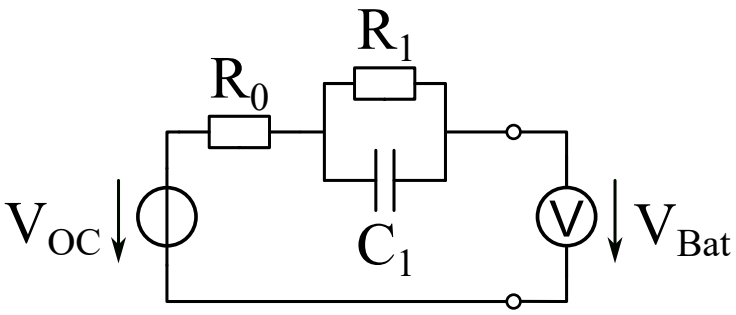


Figure 3.2: Target model, graphical representation.

3.2.2 Obtaining EEC parameter values

Theoretical derivation

In this subsection the theoretical considerations how to obtain parameters from Figure 3.2 are derived. This is done by describing the time-dependent behavior of the respective electric circuit by differential equations and discretizing them. Afterwards, coefficients of the discretized equations are fitted on observations, i.e measurable voltage and current profiles.

For the first-order Thevenin model $N_e = 1$, therefore (2.25) reduces to

$$\begin{aligned} V_{bat} &= V_{oc} - R_0 I - v_1(t) \\ \frac{dv_1}{dt} &= -\frac{1}{R_1 C_1} v_1 + \frac{1}{C_1} I(t) \end{aligned} \tag{3.1}$$

I denotes the battery current ((dis-)charging current). One can obtain the solution by integrating the first order ordinary differential equation (3.1). This cannot be done analytically in many cases, since both v_1 and I depend explicitly on the integration variable t. A promising alternative to do the integration numerically is using time-stepping methods. When using the first-order forward explicit Euler method, at the time-step k using a step-size of t_s, the transient voltage on the RC-element v_1 is given as [92]

$$v_1(t_k) = v_1(t_{k-1}) \cdot e^{-\frac{t_s}{R1C1}} + R_1 \cdot \left(1 - e^{-t_s/R_1 C_1}\right) I(t_{k-1}) \tag{3.2}$$

Obtaining the model parameters V_{oc}, R_0, R_1, C_1 (that are SoC dependent) can be done by looking at the transient response at a current pulse as described in the following. The derivation is based on the method described in the works [96]–[98]).

Multiple (dis)charge pulses can be applied to the battery. It reacts to them as follows: it comes first to an initial voltage drop or increase, and then to a transient part determined by the RC-component values. Figure 3.3 illustrates a typical voltage response profile on current pulses. Let us discuss the single sections of the figure:

1. $t \leq t_0$: The battery has been at rest for a sufficiently long time (without currents). Its voltage V_{bat} equals V_{oc}.

2. $t_0 \leq t \leq t_1$: At the beginning of the discharge pulse, an initial voltage drop V_0 takes place due to the ohmic resistance R_0. Then the voltage decreases because SoC is doing so due to the discharging process, with an additional contribution of the RC-network (v_1) described by (3.1).

3. $t_1 \leq t \leq t_2$: After drawing no more current from the battery, there is an instantaneous jump in V_{bat} again due to R_0. Since no current is being drawn, and the resting interval is sufficiently short to neglect self-discharging effects, the State of Charge can be treated as constant in this period.

$$SoC = const := SoC(t_1),\ t \in [t_1, t_2] \tag{3.3}$$

   After the initial voltage drop at the instant when the discharging current has been turned off, V_{bat} converges to $V_{oc}(SoC = SoC(t_1))$ with a characteristic transient profile. The transient profile is determined by R_1, C_1.

4. $t_2 \leq t \leq t_3$: When a charging current pulse is applied there is an instantaneous jump in V_{bat} again due to R_0. Then the voltage increases because SoC is doing so due to the charging process. There is a contribution of v_1 according to (3.1) as well.

5. $t_3 \leq t$: After charging no more current into the battery, there is an instantaneous drop in V_{bat} due to R_0. Then, V_{bat} converges to $V_{oc}(SoC = SoC(t_3))$ with a transient profile, characteristic to R_1, C_1. Since no current is being drawn or charged, and the resting interval is sufficiently short to neglect self-discharging effects, the State of Charge can be treated as constant in this period.

$$SoC = const := SoC(t_3),\ t \geq t_3 \tag{3.4}$$

In the following, a method with N, solely discharge pulses to obtain a discrete set of $\{SoC_k, V_{oc,k}, R_{0,k}, C_{1,k}, R_{1,k}\}, k \in [1, ..., N]$ is derived based on the above considerations.

Considering an immediate turning off the current at a time instant t_k, the value of I from (3.1) reduces to zero. This holds until t_m, when the new discharge pulse starts.

$$I(t) = 0,\ t \in [t_k, t_m] \tag{3.5}$$

In this current-free interval (3.1) reduces to, when one denotes $R_1 \cdot C_1$ by the time constant τ:

$$\frac{dv_1}{dt} = \frac{-1}{\tau} v_1(t),\ t \in [t_k, t_m] \tag{3.6}$$

The equation (3.6) can be solved analytically by applying separation of variables:

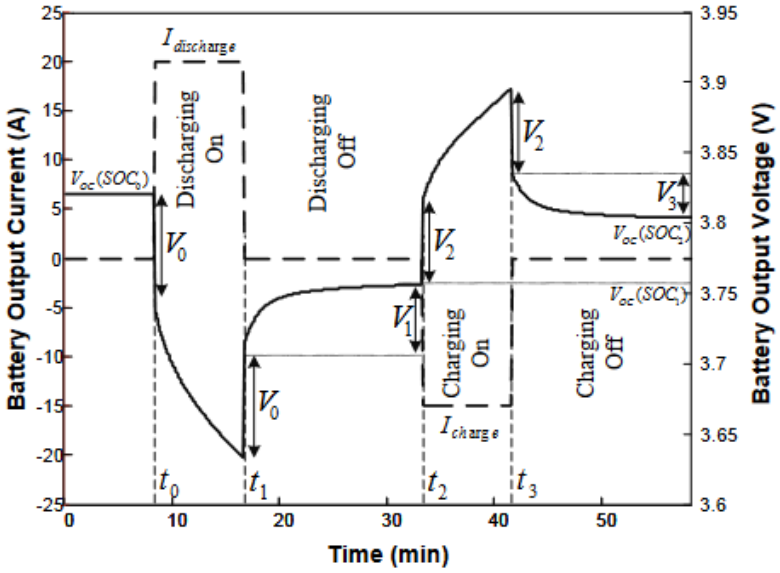


Figure 3.3: Characteristic voltage profile of the battery responding to current pulses. Taken from [97].

$$\begin{aligned} \int_{t_k}^{t} \frac{dv_1}{v_1} &= \int_{t_k}^{t} \frac{-dt}{\tau} \\ \ln\big(v_1(t)\big) - \ln\big(v_1(t_k)\big) &= \frac{-\Delta t}{\tau} + const \\ v_1(t) &= v_1(t_k) \cdot \mathrm{e}^{-\Delta t/\tau} + const \end{aligned} \tag{3.7}$$

With $\Delta t := t - t_k$.

Derivation of (3.7) and the above enumerated points mean that when applying discharge pulses and turning off current at the timestep t_k, one can record the voltage profile V_{bat} until the beginning of the next pulse t_m, and can fit it to the custom profile:

$$V_{bat}(t) = A_k + B_k \cdot \mathrm{e}^{-b_k \cdot \Delta t},\ t \in [t_k, t_m] \tag{3.8}$$

When doing in total N pulses, means $k \in \{1, \dots N\}$, $m \in \{1, \dots N\}$.

Considering points 1-5, it holds if the rest interval $(t_m - t_k)$ is large enough for the battery voltage to equilibrate, but sufficiently small to neglect self-discharge effects:

$$\begin{aligned} SoC = const := SoC(t_m) = SoC(t_k),\ t \in [t_k, t_m] \\ V_{bat}(t_m) \approx V_{oc}\big(SoC(t_m)\big) = A_k \\ v_1(t_k) = B_k \\ \tau(t_k) = \tau(t_m) = R_1\big(SoC(t_m)\big) \cdot C_1\big(SoC(t_m)\big) = \frac{1}{b_k} \end{aligned} \tag{3.9}$$

By fitting a measured voltage profile on known current pulses, one can obtain the values of the parameters A_k, B_k, b_k from (3.8). Combined with (3.9) this method delivers the values of the first-order EEC parameters as follows:

$$
\begin{aligned}
V_{oc}\big(SoC(t_m)\big) = V_{oc}\big(SoC(t_k)\big) &:= V_{oc,k} = A_k \\
R_0\big(SoC(t_k)\big) &:= R_{0k} = \frac{\Delta V_k}{I_k} \\
R_1\big(SoC(t_k)\big) \cdot C_1\big(SoC(t_k)\big) &:= R_{1k} \cdot C_{1k} = \frac{1}{b_k}
\end{aligned}
\tag{3.10}
$$

In (3.10) the internal resistance R_0 is calculated from the instantaneous voltage jumps ΔV_k at the time instants when discharging currents are turned off. I_k stands for the discharging current, that has to be constant for every respective discharging interval.

What is left is to obtain an unambiguous expression for R_{1k}, C_{1k}. This can be done by combining (3.7), (3.8) (3.10): Denote with T_k the length of the kth current-free period. In the used n, m-notation, it is

$$
T_k = t_m - t_k, \;\; k = m \tag{3.11}
$$

To get graphical representation, on Figure 3.3 T_1 would equal $T_1 = t_2 - t_1$. Note that Figure 3.3 is not consistent with the used n, m-notation.

After inserting the value of T_k into (3.7), (3.8) (3.10) the expressions for R_{1k}, C_{1k} can be obtained as follows:

$$
\begin{aligned}
R_{1k} &= \frac{B}{\left(1 - \mathrm{e}^{-b_k \cdot T_k}\right) \cdot I_k} \\
C_{1k} &= \frac{1}{b_k \cdot R_{1k}}
\end{aligned}
\tag{3.12}
$$

Note that the State of Charge has to be computed by knowing the length and height of the current pulses. With every pulse a certain amount of charge is extracted from the battery. This results in a new State of Charge at the end of the pulse k. The State of Charge after a pulse with a duration Δt_k and current I_k is

$$
SoC_k = SoC_{k-1} - \frac{\Delta t_k \cdot |I_k|}{Cap} \tag{3.13}
$$

To summarize, the presented method, based on the derived equations and parameter fitting, yields a collection of points $\{SoC_k, V_{oc,k}, R_{0,k}, C_{1,k}, R_{1,k}\}, k \in [1, ..., N]$. This is done by performing N discharge current steps with sufficiently long rest periods in between. At the same time, it is necessary to record the voltage and current profile, followed by parameter fitting. Figure 3.4 recapitulates the procedure.

Algorithm: For Loop to calculate R_0, R_1, C_1 to N distinct SoC values

For (k, m) in range $(1, N)$:

- Record between t_k, t_m after discharge for time z_k with current I_k
- Current free interval length $T_k = t_m - t_k$
- After turning off the current at time t_k, measure voltage jump ΔV_k:

$$R_0(SoC(t_k)) = \frac{|\Delta V_k|}{I_k}$$

- No current $\Longrightarrow$ $SoC = \text{const}$. Examine charge dynamics profile.
- Fit transient voltage profile on:

$$f(\tilde{t}) = A + Be^{-b\tilde{t}}$$

$$\tilde{t} = t - t_k$$

- From coefficients A, B, b resulting from the fit, calculate:

I) $A = V_{oc}(SoC(t_k))$

II) $\tau = \frac{1}{b}$

III) $R_1 = \frac{v_1(\tilde{t}=0)}{(1-e^{-T_k/\tau})} = \frac{B}{1-e^{-T_k b}}$

IV) $C_1 = \frac{1}{R_1 \cdot b}$

They belong to $SoC = SoC(t_k)$.

Figure 3.4: Algorithm to generate an array of SoC values and corresponding passive components.

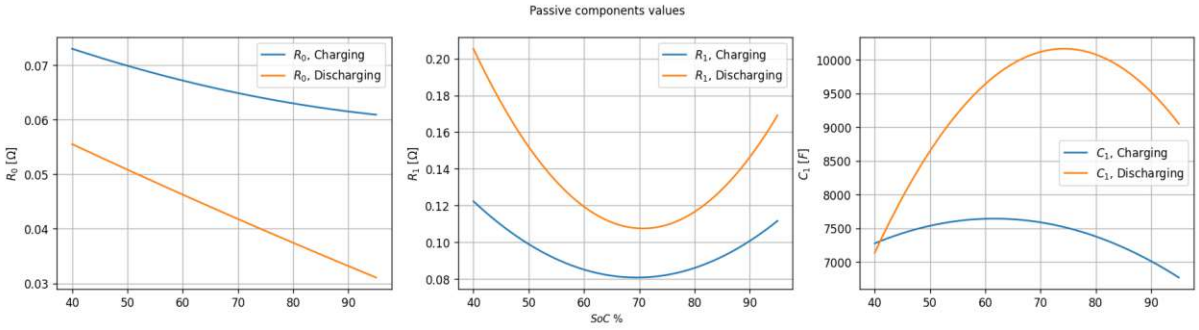


Figure 3.5: Trend of passive components values over SoC, modeled by polynomial of order 3 and parameters as presented in [92]. In this paper, only $SoC \geq 0.4$ was considered.

One remark is left over:

In the work [92] separate parametrization of resistance and capacitance values for charging and discharging is proposed. Their values obtained in the publication are re-plotted in Figure 3.5. It is noticeable, that their values actually show a higher deviation between the charging and discharging condition. Due to resource issues, in this work the parametrization is followed based on discharge pulses only. However, it has been shown

to be sufficiently accurate for charging conditions as well. See Chapter 4. Moreover, it is a common practice in the literature to consider only discharge current pulses for battery model parameterization.

3.3 Establishing smooth functional dependencies of the electric equivalent circuit model parameters

In this section a procedure that establishes smooth, even C^∞ continuous functional dependencies between the State of Charge and EEC components is derived. This is done by interpolating polynomials, whose optimal degree is determined adaptively based on the provided input data, i.e. set of points $\{SoC_k, V_{oc,k}, R_{0,k}, C_{1,k}, R_{1,k}\}_k$.

3.3.1 Numerical approach

The idea is derived in the following by taking the open-circuit voltage V_{oc} and its State of Charge dependency as example.

PhEs are based on some physical considerations, however, they cannot capture the physical processes completely, and are in the end just some fitting-custom models. The true nature of the $V_{oc} - SoC$ is described by some *unknown* function:

$$V_{oc} := f(SoC) \tag{3.14}$$

One has the "true" knowledge only in a set of discrete pair points (if measurement errors are neglected and it is assumed that the actual State of Charge is known):

$$\{x_i, f(x_i)\} := \{x_i, f_i\},\ 1 \leq i \leq N \tag{3.15}$$

when having N measurements points. This actually lets us arrive, considering from a numerical point of view, at an interpolation problem. Thereby the goal is to represent some unknown function $(f(x))$ by a set of well known basis functions $\{X_k(x)\}$, $1 \leq k \leq M$:

$$f(x) \approx \tilde{f}(x|\{X_k\}, \vec{c}) = \sum_{k=1}^{M} c_k X_k(x) \tag{3.16}$$

With the unknown coefficients c_k that can be put in the vector $\vec{c}$. They are usually calculated by minimizing the sum of squared errors between interpolated function values $\tilde{f}_i$ and known values of the "true" function f_i:

$$\vec{c} = \text{argmin}_b \,|\sum_{i=1}^{N} \left(f_i - \tilde{f}(x|\{X_k\}, \vec{b})\right)^2| \tag{3.17}$$

From a numerical point of view, there is no reason to rely on the custom equations from the literature. Moreover, fitting their coefficients numerically require "good enough" initial guesses. Without these initial guesses being appropriate, algorithms solving (3.17) (by an optimization problem) do not converge. As a remark: such algorithms are already implemented in several numerical packages, such as the `curve_fit` method from Python's `scipy.optimize` library [99].

Therefore a simply-seeming polynomial fit is considered, whose basis is just the monomial basis:

$$X_k(x) = x^{k-1} \tag{3.18}$$

This leads to the representation (note the index shift, sum begins now at $k = 0$):

$$\tilde{f}(x|\{X_k\}, \vec{c}) = \sum_{k=0}^{M-1} c_k x^k \tag{3.19}$$

For N observations (tabulated $x_i - f(x_i)$ values) one can create the so called $(N \times M)$ *design matrix* $\tilde{T}$. Modeling the observed quantity $f(x)$ by polynomial of order (M-1), one can represent the polynomial interpolation problem due to its linearity (note the sum in Eq. 3.19) via matrix form:

$$\vec{f} \approx \tilde{T}\vec{c} \tag{3.20}$$

with $\tilde{T}$ being

$$\tilde{T} = \begin{pmatrix} x_1^0 & x_1^1 & \dots & x_1^{M-1} \\ \vdots & \vdots & \vdots & \vdots \\ x_N^0 & x_N^1 & \dots & x_N^{M-1} \end{pmatrix} \tag{3.21}$$

The lower index i stands for the ith observation. The solution vector (=vector of coefficients c_k from Eq. 3.19) is given by solving the optimization problem shown in (3.17) and inserting for $\tilde{f}$ the expressions (3.21), (3.19).

$$\vec{c} = \left(\tilde{T}^T \tilde{T}\right)^{-1} \tilde{T}\vec{f} \tag{3.22}$$

The formula (3.22) for "optimal" interpolating polynomial coefficients c_k is obtained as described below:

One needs to build the first derivative in the bracket of (3.17) respective to $\vec{b}$ and force the expression to equal zero. One solves it for $\vec{b}$. The solution then delivers the right-handside of (3.22), representing the optimal interpolating coefficients c_k for the respective data-set $\{x_i, f_i\}$. They are summarized in the solution vector $\vec{c} = \vec{b}_{optimal}$. The reader can find an example for deriving this solution for a more simple problem on concrete data in [100].

A very important property of the polyinomial fitting method is that equation (3.22) always has a solution when extrapolating different data (x_i) points: In this case, the rows and columns of $\tilde{T}$ are linearly independent, and then $(\tilde{T}^T\tilde{T})$ is invertible. Thereby, this method ensures a solution for interpolating the open-circuit voltage over State of Charge behavior. Moreover, one does not even require any initial guesses that have to be good enough for other optimization algorithms.

Although the derivation might seem to be complicated, one is not even required to create and write the algorithm itself. Using `polyfit` from the `numpy` Python library, this is exactly the process being done when returning the coefficients [101].

The aforementioned points are not the only advantage of the polynomial fit. There is a mathematical theorem providing an upper error bound (or at least, convergence) for the accuracy of the interpolation:
Let $p_n(x)$ be the polynomial of degree n that interpolates $f \in C^{n+1}[a, b]$ (f is $n + 1$ -times continuously differentiable on the interval $[a, b]$) in the points $x_0 + i \cdot h_i$. In this

notation, h_i represents the distance between x_{i+1} and x_i. The function f is to be interpolated on the interval $[a, b]$.

Then, the interpolation error can be bound by

$$||f - p_n||_{\infty,[a,b]} \leq (b-a)^{n+1} \cdot \frac{||f^{(n+1)}||_{\infty,[a,b]}}{(n+1)!} \tag{3.23}$$

By the term "$||.||_\infty$" the supremum norm (in other terms: Chebyshev or infinity norm) is meant.

The upper bound provided in (3.23) has the following relevance: Given that the function is smooth enough, the interpolation converges very fast with higher degree n to the true function n, and it does it not only point-wise, but in an absolute sense (note the supremum norm!). Additionally, this method is also "safe" when choosing higher n for some function $f \notin C^{n+1}[a, b]$: the error will not get worse, one just does not get this ideal convergence ratio. The described behavior is demonstrated in Figure 3.6 where the convergence behavior of polynomial interpolation on a logarithmic scale is shown:

On the left sub-figure a smooth C^∞ function $f_1(x) = x^2 \cdot e^{2x}$ on [0,1] is interpolated. One sees the convergence rate is of approximately (even better) as $\frac{1}{(n+1)!}$. After the degree 14 machine precision is reached, that is why one cannot observe any improvement.
"Reaching machine precision" means that the interpolation error $|f - p_n|$ is smaller than the relative approximation error due to rounding in double-precision (64 bit) floating point number system according to the IEEE 754 standard on computer arithmetic [102]. Simply put, it means that the error is for $n \geq 14$ smaller that the "double" resolution $\epsilon = 2^{-53} \approx 10^{-16}$ of computers, therefore the machine cannot "see" that the interpolation error is getting smaller.

On the right sub-figure a non-smooth (not even C^1 at $x = 0$) function $f_2(x) = |x|^{1/2}$ on [-1,1] is interpolated. The convergence rate is thereby far from the optimal one, but the error is still decreasing with n.

More on the theory of polynomial fitting and numerical analysis see in the state-of-the art book of Stoer et al. [103].

Finally, let's compare the performance of some PhE and in this work proposed method to interpolate $V_{oc}(SoC)$. The procedure is the following:

1. First, tabulated $SoC - V_{oc}$ values are extracted from MATLAB's *"Battery (Table-Based)"* [104] block: A discrete set $SoC - V_{oc}$ points for the battery *Panasonic NCA103450* is taken.

2. Fitting is done by the polynomial method, $n = 15$. The polynomial coefficients are obtained by the `polyfit` method of the `numpy` Python library.

3. Finding the optimal parameters for the Nernst, Zheng, modified Shepherd, Lam and Chen & Rincon-Mora model (see equations (2.26)-(2.32)). Optimization algorithm is done by the `curve_fit` function from the Python `scipy.optimize` library. As initial guess a trial of either 0.0 or 0.1 for all the parameters is set.

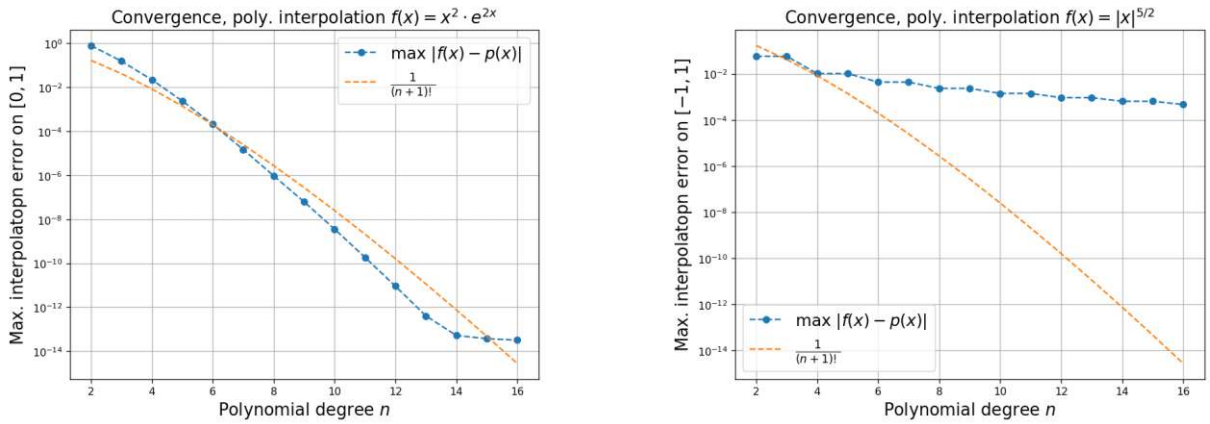


Figure 3.6: Convergence of polynomial interpolation on a logarithmic scale. Left: interpolating a smooth function on [0,1]. Right: Interpolating a non-smooth function on [-1,1]. The convergence rate is thereby far from the optimal one, but error is still decreasing with n.

Finding the optimal values for the PhEs' coefficients to appropriately interpolate the dataset is not possible this way for this input data, see Figure 3.7. An exception is the Rincon-Mora model, that, except the first term, is a polynomial of degree 3. On the other hand, one can see that the polynomial interpolation represents the dateset very well.

Note that:

- With appropriate initial coefficients the PhEs probably would have delivered more accurate results. This comparison only presents the performance of the methods by making a quick fit without bothering about fine tuning the initial parameters.
- To fit non-linear models on curves, alternatively, the function `least_squares` from Pythons `scipy.optimize` library could have been used. This however also requires initial guesses of independent variables. This method was not examined due to this requirement.

As the final step, the maximal relative error e_{max} of the two well-performing methods is compared. By this term what is meant is:

$$e_{\max} = \max \left\{ \left| \frac{y_i - \tilde{y}_i}{y_i} \right| \right\}_{i=1}^{M} \tag{3.24}$$

In the above expression, y_i denotes the ith data point of V_{oc}. $\tilde{y}_i = \tilde{f}(SoC_i)$ means the value interpolated by the function $\tilde{f}$ at the point SoC_i. In total M data points are considered.

The results with the considered dataset are:

- e_{max}, polyfit: 0.0054
- e_{max}, Rincon Mora model: 0.0179
- Relation: 3.3073

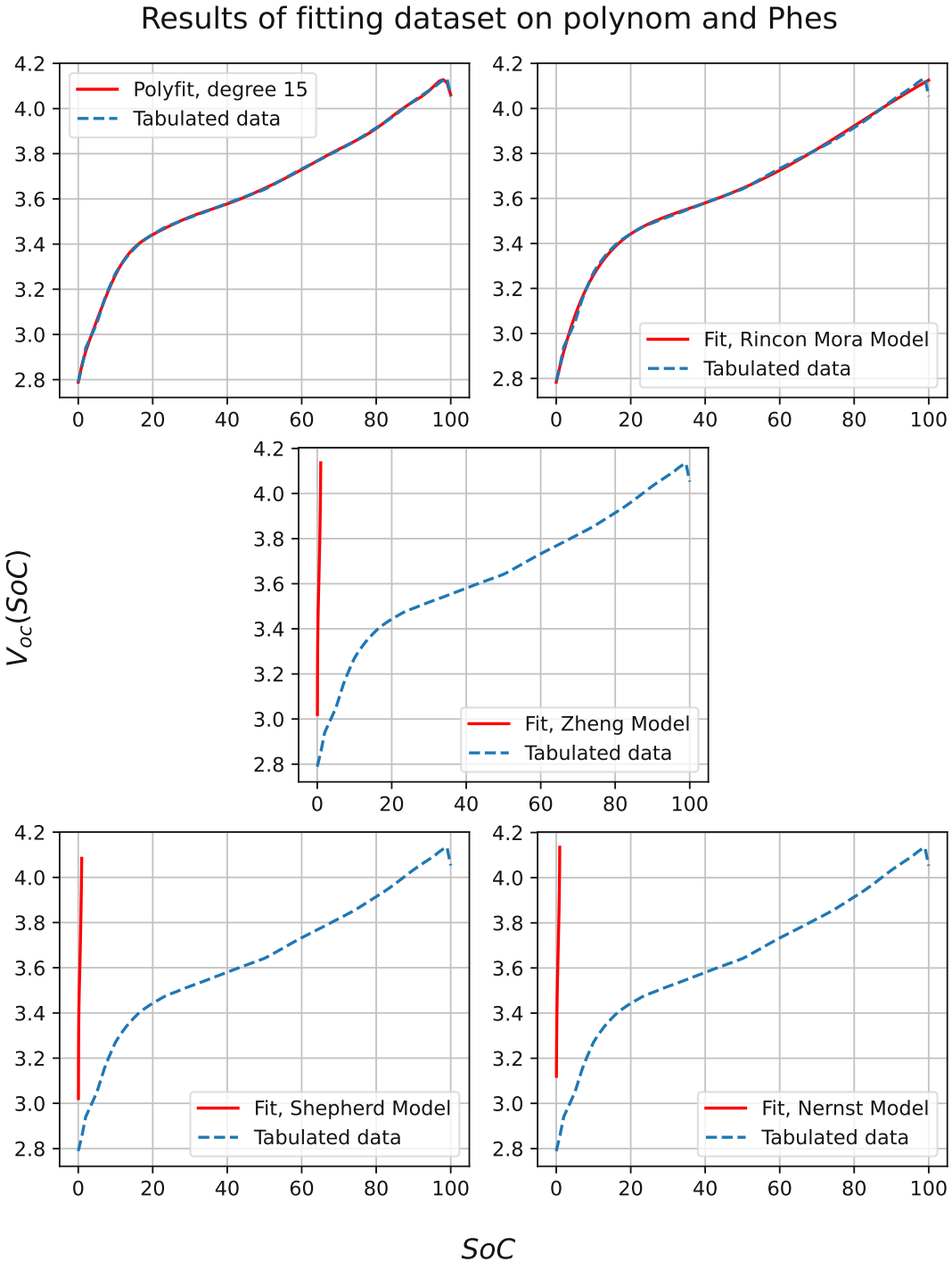


Figure 3.7: Comparison of the interpolating curves obtained by fitting on polynomials and PhEs. There was no convergence at all for Lams equation, for this reason it is not plotted.

Meaning, the polyfit method performs here three times better than the best-performing PhE, when doing a quick "on-the-fly" interpolation. Now the further usage of polynomial fitting to model the SoC-dependent parameters from Figure 3.2 can be considered as justified.

From now on, for the sake of simplicity and brevity the variable X can stand for the respective parameters on Figure 3.2:

$$X \in \{V_{oc}, R_0, R_1, C_1\} \tag{3.25}$$

3.3.2 Concrete workflow proposal

At this point, approaches have been established to

- Have a target model which describes the transient behavior of a battery under load current. This is the content of 3.2.1.
- Obtain the values of this model at discrete State of Charge points. It is described in 3.2.2.
- A method to not only have a discrete set of points of the EEC model's parameters, but also be able to model them with smooth, C^∞ functions. The idea is based on polynomial fitting. This is the content of 3.3.1.

As a next step, a method applicable in the practice based on the listed considerations is presented. The overall goal is to parameterize the first order Thevenin model presented in 3.2.1. It is based on *optimized* (in terms of time and equipment resources) pulse discharge measurements whose foundations are described in 3.2.2. They are done on a set of cells belonging to the same battery cell type. The smooth functional dependencies describing the $X(SoC)$ behavior are established by doing *adaptive* polynomial fitting on the discrete data points which are the result of the previous step. The adaptive approach allows to take into account the differences between single cells belonging to the same model. At the same time it has optimized coefficients for the respective polynomial model.

The manuscript on the method that is presented in the following and developed by the author has been accepted for publication by IEEE to the 2024 Annual Conference of the IEEE Industrial Electronics Society in Chicago, Illinois. Its single steps in order of execution are discussed in the following.

1. Capacity Determination of the Cells.

Battery cells have a nominal capacity Cap_{Nom} (in Ah) given by the manufacturer as specified data. However, it is a well known phenomenon that battery capacities vary between the single samples. Besides, capacity is dependent on the operating conditions, such as the rate of charge and discharge. At first, the cells are fully charged and discharged with a certain C-rate to determine their capacity. After doing so, the respective *actual* capacity for every cell m is available. Having N_C cells in the set it means there is a set $\{Cap_1, \dots Cap_{N_C}\}$ after this step.

2. Pulse discharge measurements.

Pulse discharge measurements are started with fully charged batteries: $SoC_m = 1, \ \forall m$. Then, discharge current pulses in defined time intervals of length Δt_k are applied. Between these intervals, the cells are let to "relax". This means, it needs to be waited for Δt_r between two discharge current pulses. The discharge current is chosen uniformly for all batteries, so that it corresponds to a C rate of C_d, respective to the *average* determined capacity of the set $\overline{SoC}$:

$$\overline{SoC} = \text{mean}\left(\{SoC_1, \dots SoC_{N_C}\}\right) \tag{3.26}$$

In the following, this discharge current corresponding to a C rate of C_d, respective to $\overline{SoC}$ is called $\overline{I}$.

According to the above considerations, the pulse discharge procedure for battery $m \in \{1, \dots N_C\}$ is as follows:

- With every pulse, some charge from the battery is extracted. This results in a new state of charge for the battery m at the end of the pulse i:

$$SoC_i^m = SoC_{i-1}^m - \frac{\Delta t_k \cdot |\overline{I}|}{Cap_m} \tag{3.27}$$

- At the moment when a discharge period is finished and the current drops to zero, an instantaneous voltage jump can be observed. This is followed by a transient phase due to the RC-element as described in 3.2.2.

- The fact that individual cells show some variation in their true capacity can be exploited. This means that even if discharge pulses of the same length and current are applied, the resulting steps of SoC, ΔSoC^m for cell m will be different from the others. This will be important when doing the polynomial fitting.

At the end of this procedure one has for every cell m

- A time-vector $\vec{t}^m$, containing the discrete set of time-points, recorded in intervals according to the sampling time t_s. The $\vec{t}^m$ vectors are the same for all m.

- A current- vector $\vec{I}^m$, containing the discrete set of current values. They are recorded at the time instants being elements of $\vec{t}^m$. Assuming an ideal equipment and no measurement errors the elements of $\vec{I}^m$ are either 0 or $\overline{I}$, and the $\vec{I}^m$ vectors are the same for all m.

- A voltage-vector of the measured battery voltage recorded at the time instants being elements of $\vec{t}^m$: $\vec{V}_{bat}^m$. Even if having an ideal equipment and no measurement errors, this is expected to be *non*-uniform for the cells m. This is because of two reasons: First, the State of Charge steps due to the pulses with $\overline{I}$ are different for all cell m, see (3.27). Secondly, not only in the capacities, but also the EEC components of the single samples can be differ in general. It means that not all samples of some manufactured system (e.g. battery cells) behave completely identically, which is a well-known and observed property.

These above arrays $\vec{t}^m, \vec{I}^m, \vec{V}_{bat}^m$ are the input of the subsequent step, when the $SoC_i^m - X_i^m$ pairs for every cell m, every pulse i are extracted.

3. $SoC - X$ data points

After **step 2.**, the next task is to obtain the $SoC_i^m - X_i^m$ pairs for every cell m, every pulse i. One takes for every cell m the recorded arrays $\vec{t}^m$, $\vec{I}^m$,$\vec{V}_{bat}^m$ and according to the method described in 3.2.2, needs to conduct the parameter fitting on the voltage-time-current profile of all cells separately. This delivers for every cell m a set of discrete points $\{SoC_i^m, V_{oc,i}^m, R_{0,i}^m, R_{1,i}^m, C_{1,i}^m\}_i$.

4. Adaptive polynomial fitting

After the respective SoC_i^m - X_i^m data points are available, the idea is to use the method of adaptive polynomial fitting, based on the following brief theoretical considerations: A higher-degree polynomial allows the fitted model to capture a richer variety of relationships. A potential flaw of this method is that it makes the fit more dependent on the individual observations, which adds random variability and can lead to worse predictions [105]. One model is considered as an overfit if its predictions are not better than those of another simpler (in the considered case: lower polynomial degree) model [105]. Adaptive order polynomial fitting on the recorded data of *multiple* samples of a certain system (here: the respective battery cell-type) takes the variations between the single cells into account, while smoothing the effect of noise sources. It is also intended to prevent excessive focus on the properties of a single sample.

```
for run in range(0,M):
    residuals = []

    valiadation_set = dataset[ shuffled_indices1 ]
    modeling_set = dataset[shuffled_indices2 ]

    for degree in range(nMin,nMax):
        coeffs_train = polyfit(x=SoC[modeling_set], y=Y[modeling_set], degree)
        poly = polynom(coeffs_train)
        residuals.append( (poly(SoC[valiadation_set]) - Y[valiadation_set])^2  )

    opt_degree.append( WHERE(residuals == MIN)  )


OptimalDegree = round( opt_degree.mean() )
```

Figure 3.8: Pseudo code for the adaptive fitting algorithm which is applied to each first order Thevenin-model parameter $X \in \{V_{oc}, R_0, C_1, R_1\}$. Note that in the pseudo-code the components referred as `Y`

This fitting method is achieved through the following process: The available discrete set of first order Thevenin parameter X measurement data of all cells are randomly shuffled and split into two datasets for *modeling* and *validation*. Afterwards, the fit is performed on the modeling set for each EEC parameter to obtain the interpolating polynomials $p_{n,X}(SoC)$ of degree n. Then, the quadratic error (residual)

$$e_q = \sum_{k,validation\_set} \left(p_{n,X}(SoC_k) - X_k\right)^2 \tag{3.28}$$

is calculated by comparing the aforementioned fit with the *validation* dataset. X denotes the respective EEC component, i.e. $X \in \{V_{oc}, R_0, C_1, R_1\}$. This is done for every polynomial degree in a range of $[n_{Min}, n_{Min}+1, \ldots, n_{Max}]$. Finally, the optimal polynomial degree is obtained by choosing the one with the minimal error e_q.

To avoid randomness obtaining the optimal degree is done multiple times (M runs). In this case, the optimal polynomial degree n_o is determined as the rounded integer mean

of the individual optimal degrees calculated across all runs. The improved method is summarized in the pythonic pseudo-code representation in Figure 3.8.

3.4 Implementation of the battery pack

In the previous section, the framework for parameterizing individual battery cells according to circuit-based approaches has been established. In many applications, a complete set of batteries is used to obtain sufficient power. This is especially the case for electric vehicles. For this reason, it is necessary to have a model for scaling from battery to pack level. The purpose of this section is the derivation of a method to do this. The main idea is to use the linearity of RC electrical circuits and differential equations used in the previous section to model batteries.

3.4.1 Building up the battery pack

To simulate the pack, one needs to obtain its equivalent circuit. This is composed of the single components X of the cells being contained in it. One can imagine it as depicted in Figure 3.9.

The main idea is to exploit the linearity of electric circuits and differential equations describing their behavior. By using this consideration, the circuit representation of the entire pack with an equivalent internal resistance, R_0^{pack}, RC-element R_1^{pack}, C_1^{pack} and voltage source V_{oc}^{pack} can be constructed. For their calculations, for the sake of simplicity, it is assumed, that the cells building the respective pack, have the same X-values. Removing this condition would lead to equations that are more algebraically complex, yet the same considerations would apply. How to obtain the pack EEC-components is derived in the following.

The ohmic resistance of the pack, R_0^{pack}: Having n_s resistances in series, the total resistance is obtained by adding up their values. Having n_p in parallel, the sum of their inverses equals the inverse of the total resistance. Via this consideration, R_0^{pack} is given by

$$\begin{aligned} R_0^{pack,ser} &= R_0^{cell} \cdot n_s \\ \frac{1}{R_0^{pack}} &= \sum_{i=1}^{n_p} \frac{1}{R_0^{pack,ser}} = \frac{n_p}{R_0^{pack,ser}} \\ \Longrightarrow R_0^{pack} &= \frac{n_s \cdot R_0^{cell}}{n_p} \end{aligned} \tag{3.29}$$

The upper index "*ser*" stands for the value of equivalent resistance, that would replace the n_s single resistances connected in series in *one row* of Figure 3.9.

The total resistance in the equivalent RC element R_1^{pack}: With the same consideration as above

$$\begin{aligned} R_1^{pack,ser} &= R_1^{cell} \cdot n_s \\ \frac{1}{R_1^{pack}} &= \sum_{i=1}^{n_p} \frac{1}{R_1^{pack,ser}} = \frac{n_p}{R_1^{pack,ser}} \\ \Longrightarrow R_1^{pack} &= \frac{n_s \cdot R_1^{cell}}{n_p} \end{aligned} \tag{3.30}$$

The capacitance of the pack, C_1^{pack}: Having n_p capacitances in parallel, the total capacitance is obtained by adding up their values. Having n_s in series, the sum of their inverses equals the inverse of the total capacitance. Via this consideration,

$$\begin{aligned} C_1^{pack,par} &= C_1^{cell} \cdot n_p \\ \frac{1}{C_1^{pack}} &= \sum_{i=1}^{n_s} \frac{1}{C_1^{pack,par}} = \frac{n_s}{C_1^{pack,par}} \\ \Longrightarrow C_1^{pack} &= \frac{n_p \cdot C_1^{cell}}{n_s} \end{aligned} \tag{3.31}$$

The upper index "*par*" stands for the value of equivalent capacitance that would replace the n_p single capacitances connected in parallel in *one column* of Figure 3.9.

The open-circuit voltage of the pack, V_{oc}^{pack}: Having n_s voltage sources (with V_{oc}^{cell} volts each) in series, they yield a voltage of

$$V_{oc}^{pack,par} = V_{oc}^{cell} \cdot n_s \tag{3.32}$$

It does not play a role how many of them are connected parallel.

The capacity of the pack, Cap_0^{pack}: Having n_p charge sources (witch Cap_0^{cell} Ampere-hours each) in parallel, they yield a capacity of

$$Cap_0^{pack} = Cap_0^{cell} \cdot n_p \tag{3.33}$$

It does not play a role how many of them are connected in serial.

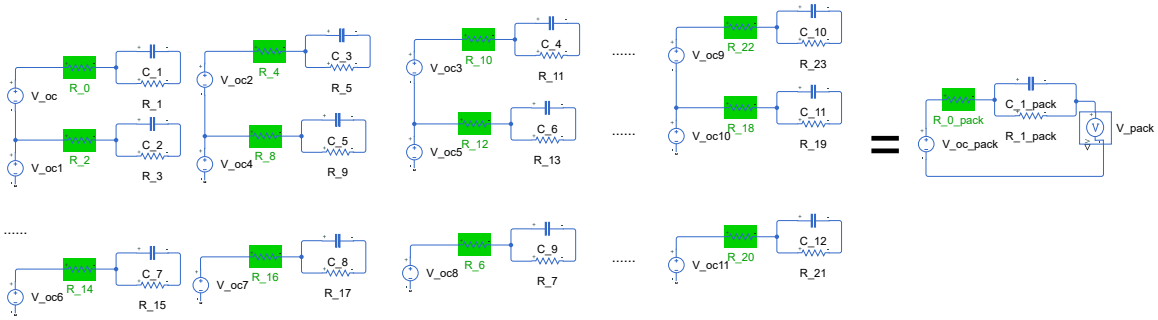


Figure 3.9: Consider the battery pack as one RC-Circuit, whose components are calculated from the single cells' passive components.

3.4.2 User-defined correction factors

With equations (3.29) - (3.33) one can scale up the the properties from cell to pack level. For the ideal case no deviations of pack voltage, capacity etc from the ones calculated with (3.29) - (3.33) could be observed. This is apparently not how reality works, see subsection 2.4.2, where the package voltage and capacity were not an integer multiple of the cell's voltage and capacity, as it would be ideally the case.

To capture the above described, for a single EV (or battery pack) specific behavior, the factors α_p, α_s are introduced:

$$\begin{aligned} Cap^{pack} &= n_p \cdot Cap^{cell} \cdot \alpha_p \\ V_{oc}^{pack} &= n_s \cdot V_{oc}^{cell} \cdot \alpha_s \end{aligned} \tag{3.34}$$

To have a value for α_p, α_s, one can consider the deviations between ideally upscaled and car-manufacturer-given parameters. For the example of Renault Zoe ZE50 R135 the LGES E79 cell was found to be used in the pack (see subsection 2.4.2). Below, the manufacturer data on packaging configuration is given. Manufacturer data of the identified LGES cell that is the building block of the pack is shown as well.

- Packaging: $n_s = 96, n_p = 2$ [75].
- Cell voltage: $V_{oc}^{cell} = 3.69\,\text{V}$ [77]. Car pack voltage: $V_{oc}^{pack} = 350\,\text{V}$ [75]
- Cell capacity: $Cap_0^{cell} = 78\,\text{Ah}$ [77].
- Car pack capacity: $Cap_0^{pack} = \frac{54.7\,\text{kWh}}{350\,\text{V}} = 156.286\,\text{Ah}$

One can calculate the correction factors by comparing the given pack values with the ones that would be observed when the "upscaling" would be ideal according to (3.29) - (3.33).

$$\begin{aligned} \alpha_s &= \frac{V_{oc}^{pack}}{n_s \cdot V_{oc}^{cell}} = 0.988 \\ \alpha_s &= \frac{Cap_0^{pack}}{n_p \cdot Cap_0^{cell}} = 1.0018 \end{aligned} \tag{3.35}$$

One could argue to introduce similar factors for the passive components values, i.e.

$$\begin{aligned} R_0^{pack} &= \frac{n_s \cdot R_0^{cell}}{n_p} \cdot \alpha_{R0} \\ R_1^{pack} &= \frac{n_s \cdot R_1^{cell}}{n_p} \cdot \alpha_{R1} \\ C_1^{pack} &= \frac{n_p \cdot C_1^{cell}}{n_s} \cdot \alpha_{C1} \end{aligned} \tag{3.36}$$

But for this, pulse or EIS-measurements should be conducted at least for one SoC value of the entire pack. Then insertion of the respective values at this considered SoC into equation (3.36) would deliver a way to mathematically capture the non-ideal scaling of the passive components. However, there where no resources available to obtain pack-level passive components values. For this reason, these correction factors are not considered, i.e are set to 1 in the following.

3.5 Simulation of battery (pack) voltage

After having obtained the SoC-dependency by the method described in 3.2 and 3.3, one has an analytical expression in form of an interpolating polynomial p_X for every $X \in \{V_{oc}, R_0, R_1, C_1\}$.

$$X(SoC) \approx p_X(SoC) \tag{3.37}$$

Under current flow, the State of Charge evolves with the time, therefore all passive components have the time as an implicit parameter.

$$X = X(SoC) = X(SoC(t)) \tag{3.38}$$

The above consideration neglects the degradation effects for the period of simulated battery operating conditions. If such effects would be taken account or play a role in the considered time-scale, the time t would be an explicit variable as well. This would mean the functional dependency of X would be given as $X = X(SoC(t), t)$.

The assumption in (3.38) can be supported by the following arguments:

- Duration of considered battery operation (up to several hours) is so small compared to time scales at which calendar aging (months, years) is relevant that it is negligible.
- Cyclic aging is negligible for the respective simulation because the number of simulated cycles is sufficiently small.
- Battery (pack) is not exposed to such extreme conditions so that degradation effects would become relevant within this short time and few cycles. This poses the condition on the simulation setup of not providing unrealistic high (more than several C) charging or discharging currents. In the considered use cases, this requirement is fulfilled.

In this work the battery (pack) voltage V_{pack} is simulated in discrete timesteps t_i. In the following, the abbreviations

$$\begin{aligned} X(SoC(t_i)) &:= X_i \\ I(t_i) &= I_i \\ SoC(t_i) &:= SoC_i \end{aligned} \tag{3.39}$$

will be used for the sake of brevity, $X \in \{V_{oc}, R_0, R_1, C_1, V_{bat}\}$. It does not matter, whether one considers a single battery cell, or a pack decomposed of $n_s \times n_p$ cells: In the latter case one considers the equivalent circuit representation of the pack. This would mean

$$\begin{aligned} V_{oc} &\mapsto V_{oc}^{pack} \\ R_0 &\mapsto R_0^{pack} \\ R_1 &\mapsto R_1^{pack} \\ C_1 &\mapsto C_1^{pack} \end{aligned} \tag{3.40}$$

The "*pack*" values are obtained through pre-processing the functional dependencies as described in 3.4.1 and 3.4.2. Therefore, in the following, the variables X^{pack} and X will be treated the same way.

After these considerations, in every time-step the battery (pack) voltage can be simulated. It is given by (3.1), what still needs to be solved in a discrete numerical way. When inserting discrete values instead of the continuous ones, minding the notation as defined in (3.39)

$$V_{bat,i} = V_{oc,i} - R_{0,i} \cdot I_i - v_{1,i} \tag{3.41}$$

Since v_1 is not given by an explicit analytic formula, but by a first order ordinary differential equation in time (see (3.1)), its calculation cannot be done by directly inserting the discrete values of X_i, I_i. It can be calculated by an explicit first-order time stepping method, that results in (3.2). It translates to

$$v_{1,i} = v_{1,i-1} \cdot \exp\left(\frac{-(t_i - t_{i-1})}{R_{1,i}C_{1,i}}\right) + \left(1 - \exp\left(\frac{-(t_i - t_{i-1})}{R_{1,i}C_{1,i}}\right)\right) \cdot I_{i-1}R_{1,i} \tag{3.42}$$

Inserting (3.42) into (3.41) yields an explicit formula for the battery (pack) voltage V_{bat} for every considered timestep t_i.The ingredients of (3.41), (3.42) are:

- State of Charge at timestep i: $SoC_i = SoC_{i-1} + (t_i - t_{i-1}) \cdot I_{i-1}$
- Based on SoC_i, the values of X_i
- The current from the previous timestep I_{i-1}
- Based on X_i, I_{i-1}, the voltage of RC-element $v_{1,i}$ can be calculated
- Out of the above quantities $V_{bat,i}$ can be obtained.

The initial parameters are necessary, since there is actually a differential equation of first order in time to be (numerically) solved. This means, the initial values $v_{1,0}, I_0, SoC_0$ are needed. If one considers the beginning of a complete (dis)charge process that starts after a sufficiently long time interval in which there were no battery currents, the initial parameters can be given as below:

$$\begin{gathered} I_0 = 0 \\ V_{bat,0} = V_{oc} \text{ (no current for sufficiently long time)} \\ \Longrightarrow v_{1,0} = V_{oc} - V_{bat,0} - R_{0,0} \cdot I_0 = 0 \end{gathered} \tag{3.43}$$

Nevertheless, a specification of other initial parameters is very well possible. That becomes relevant if one has multiple charging-algorithm stages, see next sub-sections.

In the subsequent section the implementation of most common charging algorithms are presented based on the established battery voltage simulation model.

3.6 Implementation of charging algorithms

Up to this point, the methods and differential equations to model the battery (pack) voltage have been established. As it is discussed in 2.4.3, this observable couples back on the value of the requested charging voltage and current. In the following a modular approach is presented, which separately implements:

1. Simulation of single battery cells.
2. Based on 1., the simulation of battery packs with corresponding up-scaling of single cells' parameters. This is derived based on simple electric-circuit based considerations.
3. Charging algorithms, which have as primary control parameter the output voltage of step 2. Additionally, the requested current and elapsed time.

The foundations of the first two points are described in 3.2 - 3.5, forming the basis for the third step. The latter will be described in the following subsections.

3.6.1 Implementation of CC and MCC charging

For a constant-current charging mode, one takes (3.41) and (3.42), and inserts for $I_i = I_s, \forall i \in \{1, \dots N_s\}$ in every constant-current stage s. By the index s the respective constant-current stage is denoted: $s \in \{1, \dots n_{stage}\}$, when having n_{stage} constant current stages in the respective charge protocol. For every stage s, there is a pre-defined, well known current I_s (that corresponds to some C-rate C_s). The number of stages n_{stage} is given at the beginning of the process as well. This gives an explicit formula for the evolution of the voltage profile for every stage s

$$\begin{aligned} V_{bat,i} &= V_{oc} - R_{0,i} \cdot I_s - v_{1,i} \\ t_i &\in T_{s-1}^{MCC}, T_s^{MCC} \end{aligned} \tag{3.44}$$

How the above formula can be evaluated is described in 3.5.

T_0^{MCC} is the time when MCC charging begins. The other T_s^{MCC} are time instants, when transition to the next, $s+1$th stage happens. $T_{n_{stage}}^{MCC}$ is the time instant that denotes the end of the MCC procedure.

Transition from one stage to another is coupled on the condition

$$V_{bat} > V_{thr} \tag{3.45}$$

where V_{thr} is a threshold voltage described in 2.3. Therefore, in every timestep $V_{bat,i}$ is simulated and the condition $V_{bat,i} > V_{th}$ has to be checked.

Another approach to set a transition criterion is to check the internal variable SoC instead of the battery voltage. Every stage goes until a defined State of Charge $SoC_{th,s}$. This means that the condition

$$SoC_{bat,i} > SoC_{th,s} \tag{3.46}$$

has to be checked.

Depending on the choice of criterion, if condition (3.45) or (3.45) is true, the current for the next step will be modified to the next-predefined value: $I_s \mapsto I_{s+1}$. This algorithm continues until

- The battery (pack) is charged "enough", i.e $SoC_i \geq SoC_{max}$ or
- Another algorithm (such as CV) begins. That is also coupled on reaching a certain State of Charge value that indicates the end of the MCC-phase. It is denoted by SoC_{MCC}. Therefore, the SoC-based end criterion for MCC charging translates to

$$SoC_i \geq SoC_{MCC} \tag{3.47}$$

- There is more time elapsed than the maximal, pre-set charging time. This time is denoted by T_{max}. Therefore, the time-based end criterion for MCC charging translates to

$$t_i > T_{max} \tag{3.48}$$

The initial condition for MCC stage can be either

1. As defined in (3.44) and by user-defined SoC_0 or
2. Inherited from the end of the previous algorithm, if there is any.

A brief explanation why both SoC and battery voltage based transition conditions are implemented:

In the literature, the theoretical development of charging algorithms is usually based on reaching a certain State of Charge. The reason is that this value is an actual, however directly non-measurable indicator of the battery's condition. In real BMS, what can be measured is the battery pack voltage V_{bat}. Out of this observable the BMS has algorithms to deduce the SoC. In this work both way of thinking is considered. Therefore, both trigger-point conditions are implemented in the simulation methods. Moreover, as discussed in 2.4.3, since this work develops a simulation framework, one does not have to bother about obtaining the internal, physically non-observable State of Charge by voltage measurement values.

3.6.2 Implementation of CV charging

After reaching a certain State of Charge (that is indicated by reaching a certain open-circuit voltage V_{oc}), charging switches to the constant voltage mode in order to protect the battery and avoid an overvoltage at the charge terminals. This means that the requested current has to fulfill the condition for the terminal voltage:

$$V_{bat}(t) = V_{oc}(t) - R_0(t) \cdot I - v_1(t) \stackrel{!}{=} const \tag{3.49}$$

This means that the first time derivative of the above expression is forced to be zero. In the following, the time derivative is denoted by a dot above the variable to be derived. By using the product rule

$$\frac{dV_{bat}(t)}{dt} = \dot{V}_{oc} - (\dot{R}_0 \cdot I + R_0 \cdot \dot{I}) - \dot{v}_1 \stackrel{!}{=} 0 \tag{3.50}$$

In the equation (3.50) the time derivatives of V_{oc}, R_0 and I can be replaced by using backward and forward differences. The value of $\dot{v}_1$ is given by equation (3.1).

$$\begin{aligned} \dot{V}_{oc} &= \frac{dV_{oc}}{dt} \approx \frac{V_{oc}(t) - V_{oc}(t-\Delta t)}{\Delta t} \\ \dot{R}_0 &= \frac{dR_0}{dt} \approx \frac{R_0(t) - R_0(t-\Delta t)}{\Delta t} \\ \dot{v}_1 &= \frac{dv_1}{dt} = -\frac{1}{\tau(t)} \cdot v_1(t) + \frac{I(t)}{C_1(t)} \\ \dot{I} &= \frac{dI}{dt} \approx \frac{I(t+\Delta t) - I(t)}{\Delta t} \end{aligned} \tag{3.51}$$

If one restricts the time interval in that (3.49) is to be solved on the duration of charging, the time-dependency of the EEC-components V_{oc}, R_0, R_1, C_1 is only governed by the time evolution of the State of Charge. This means that the following effects can be neglected:

- Calendar aging: This is justified by the fact that the duration of fast charging takes maximal several hours. This is much smaller than the time scale (years, months) where calendar aging becomes relevant.
- Degradation effects while operation: it is assumed that the battery pack is not exposed to such extreme conditions that degrading effects would become relevant while charging. This consideration is again justified, since charging protocols and algorithms are designed in a way to protect batteries from extreme degradation. This specially holds for the considered CV-mode.

The above considerations imply that the differential coefficients from (3.51) can be calculated as

$$\begin{aligned} \frac{V_{oc}(t) - V_{oc}(t-\Delta t)}{\Delta t} &= \frac{V_{oc}(SoC) - V_{oc}(SoC^-)}{\Delta t} := \Delta V_{oc} \\ \frac{R_0(t) - R_0(t-\Delta t)}{\Delta t} &= \frac{R_0(SoC) - R_0(SoC^-)}{\Delta t} := \Delta R_0 \\ \dot{v}_1 = \frac{dv_1}{dt} &= \frac{-v_1(t)}{\tau(SoC)} + \frac{I(t)}{C_1(SoC)} \end{aligned} \tag{3.52}$$

where SoC^- is given as

$$SoC^- := SoC(t-\Delta t) \approx SoC(t) - \Delta t \cdot I(t) \tag{3.53}$$

Now, one can insert the expressions from (3.51), (3.52) into equation (3.50), yielding the condition for the required current in the next time-step in order to "hold the battery voltage". This required current shall be denoted by $I(t+\Delta t) := I^+$.

$$\Delta V_{oc} - \Delta R_0 \cdot I - R_0 \cdot \frac{I^+ - I}{\Delta t} - \left(-\frac{1}{\tau(t)} \cdot v_1(t) + \frac{1}{C_1} \cdot I \right) = 0 \tag{3.54}$$

The calculation of v_1 is according to (3.2).

From equation (3.54) one can explicitly calculate the requested charging current for the next time-step, I^+:

$$I(t+\Delta t) := I^+ = \frac{\Delta V_{oc} + 1/\tau \cdot v_1 + (\Delta R_0 + R_0/\Delta t) \cdot I}{\Delta R + R_0/\Delta t + 1/C_1} \tag{3.55}$$

End criterion of the CV-mode is indicated by either

- Reaching the point where the battery is charged "enough", i.e. $SoC_i \geq SoC_{max}$
- Or the charging current drops below the pre-set, threshold value denoted by I_{cutoff}. Therefore, the current-based end criterion for CV charging translates to

$$I_i \leq I_{cutoff} \tag{3.56}$$

By using these considerations, the pseudo-code in Figure 3.10 for the simulation approach for CV-mode can be developed.

```
Initialize:
    I = I_init; Inext = I_init;
    SoC = SoC_init; Vocprev = Voc(SoC_init);
    R0prev = R0(SoC_init); v1prev = v1_init;
    R0 = R1 = Voc = -1;

    while(I < Icutoff):
        SoC += (I * dt)/Cap;
        C1 = C1(SoC); R1 = R1(SoC); R0 = R0(SoC); Voc = #update parameters
        T1 = R1 * C1; #time-constant tau
        dVoc = (Voc - Vocprev)/dt;
        dR0 = (R0 - R0prev)/dt;

        v1 = v1(v1prev, I, SoC) #calculate v1 with timestep method

        Inext = ( dVoc + v1/T1 + (dR0 + R0/dt)*I )/( dR + R0/dt + 1/C1 )

        I, Vocprev, R0prev, v1prev = [Inext, Voc, R0, v1] #for next iteration
```

Figure 3.10: Pseudo-code for the implementation of the CV-phase. Abortion is executed when the charging current goes below some user-defined cutoff value `Icutoff`. `SoCinit, v1_init` are the values of SoC and v_1 at the timestep when switching to CV mode. `Cap` is the capacity of the considered battery (pack).

3.6.3 Implementation of boost charging

The idea of this protocol is to add an initial CC or CV phase with a high voltage/current, V_{boost}, I_{boost} respectively. This "boosting interval" shall reduce the overall charging time, with minimized degradation of the battery, see 2.3 and [106]. Since there is no common approach in the literature which version is to be applied, and how long or until which SoC one needs to boost charge, this is up to the user to choose. After the boost-period, MCC-CV or CC-CV period follows to finish charging. There are two simulation options implemented for boost charge: either in CV or in CC mode.

CV - defined boost charging

The implementation follows the idea for CV-defined boost charging by Notten et al [106].

The idea is the following: start the boost charging by jumping to the boost battery voltage. This means, apply a current so that

$$V_{bat} = V_{boost} \tag{3.57}$$

and hold this (in general very high) voltage until the SoC reaches some cutoff value. This means one can actually use the CV algorithm presented in 3.6.2. The only hurdle is to "find" the starting current of this process, I_{BS}. This can be done by solving an equation with respect to the necessary current for reaching the boost voltage in the first timestep of BC phase.

The aforementioned equation for I_{BS} is given by taking (3.44) and insert I_{BS} in place of the charging current:

$$V_{bat} = V_{oc}(SoC_0) - R_0(SoC_0) \cdot I_{BS} - v_{1,0} \stackrel{!}{=} V_{boost} \tag{3.58}$$

Here $SoC_0, v_{1,0}$ denote the State of Charge and transient voltage when entering the boost charge phase. V_{boost} is given according to the requirement of the user/charging protocol specification. The only unknown in the above equation is I_{BS}, one can solve for it, yielding the expression

$$I_{BS} = \frac{V_{oc}(SoC_0) - v_{1,0} - V_{boost}}{R_0(SoC_0)} \tag{3.59}$$

After having jumped to $V_{bat} = V_{boost}$ the condition

$$V_{bat} = V_{boost} \stackrel{!}{=} const \tag{3.60}$$

needs to be hold. This leads again to the differential equation as in (3.50), with the initial condition $I_0 = I_{BS}$. The required current values to keep the battery voltage at V_{boost} in the subsequent time steps can be calculated by the method described in 3.6.2.

This constant-voltage boost charge algorithm runs until

- The battery (pack) is charged "enough", i.e $SoC_i \geq SoC_{max}$ or

- Another algorithm (such as MCC) begins. That is also coupled on reaching a certain State of Charge value that indicates the end of the CV boost-phase. It is denoted by $SoC_{Boost,CV}$. Therefore, the SoC-based end criterion for CV-boost charging translates to
$$SoC_i \geq SoC_{Boost,CV} \tag{3.61}$$
This is the most praxis relevant case.

- There is more time elapsed than the maximal, pre-set charging time. This time is denoted by T_{max}. Therefore, the time-based end criterion for CV boost charging translates to
$$t_i > T_{max} \tag{3.62}$$

CC -defined boost charging

In this case, the battery (pack) is charged with the high, pre-defined constant boost-charge current I_b. This leads to the implementation algorithm of the CC protocol, described in 3.6.1 using the time stepping equation (3.44) with $n_{stage} = 1$, $I = I_b$.

This constant-current boost charge algorithm runs until

- The battery (pack) is charged "enough", i.e $SoC_i \geq SoC_{max}$ or

- Another algorithm (such as MCC) begins. That is also coupled on reaching a certain State of Charge value that indicates the end of the CC boost-phase. It is denoted by $SoC_{Boost,CC}$. Therefore, the SoC-based end criterion for CC-boost charging translates to
$$SoC_i \geq SoC_{Boost,CC} \tag{3.63}$$
This is the most praxis relevant case.

- There is more time elapsed than the maximal, pre-set charging time. This time is denoted by T_{max}. Therefore, the time-based end criterion for CC boost charging translates to
$$t_i > T_{max} \tag{3.64}$$

3.6.4 Arbitrary current profile

Until this point, a model that can be scaled for an arbitrary number and configuration of cells has been developed . After obtaining the electric circuit-based representation of the respective battery (pack) and the SoC-dependencies of its components, with (3.41) one is able to calculate its voltage for every timestep if the current values I_i are available. This means, the developed model does not only allow to simulate common charging protocols: Since the necessary equations and their solution method is set, it is possible to describe the evolution of the pack's voltage with an arbitrary load-current. This can be either charging (negative) or discharging (positive). The discretized calculation of battery voltage requires

- Prescribed, required current profile

- Initial state of charge

- Initial transient-voltage value v_{10}.

Then, applying (3.41) makes it possible to simulate the voltage profile according to the respective current.

3.7 Post-processing of the Results

Physical considerations and an understanding of the EEC model's parameters do not only allow to simulate the battery voltage up-scaled to an arbitrarily built-up pack, but also to give an estimation about

1. The output or charging power being delivered by or into the battery.
2. The amount of energy and charge that has been charged into the battery over time.
3. Also, in case of discharging, the amount of energy and charge that has been extracted from the battery over time.
4. Based on the previous two points, the charge- or (for real-life scenarios more relevant case) energy efficiency when cycling under a certain load profile.
5. Power loss at any time instant while (dis)charging process.
6. Based on (5.) the energy loss at a certain time-step of the discharging process.

In the following, the calculation method and exact definition of the mentioned quantities will be explained.

1-3. (Dis-)charging power and energy

First it is to be exactly defined what is meant by these terms. It is a well known phenomenon that not all the stored charge or energy in a battery can be re-extracted and its amount is dependent on the load current profile, i.e the C-rate. A relevant use-case scenario is considering the voltage at the battery terminal (this is governed by (3.1)) and the prescribed current profile.

The charging power naturally arises from the formula

$$P_{charge} = I_{charge} \cdot V_{charge} \tag{3.65}$$

I_{charge} is a given parameter for every timestep (see 3.6). The necessary charging voltage, V_{charge} is ideally the battery voltage at this time-step: That is, the minimum thermodynamic-kinetic-given necessary voltage for the Li-intercalation. In reality, the actual applied voltage is by some factor ξ higher than this ideal value:

$$V_{charge}(t) = V_{bat}\Big(SoC(t), I_{charge}(t)\Big) \cdot \Big(1 + \xi\Big) \tag{3.66}$$

the method for calculating $V_{bat}\Big(SoC(t), I_{charge}(t)\Big)$ is described in 3.2 and 3.6. In this work, the value of ξ is chosen to be 2 %. Therefore, the exact definition of the charging power in this work

$$P_{charge}(t) = I_{charge}(t) \cdot V_{bat}\Big(SoC(t), I_{charge}(t)\Big) \cdot \Big(1 + 0.02\Big) \tag{3.67}$$

The discharging power at a certain time instant is the product of the prescribed current I_{disch} and the deliverable voltage (V_{bat}) at this moment:

$$P_{disch}(t) = I_{disch}(t) \cdot V_{bat}\Big(SoC(t), I_{charge}(t)\Big) \tag{3.68}$$

When having the (dis)charged power profile over time, one can calculate the (dis)charged amount of energy by integrating over the power profile:

$$\begin{aligned} E_{charge}(t) &= \int_{t_0}^{t} P_{charge}(\tau)\,\mathrm{d}\,\tau \\ E_{disch}(t) &= \int_{t_0}^{t} P_{disch}(\tau)\,\mathrm{d}\,\tau \end{aligned} \tag{3.69}$$

the expressions for $P_{charge},\ P_{disch}$ are given in (3.67) and (3.68).

4. Cycling - Efficiency

In analyzing battery usage, one may seek to quantify the total energy extracted or available under specific conditions after a certain amount of charge input under some circumstances. This can be expressed by determining the energy extracted from a battery following a charge input under specified operating conditions. Therefore the cycling efficiency η_{cyc} is defined as follows:

$$\eta_{cyc} := \frac{E_{disch}}{E_{charge}} \tag{3.70}$$

As it can be seen by considering (3.67), (3.68) and (3.69), η_{ch} is dependent on the prescribed current profile because E_{charge} and E_{disch} do so. This value holds particular significance for complete cycles, where the process initiates and concludes at identical State of Charge levels.

5.-6 Power and energy loss

These are the most ambiguous terms. Losses could of course include any contributions being independent of the battery pack itself: coming from finite conductivity of surrounding cables, heating or cooling power (preconditioning before starting operating the battery), for EVs charging loss due to the on-board charger etc. It might be beneficial to be able to get a measure for these battery-independent losses separately.

The "loss" arising from the battery is given due to the fact, that it can deliver a voltage which deviates from the theoretically deliverable voltage at a certain State of Charge. This theoretical voltage V_{th} is given by the laws of thermodynamics (see 2.1). Applying the assumption (2.8) that the open-circuit voltage V_{oc} corresponds to the theoretical voltage with a good approximation, the battery-loss in terms of deliverable voltage can be summarized as

$$V_{bat}(SoC(t), I(t)) \neq V_{th}(SoC(t)) \approx V_{oc}(t) \tag{3.71}$$

In this work only the losses related to the battery are defined and calculated. Therefore the charging losses with (3.67) are

$$\begin{aligned} P_{loss,ch}(t) &= P_{ideal,ch} - P_{charge} \\ &\approx \left| V_{oc}\big(SoC(t)\big) \cdot I_{charge}(t) - V_{bat}\big(SoC(t), I_{charge}(t)\big) \cdot I_{charge}(t) \right| \\ &= \left| \Big(V_{oc}(SoC(t)) - V_{bat}\big(SoC(t), I_{charge}(t)\big) \Big) \cdot I_{charge}(t) \right| \end{aligned} \tag{3.72}$$

and the discharging-related loss is

$$\begin{aligned} P_{loss,disch}(t) &= P_{ideal,disch} - P_{disch} \\ &\approx \left| V_{oc}\big(SoC(t)\big) \cdot I_{disch}(t) - V_{bat}\big(SoC(t), I_{disch}(t)\big) \cdot I_{disch}(t) \right| \\ &= \left| \Big(V_{oc}(SoC(t)) - V_{bat}\big(SoC(t), I_{disch}(t)\big) \Big) \cdot I_{disch}(t) \right| \end{aligned} \tag{3.73}$$

The energy loss is given by integrating both the charging and discharging related loss-power

$$E_{loss} = \int_{t_0}^{t} P_{loss,ch}(\tau)\,\mathrm{d}\,\tau + \int_{t_0}^{t} P_{loss,disch}(\tau)\,\mathrm{d}\,\tau \tag{3.74}$$

Application

For the simulation case always discrete time-steps are considered. Therefore, in all the above equations the time-stepping methods derived in the previous subsections have to be applied in order to evaluate the respective expressions. This means,

$$\begin{gathered} SoC(t),\ V_{bat}\big(SoC(t), I_{\mathrm{disch}}(t)\big),\ I(t) \mapsto SoC(t_k), V_{bat}\big(SoC(t_k), I_{\mathrm{disch}}(t_k), I(t_k)\big) \\ P(t) \mapsto P(t_k) \\ E(t) = \int_{t_0}^{t} P(\tau)\,d\tau \mapsto E_k = \sum_{\kappa<k} P(t_\kappa)\Delta t_\kappa \end{gathered} \tag{3.75}$$

3.8 Software implementation

In the previous sections, both the theory and equations for simulating charging algorithms and battery voltages based on given load currents have been developed. In this section, the software implementation of these algorithms is presented. A starting point is to parameterize the respective cells.

The structure of the Python-language based software is depicted in Figure 3.11. The principle is ***SISSiP***: Save-Import-Scale-Simulate-PostProcess, in this order. All of these parts are evaluated in the following sub-sections.

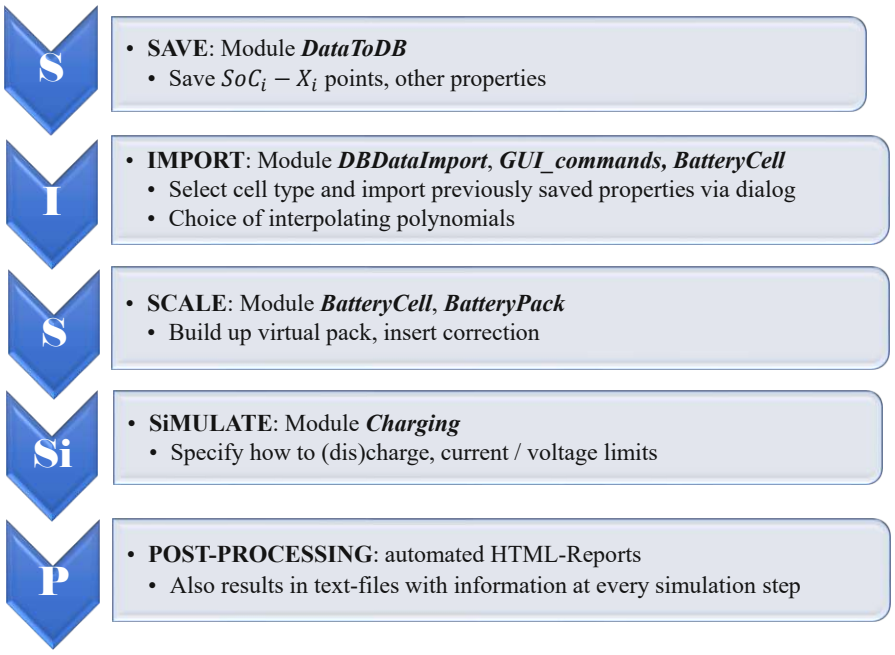


Figure 3.11: Main points of the software implementation. Also names of the responsible modules are shown.

The software provides a graphical user-interface and user dialogs to specify simulation parameters. In some cases, to allow a quicker start for practiced users, specifications can be also given in .txt and .json files. The order of questions popping up by the program is shown in Figure 3.12.

After having answered all these prompts, the program creates an instant of `class Charging`. It contains the implementations of all the discussed charging algorithms and the necessary parameters for their calculations, EVSE-limitations (supported current and voltage range), simulation step-size and a few I/O functions.

The `class Charging` has a nested instant of `class BatteryPack`. This is the virtual battery pack whose behavior is to be simulated under the specified conditions. It has the attributes specifying the pack configuration (`nser, npar`), and also the α-correction factors (`alpha_ns, alpha_np`) discussed in 3.4. It contains the pack-capacity and EEC-parameters as well. They are calculated from the underlying cell's qualities as described in 3.4. The properties of the aforementioned cell are put into the `class BatteryCell`.

The individual battery cell, represented by `class BatteryCell`, stands as a distinct entity. It establishes a "has-a" relationship with `class BatteryPack`. `BatteryCell` contains attributes specifying its EEC-components: `*Data` are the discrete X-parameter points and `*Coeff` are the coefficients of the interpolating polynomials that are calculated from them as described in 3.3.2. Also information about the temperature(s) where `*Data` was recorded and other additional information (name, weight, chemistry, capacity) is put into this class.

The described class-structure and hierarchy is shown in Figure 3.13.

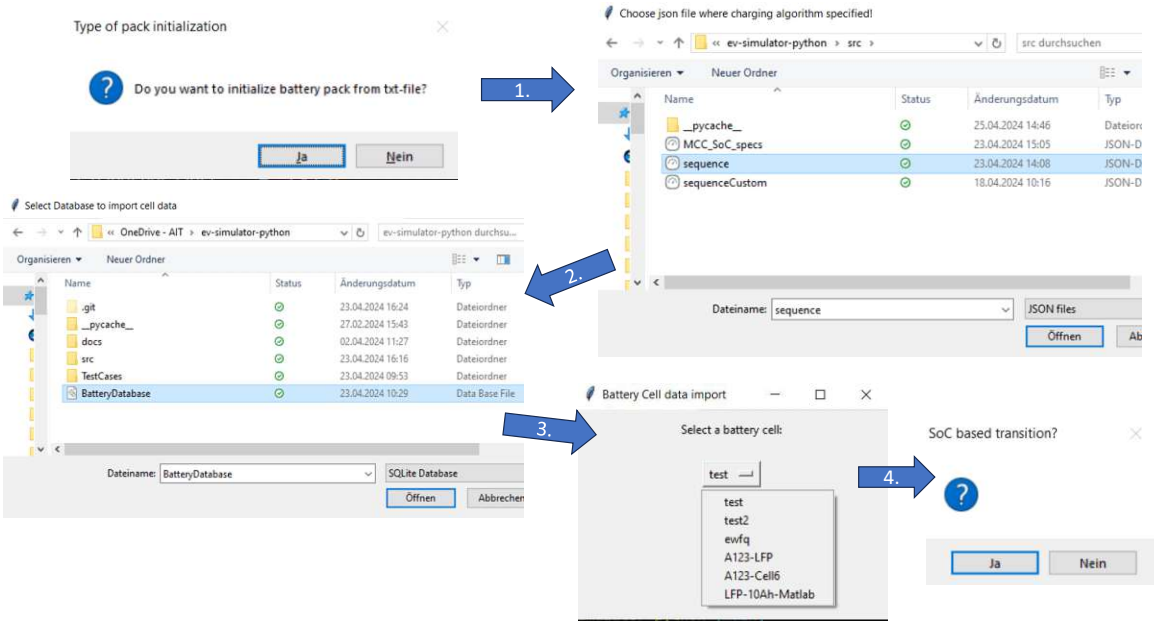


Figure 3.12: User dialog steps of the software.

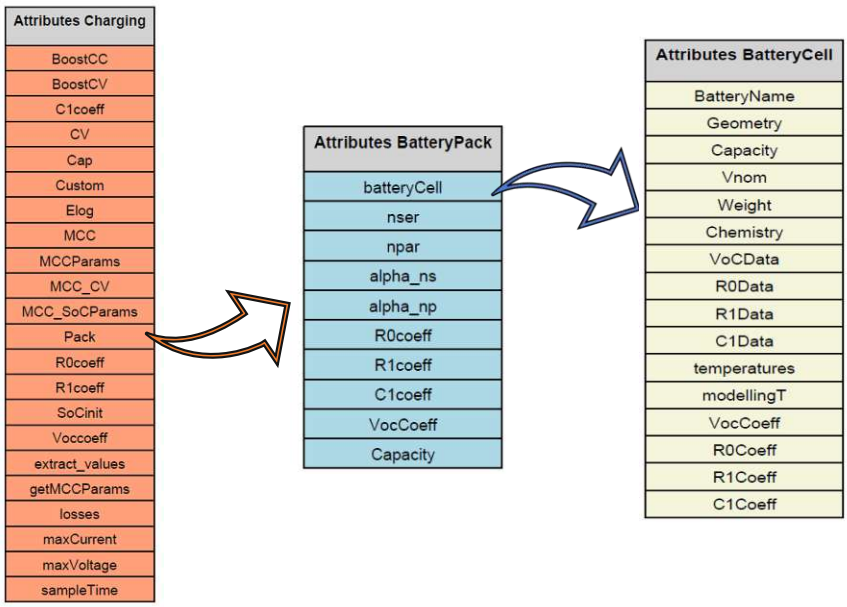


Figure 3.13: Hierarchy and attributes of the three basic Python classes.

3.8.1 SAVE cell to database

For every component X of the first order Thevenin model a discrete set of $SoC_i - X_i$ points and the corresponding temperature, at which it was recorded is needed. The rules for saving them in a previously, by the author implemented SQLite database are:

1. For the respective component X, the number of corresponding SoC points has to be the same as the discrete X-points. This allows to do polynomial interpolation when the cell is later chosen to run a simulation with it, see 3.3.

2. At the beginning of the X-array, the recorded ambient temperature during testing has to be provided. Meaning: it is possible to conduct the pulse discharge measurements with the same current profile at different temperatures T_j, $j \in [1, \dots J]$ if having done the measurements at J different temperatures. Having N pulses in the pulse discharge procedure means that an $(N+1) \times J$-array for the respective X value is saved. The array's first row is the vector of the recorded temperatures, at which the measurements were done.

3. Every cell needs to have a unique name, under which it is saved. It is the so-called "key" in the respective database.

4. The EEC parameters are in SI-units, capacity has to be given in Ampere-hours, weight in grams.

5. The input data for SoC, X arrays is in .dat-format, separated by semicolons ";"

This procedure needs to be done once per cell, whose data is then saved in a SQLite database. For the simulation, properties will be automatically loaded from there after having selected the respective battery cell by its name. The tool is designed on purpose in a way that for one cell under the ID "Name" for all of its components a different set of $SoC - X$ pairs can be chosen. This allows to examine the behavior of imaginary batteries: For example, what if battery B, that has been optimized in terms of its internal resistance R_0, would be also optimized in terms of its open-circuit-voltage characteristics?

3.8.2 IMPORT cell properties from database and SCALE up pack

This is the starting point of the subsequent charging-simulation. In this step, the user is asked via graphical user interface (GUI) dialogs to choose from the batteries whose properties and discrete points of $SoC-X$ measurements are stored in a database. X denotes the respective component from the cell's electric equivalent circuit, i.e $X \in \{V_{oc}, R_0, R_1, C_1\}$. This happens the following way:

1. Choose via drop-down menu the cell type by name, out of the stored ones

2. Choose the ambient-temperature, at which one wants to run the simulation. The user can provide an arbitrary value. Temperatures, at which $R_0 - SoC$ values are recorded, are displayed. They are chosen because the ohmic resistance's temperature dependence is the most crucial. Always the dataset, whose respective temperature is the closest to the user-provided one, is chosen. Usually, the $SoC - X$ data points that are extracted from the method described in subsection 3.2.2, have been recorded at the same set of temperatures for a certain battery cell.

3. Choose the degree of interpolating polynomial for V_{oc}, R_0, R_1, C_1

4. Choose the number of cells in parallel (n_p) and in series (n_s) in the pack

5. Choose the factors α_s, α_p that stand for the non-idealistic behavior due to packaging

All of these points can be done either per user-dialog or per dedicated text files. The rules are

- If choosing the option "No" at the question in Figure 3.12 step 1, the user will be guided by subsequent dialog windows to provide cell and battery pack specification data.

- If choosing "Yes" the above parameters will be read-in from the file *valueSpecs.txt*.

After making these steps, first the functional $SoC - X$ dependency via polynomial fitting as described in section 3.3 is obtained, for the respective battery *cell*. With these steps, in the Python simulation code, the `class BatteryPack` is built up out of the `class BatteryCell`. To do so, the attributes `BatteryPack.alpha_np`, `BatteryPack.alpha_ns` (correspond to α_p, α_s from equation 3.34) are used based on the method described in 3.4.1. The cell type from the database is always chosen by an user-dialog .

3.8.3 SiMULATE (dis)charging modes

In the software the charging modes that are discussed in 3.6 are implemented. Their software implementation follows exactly the logic in 3.6.

Implementation of MCC charging

This mode is implemented as the attribute `MCC` of `class Charging`. In this case, the user can define

1. The number of stages

2. The C-rates at the certain stages

3. The transition (cutoff) voltage V_{cutoff} when the algorithm moves to the next charging current step (if "SoC based transition?" from Figure 3.12 was answered with "No") or the State of Charge values when the algorithm moves to the next charging current step ("SoC based transition?" from Figure 3.12 was answered with "Yes")

4. State of Charge value when algorithm moves to the next charging mode

Then, the simulation iterates as long as the pack's SoC reached the user defined transition value. The calculations in the iteration steps are done as described in 3.6.1.

Implementation of Boost charging

In this case, the user can define

1. The boost current I_b (for `BoostCC`) or boost voltage (for `BoostCV`) at this stage
2. The transition State of Charge values when the algorithm moves to the next charging step

If choosing the option voltage-based boost charging the following has to be done: In the first iteration step I_{BS} in order to jump to the specified boost voltage has to be calculated. Its calculation is done as given by (3.59). Subsequently, the CV-algorithm developed in 3.6.2 can be run until reaching the transition condition. If $V_{boost} >$ `maxVoltage`, the boost voltage is replaced by the maximum voltage the EVSE can deliver (`maxVoltage` in Figure 3.13).

In case of current-based boost charging, the boost-charging current I_b can be prescribed. The battery is charged with this current until the maximum deliverable voltage by EVSE, or a pre-defined SoC value is reached. In this case, the MCC-algorithm protocol is run with 1 stages and one current value I_b. If the specified value I_b or I_{BS} exceed the maximum deliverable current (`maxCurrent`), they are set to its value.

Implementation of CV charging

When switching to this mode, the goal is to hold the battery voltage, at which this mode is entered. This is done by requesting in every timestep iteratively the next current as described in 3.6.2. The simulation runs as long as either

- Battery is charged enough or
- The current drops below some threshold (specified by user) or
- Maximal, set simulation time is exceeded

It is implemented as the attribute `CV` of `class Charging`.

Implementation of MCC-CV charging

This is a combination of an MCC and a CV charging mode: first calling `Charging.MCC` then subsequently `Charging.CV`. The initial parameters of `Charging.CV` are inherited from the end of `Charging.MCC`.

Implementation of simulating battery pack behavior under arbitrary load current

This mode is implemented as the attribute `Custom` of `class Charging`. In this case, the user can choose the current profile from a csv-file. The latter has to contain the array of time instants t_k and the array for (dis)charging current values I_k at the given time instants. The program reads in the arrays and iterates over them, solving in every iteration the equations for the battery current (see 3.6). It is ensured that neither too

much current can be loaded into the pack nor too much can be extracted from it.

For this purpose the ideally strict condition

$$SoC \leq 1 \wedge SoC \geq 0 \tag{3.76}$$

is relaxed to

$$SoC \leq 1.05 \wedge SoC \geq -0.05 \tag{3.77}$$

The condition described in (3.77) has the purpose to consider the fact that actual capacity is C-rate dependent. Since this tool is intended to be applied to some arbitrary C-rate(s), it makes sense to not strictly stick to the ideal condition in (3.76).

3.8.4 POST-PROCESSING implementation

While running the simulation (see previous subsection) for every timestep, several properties are calculated and can be exported into txt-files. In the automatically generated *"demo.txt"* the following columns can be found:

- `Time`: The real time at which the simulation step was run. It is being implemented with the intention of integrating the software into an electric vehicle emulator at a later stage.
- `Elapsed[s]`: Elapsed simulated time from a user-defined zero-point. By default the beginning of the simulation process denotes `Elapsed[s] = 0`.
- `BatteryPack Voltage[V]`: It is V_{bat} at the time-step `Elapsed[s]`.
- `SoC`: State of Charge of the battery pack at `Elapsed[s]`..
- `Charging current [A]`: Current flowing into (by the program's convention then positive) or being extracted from the battery (by the program's convention then negative) at time-step `Elapsed[s]`.
- `Charged [Ah]`: Integrated current, by the Coulomb counting method.
- `Power[kW]`: Power being delivered by or into the battery. It is calculated by (3.67) in charging mode (`ChargingCurrent > 0`) and by (3.68) in discharging mode (`ChargingCurrent < 0`).

In the file *energyLog.txt* the (dis)charged amount of energy at time interval `dt` is calculated. `dt` is given as the difference between two subsequent timesteps:

```
dt[i] = Elapsed[s][i+1] - Elapsed[s][i]
```
(3.78)

- If `ChargingCurrent > 0` then the energy-entry will have a positive sign and it equals

```
dE = Power[kW] * dt
```
(3.79)

at timestep `Elapsed[s]`.

- If `ChargingCurrent < 0` then the entry will have a a negative sign and it equals

$$\texttt{dE = Power[kW] * dt} \tag{3.80}$$

  at timestep `Elapsed[s]`, same as in the charging case. The difference is that `dE` has a negative sign in this case, because `Power[kW]` does so.

This allows to separately consider and calculate the

- Charged amount of energy during the process. It is called `Echarge` and calculates as:

$$\texttt{Echarge = sum(energyLog>0)} \tag{3.81}$$

- Discharged amount of energy during the process. It is called `Edischarge` and given by:

$$\texttt{Edischarge = sum(abs(energyLog<0))} \tag{3.82}$$

Then the cycling efficiency in case of full cycles, `eta_cyc` as defined in 3.7 is easily calculated by

$$\texttt{eta\_cyc =Edischarge / Echarge} \tag{3.83}$$

Also losses in terms of energy can be calculated by the software. For this purpose, the file *logLosses.txt* contains the loss-power at every time-step `Elapsed[s]` according to its definition in 3.7. At the end of the simulation the arrays `timevector` (created from `Elapsed[s]`) and `P_loss` are also available. Having these two entities allows to calculate the losses in terms of energy (`ElossCharge, ElossDisch`) as described in 3.7.

Translated to the software case, it can be implemented as follows:
Denoting the set of indices of the time-instants in charging mode charged (i.e `ChargingCurrent>0`) with `charging`, and the set of indices of the time-instants where in discharge mode (i.e `ChargingCurrent<0`) with `discharge` respectively it holds

$$\texttt{ElossCharge = sum(P\_loss[charged] * timevector[charged])} \tag{3.84}$$

and

$$\texttt{ElossDisch = sum(P\_loss[discharge] * timevector[discharge])} \tag{3.85}$$

According to (3.74) the "lost" energy `Eloss` during the procedure is

$$\texttt{ElossCharge + ElossDisch} \tag{3.86}$$

At the end of the simulation, the text files are loaded into a jupyter-notebook, and the described calculations are performed. Also plots are generated and the report is automatically exported as an HTML file. After opening the default-generated jupyter-notebook, the simulation data is already loaded into `numpy` and `pandas` arrays. Therefore plots can be easily modified by conventional Python commands, and other desired aspects of the simulation can be highlighted or extracted. A default exported report (compressed) is shown in Figure 3.14.

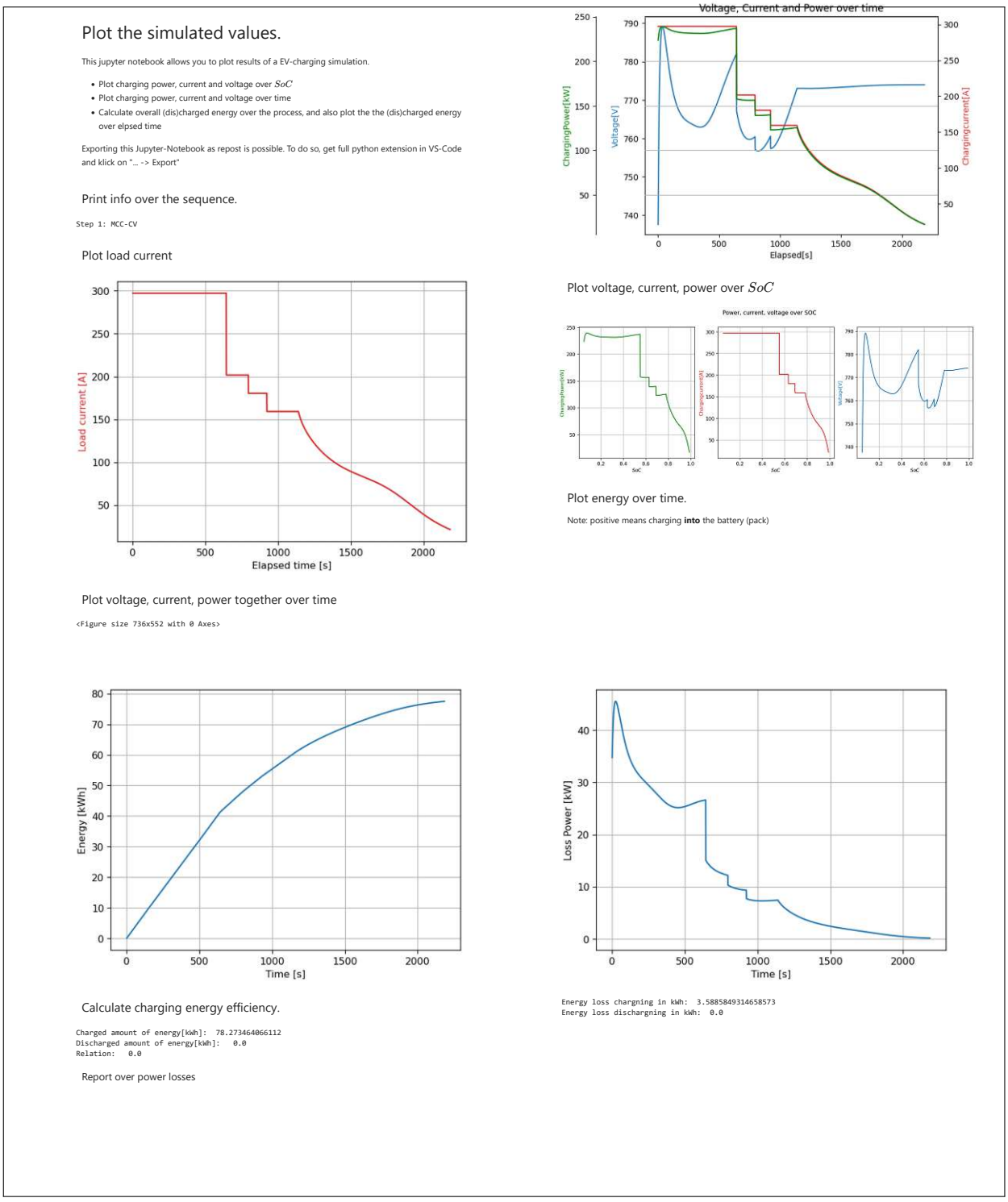


Figure 3.14: Default format of HTML-report.

4 Results

4.1 Validation of the battery model

In order to experimentally validate the adaptive polynomial fit and numerical battery-voltage simulation method (framework that is developed in Chapter 3) 10 samples of type 18650 $LiFePO_4$ batteries underwent the presented parametrization workflow. These tests were conducted in the *Battery Test Laboratory* of Austrian Institute of Technology GmbH.

In Figure 4.1 the obtained functional SoC-dependencies based on adaptive polynomial fitting of all of the 10 cells are depicted. It seems that variations and oscillations are more pronounced when the State of Charge approaches either 1 or 0. What is notable is that deviations are particularly pronounced for low or high SoC values. In this regime the deviations between the overall fit and single samples behavior will gain therefore more significance compared to the middle State of Charge area. But still, as discussed it 3.3 it is numerically still "the best we can do". This statement is proven in the next steps. To do so, the recorded voltage profile is re-simulated in three ways (Approach 1-3). Subsequently they are analysed in terms of their accuracy. For this purpose *one, arbitrary* sample's voltage profile of the set is targeted. In the presented particular case it is cell #2.

Approach 1 The functional State of Charge-X dependency obtained by *adaptive polynomial fitting* was taken to re-simulate the voltage profile during the pulse discharge measurement. In particular, simulation result are obtained by the software whose implementation is the subject of 3.8.

- For $SoC - X$ dependencies, the interpolation polynomials resulting from the adaptive method described in 3.3.2 were taken.
- For numerical battery simulation the equations and time-stepping methods as described in 3.2, 3.6 were used.
- A battery pack with $n_p = n_s = 1$ and $\alpha_p = \alpha_s = 1.0$ was used.
- The time resolution was 1 s

The simulation results compared with the recorded voltage of the chosen cell #2 are shown in Figure 4.2.

Approach 2 For this case the $SoC - X$ functional dependency is obtained by "classically" doing a polynomial fit of high ($n = 15$) polynomial degree on *solely one* cell's data. For the considered case it was cell #6. Therefore it is aimed to represent the effect of not considering fluctuations in single samples properties what would be the case for adaptive polynomial fitting: Functional dependency is obtained by "fine-tuning" the interpolating

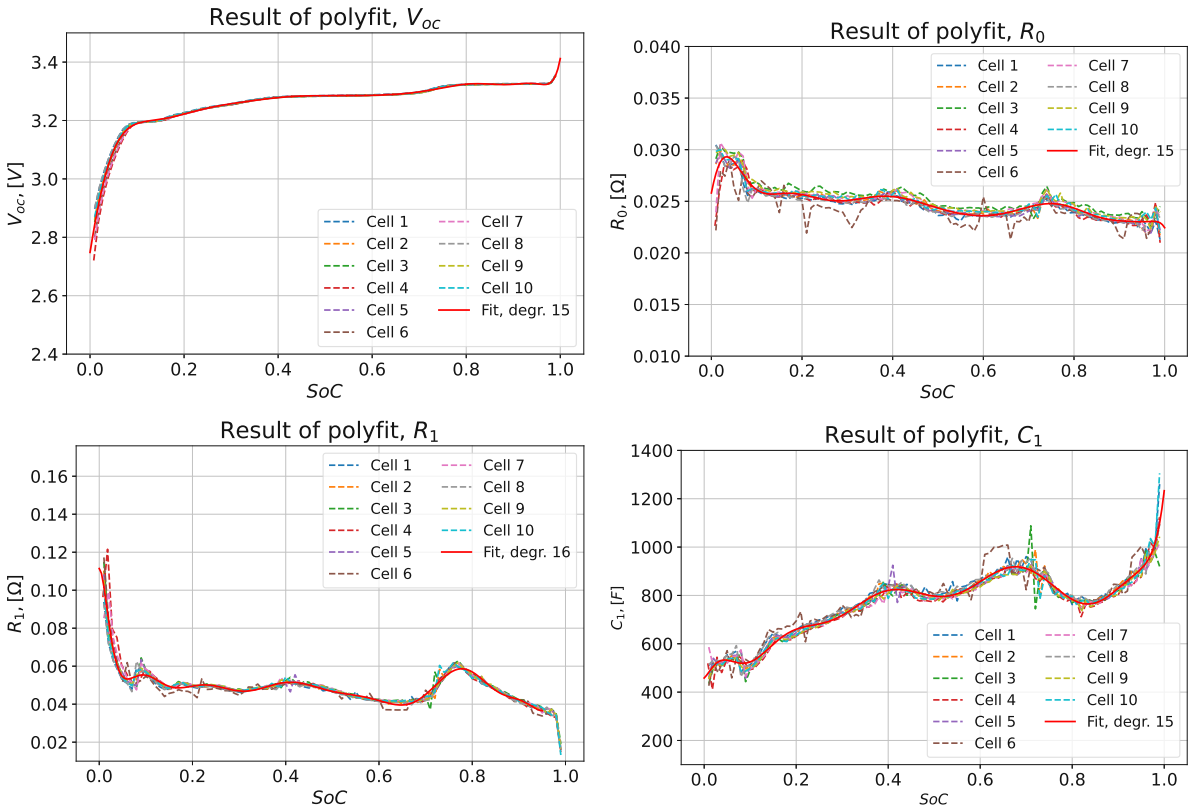


Figure 4.1: Functional dependencies of EEC parameter values on the SoC. The optimal polynomial degree determined by the algorithm is depicted in the legends.

functions on one cell's properties. This would of course yield accurate results for this respective cell, but performs (more) poorly when applying the model to another sample (cell #2) belonging to the same system.

The comparison with **Approach 1** (adaptive fit) is shown in Figure 4.3 and in Figure 4.4. The comparison demonstrates the rationale behind adaptively fitting to the system properties that are "hidden" within all samples of a specific model. Despite using the same polynomial degree and model, focusing on only one sample (here #6) gives less accurate results.

To optimally illustrate the deviations, a time-frame of 1000 s in Figure 4.4 is depicted, where the effects of different modeling data and method can be clearly seen.

Approach 3 This step should present the effect of using smooth interpolating functions (polynomials) instead of lookup tables. Lookup tables are used as state of the art in battery modeling for approximating a mathematical function such as the SoC dependency of model parameters. This is the method which is used in the commercially available popular tool for battery modeling: MATLAB's battery framework. It provides a simulation model for voltage profiles based on lookup-tables [104], [107]. It implements the already presented RC-circuit battery model by creating lookup tables which are functions of SoC. To acquire the latter one, the block *Estimation Equivalent Circuit Battery*[107] integrates (dis)charge currents. Generating lookup tables is based on tabulated $SoC - X$ data of a *specific* cell.

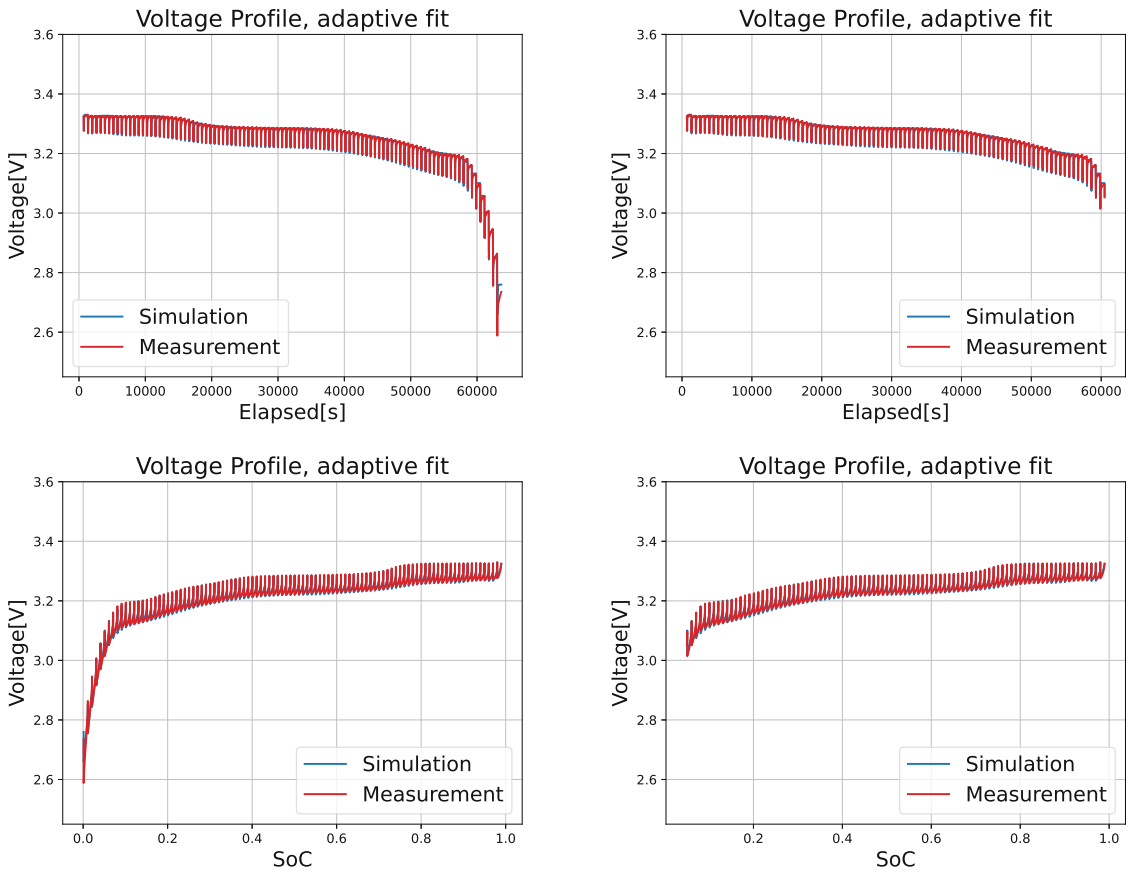


Figure 4.2: Re-simulating the voltage profile of cell #2. $SoC - X$ dependencies are obtained by the adaptive polynomial fitting method described in 3.3.2. Here in the second row the voltage evolution over the State of Charge (that implicitly contains time) is plotted.

Since MATLAB is a renowned state of the art in software-based reference modeling, it is taken as another baseline. For this purpose, especially the auto-block *Estimation Equivalent Circuit Battery*[107] from the *Powertrain Blockset* was used. For time-dependent simulations this tool has already implemented an fourth-order ordinary differential equation solver [108]. It is based on the fourth-order Dormand-Prince method whose foundations are the explicit Runge-Kutta numerical integration formulae [109]. This is even a more sophisticated algorithm than the simpler backward and forward first-order time stepping methods that are used in this work. To give MATLAB a "head start" in terms of comparing the modeling approaches, the *Estimation Equivalent Circuit Battery* is allowed to take **exactly** the data points of the considered cell #2 for re-simulation purposes. The results of MATLAB-simulations are depicted in Figure 4.5.

As stated earlier, a comparison of **Approach 3** and **Approach 1** have the purpose to present the supremacy of smooth higher order interpolation functions over solely C^0 continuous lookup tables. The reader may recall the convergence behavior in (3.23).

The errors for all the three methods

- Proposed adaptive parameter fitting combined with physical-numerical models in this work, $SoC \in [0.0, 0.99]$ *(A)*

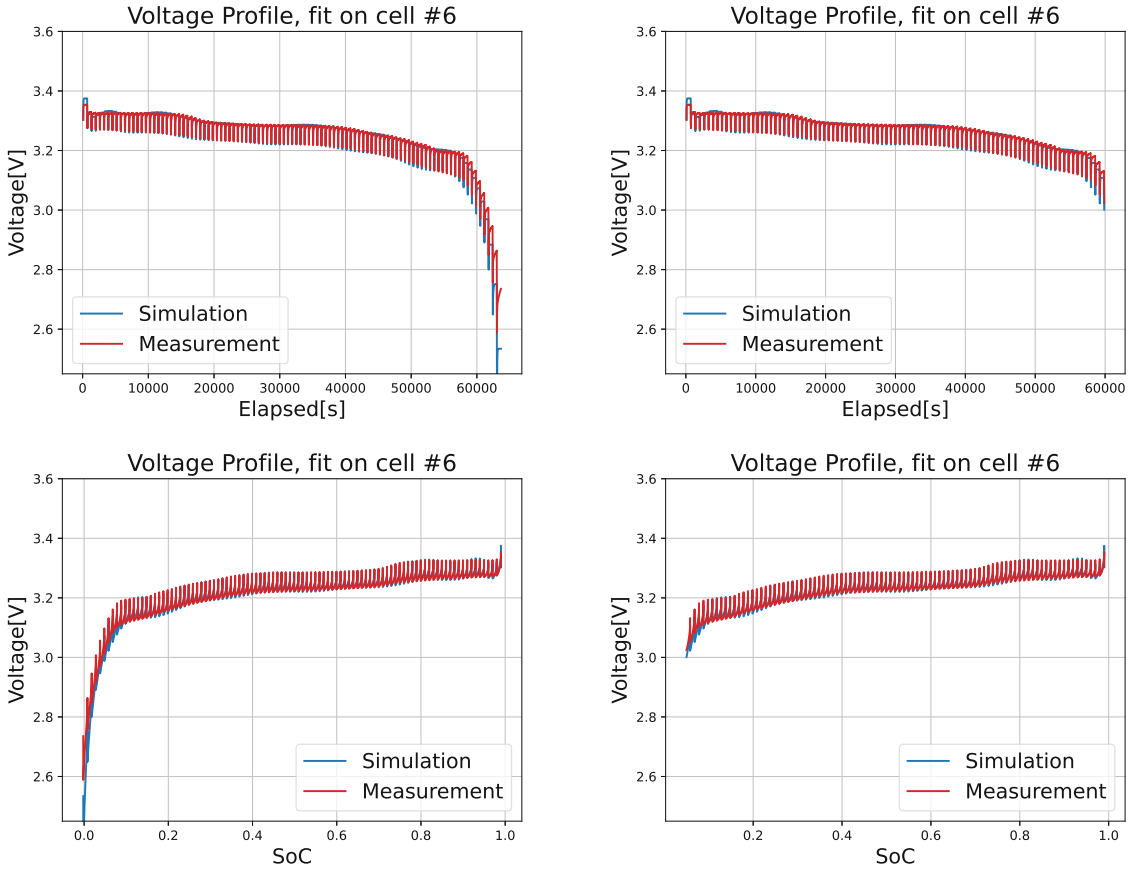


Figure 4.3: Re-simulating the voltage profile of cell #2. $SoC - X$ dependencies are obtained by concentrating only on properties of cell #6. Here in the second row the voltage evolution over the State of Charge (that implicitly contains time) is plotted.

- Proposed adaptive parameter fitting combined with physical-numerical models in this work, $SoC \in [0.05, 0.99]$ *(B)*
- Conventional polynomial parameter fitting combined with physical-numerical models in this work $SoC \in [0.0, 0.99]$ *(C)*
- Lookup table (MATLAB software) $SoC \in [0.0, 0.99]$ *(D)*

are compared in Table 4.1. The quantity *Mean Error %* is the *Mean error [V]* relative to the "plateau" of open-circuit voltage ($\approx 3.3\,\mathrm{V}$). To visualize the differences, it is referred to the bar-plot in Figure 4.6.

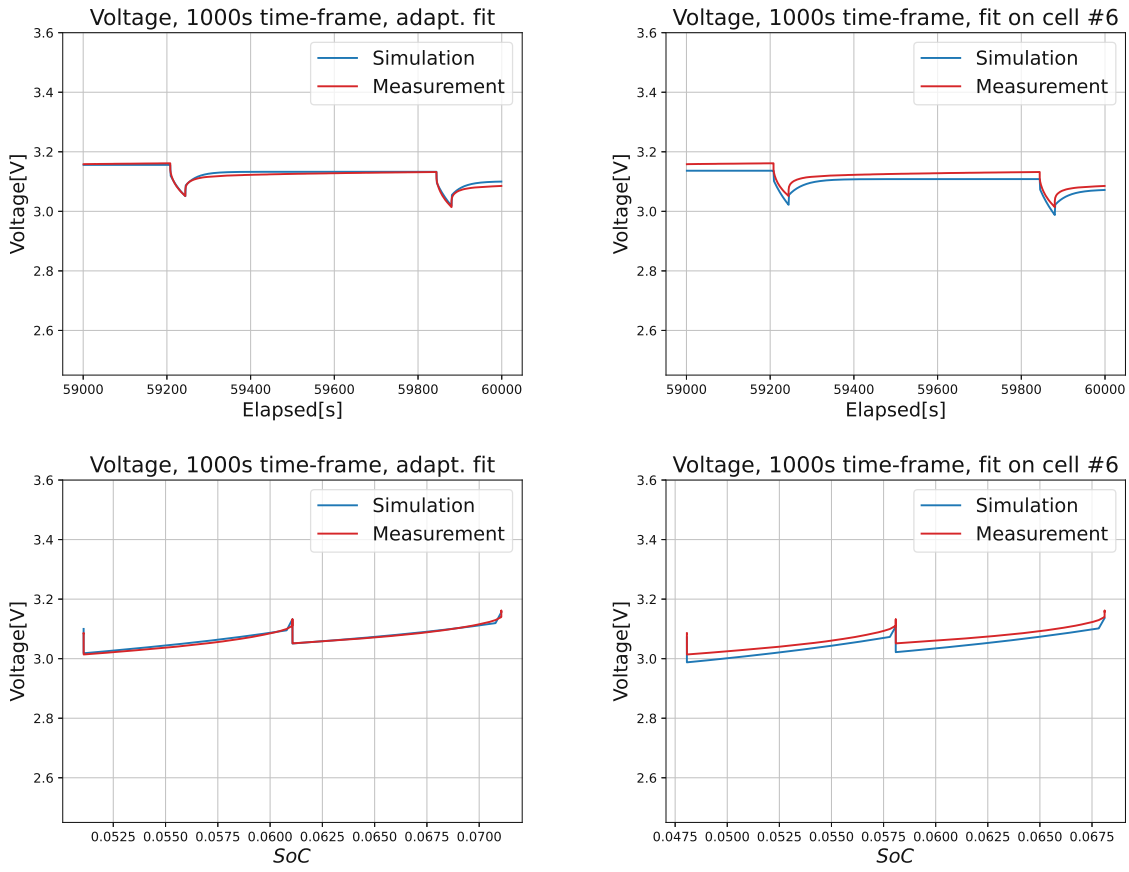


Figure 4.4: Comparison of a 1000 s timeframe of the simulation and experiment. In the left column, $SoC - X$ dependencies are obtained by the proposed adaptive polynomial fitting method. In the right column, $SoC - X$ dependencies are obtained by fitting on data of cell #6, with same degree as for the adaptive method.

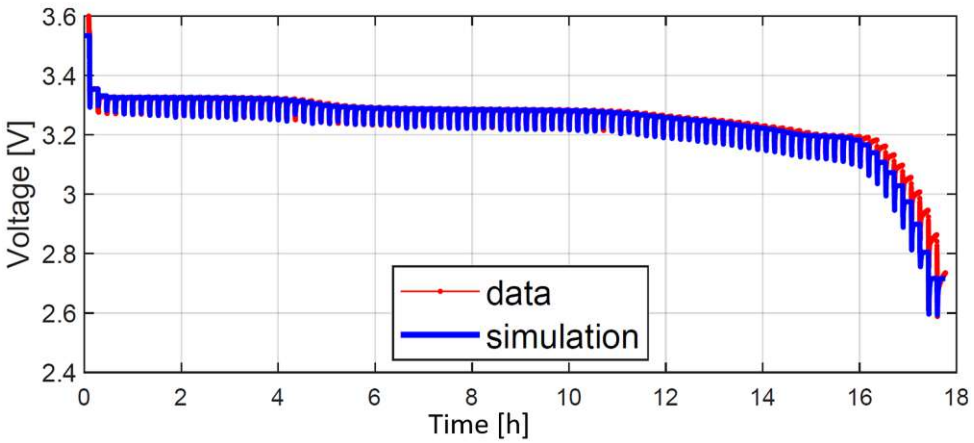


Figure 4.5: Recorded voltage, current and time profile for the pulse discharge measurement. The plotted values belong to cell #2. Simulation was done by considering only the X_i values of cell #2, using higher-order explicit Dormand-Price formula embedded into MATLAB's *Estimation Equivalent Circuit Battery*.

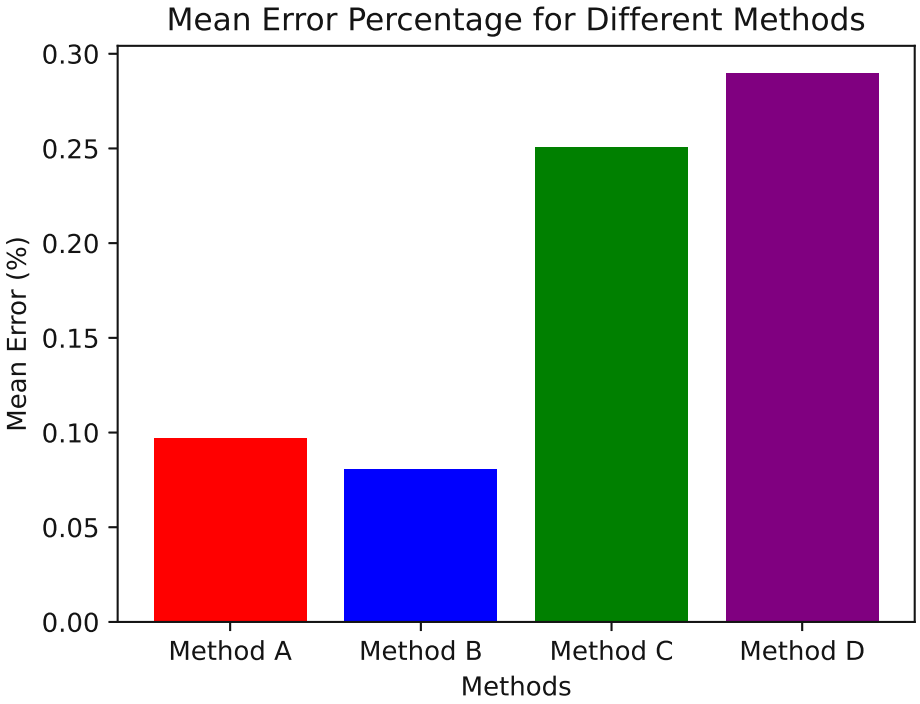


Figure 4.6: The mean error of compared methods

	Method A	Method B	Method C	Method D
Mean error [V]	0.00321	0.00266	0.00827	0.00956
Max error [V]	0.07226	0.02992	0.30594	0.17765
Mean Error %	0.097%	0.0806%	0.2506%	0.2897%

Table 4.1: Comparison of errors between different methods.

4.2 Validation of the EV-charging simulation framework

This section is dedicated to validate and present the function of SoC-based simulation using the developed software. After successfully validating the battery model which is the basis of the framework, simulated charging curves with publicly available ones are compared. It is targeted to reproduce the charging curve of the EV *Hyundai IONIQ 6* [110]. As baseline the publicly available charging curve of the respective car from Fastned [111] is taken.

There is no accessible exact data of the battery used in the EV (also nothing about the chemistry). Therefore the only currently available high power-battery, whose pulse discharge measurement data are provided by MATLAB is taken as building block of the underlying EV battery pack. It is stated to be a type of "lithium-ion polymer" (LiPo) type with a capacity of 10.006 Ah [112]. There is only data for one single cell of this type available. Therefore it is not possible to use the method of adaptive polynomial fitting on multiple sample's data, whose efficiency has been proven in the previous section.

For the aforementioned LiPo cell the $SoC_i - X_i$ values are obtained with MATLAB's PulseSequence object out of the provided pulse-discharge profile. For a quality check, the pulse discharge profile was re-simulated with the built-in method of the MATLAB PulsSequence object, called *plotSimulationResults*. The profile is shown in Figure 4.7.

In the following a method how to scale up the virtual battery pack for the subsequent charging-simulation is derived. By doing so, accurate results can be reproduced. Solely publicly available data of the target electric vehicle Hyundai IONIQ6 are taken as simulation input.

Reproducing such curves, where only some data plotted over SoC are available is very difficult based on the voltage-based transition criterion. The situation is of course different in case a voltage profile is available.

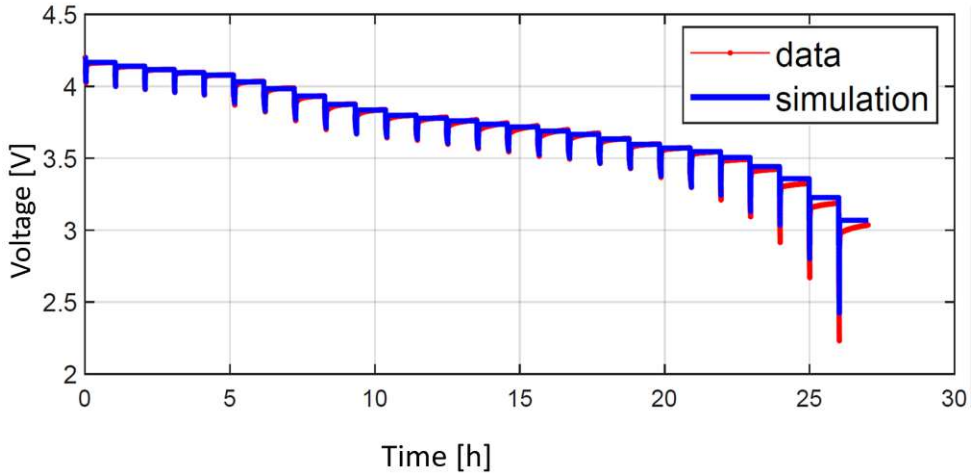


Figure 4.7: Pulse Discharge measurement profile of the 10 Ah LFP battery from MATLAB.

The single steps, how to simulate an real-scenario EV charging procedure with the developed software and framework are listed below. They are followed and accompanied by

comparisons of publicly available parameters of the respective car and charging procedure with the simulation results.

1. Manufacturer Data of the EV

The manufacturer data is taken based on the information of the EVDB website [110].

- Pack Configuration: $n_s = 192$, $n_p = 2$
- Nominal voltage of the pack $V_p = 697\,\mathrm{V}$
- Usable battery capacity (energy) $E_p = 74.0\,\mathrm{kWh}$

2. Available Cell type

The only data which could be extracted for the respective cell type was based on its provided pulse discharge measurement data.

- Capacity cell : $Cap_c = 10.006\,\mathrm{Ah}$
- Nominal cell voltage: $V_c \approx 3.8\,\mathrm{V}$ (taken from the plateau of the V_{oc}-curve)

3. Get $n_s, n_p, \alpha_s, \alpha_p$

They can be calculated from above data. How to do so, is described below:

- The number of cells of the parametrized type, required to be ordered in series in the considered virtually-built pack n_s, and the correction factor α_s are given by the condition that the virtual pack needs to deliver the same nominal voltage as the EV.

$$\begin{aligned} n_s &= \mathrm{int}\left(\frac{V_p}{V_c}\right) = 183 \\ \alpha_s &= \frac{V_p}{n_s \cdot V_c} = 1.0023 \end{aligned} \tag{4.1}$$

- The number of cells of the parametrized type, required to be ordered in parallel in the considered virtually-built pack n_p, and the correction factor α_p are given by the condition that the virtual pack needs to have the same nominal capacity as the EV. We denote the charge-related capacity of the EV by Cap_p, and calculate it as the usable battery capacity (energy) divided by the nominal voltage of the pack.

$$\begin{aligned} Cap_p &\approx \frac{E_p}{V_p} = 106.169\,\mathrm{Ah} \\ n_p &= \mathrm{int}\left(\frac{Cap_p}{Cap_c}\right) = 10 \\ \alpha_p &= \frac{Cap_p}{n_p \cdot Cap_c} = 1.061 \end{aligned} \tag{4.2}$$

4. Design the charging parameters

For this purpose the Fastned-charging-curve (see Figure 4.11) can be analyzed. On the curve 4 CC-Phases can be observed, followed by a CV phase. From the provided curve simulation input values can be estimated. It is assumed that the EVSE is available to deliver the necessary voltage (i.e assume an $800V$ -architecture). What is further provided by Fastned, is that the maximal charging power accounts to 233 kW. This is observed in Phase 1.

4.1 Phase 1: Here the maximal charging power $P_1 = 233\,\mathrm{kW}$ can be observed. If the assumption that the battery is at its nominal voltage is taken, then the charging current and thereby the respective C-rate can be calculated in the first approximation.

$$\begin{aligned} I_1 &= \frac{233\,\mathrm{kW}}{697\,\mathrm{V}} \approx 334.28\,\mathrm{A} \\ \Longrightarrow C_1 &= \frac{I_1}{Cap_p} = 3.147\,\mathrm{h}^{-1} \end{aligned} \tag{4.3}$$

This first CC-phase goes until $SoC_1 \approx 0.55$

4.2 Phase 2: here a charging power $P_2 \approx 175\,\mathrm{kW}$ can be observed. If the assumption that the battery is at its nominal voltage is taken, then the charging current and thereby the respective C-rate can be calculated in the first approximation:

$$\begin{aligned} I_2 &= \frac{175\,\mathrm{kW}}{697\,\mathrm{V}} \approx 251.1\,\mathrm{A} \\ \Longrightarrow C_2 &= \frac{I_2}{Cap_p} = 2.365\,\mathrm{h}^{-1} \end{aligned} \tag{4.4}$$

This second CC-phase goes until $SoC_2 \approx 0.63$

4.3 Phase 3: here a charging power $P_3 \approx 160\,\mathrm{kW}$ can be observed. If the assumption that the battery is at its nominal voltage is taken, then the charging current and thereby the respective C-rate can be calculated in the first approximation.

$$\begin{aligned} I_3 &= \frac{160\,\mathrm{kW}}{697\,\mathrm{V}} \approx 229.56\,\mathrm{A} \\ \Longrightarrow C_3 &= \frac{I_3}{Cap_p} = 2.162\,\mathrm{h}^{-1} \end{aligned} \tag{4.5}$$

This third CC-phase goes until $SoC_3 \approx 0.69$

4.4 Phase 4: here a charging power $P_4 \approx 152\,\mathrm{kW}$can be observed.If the assumption that the battery is at its nominal voltage is taken, then the charging current and thereby the respective C-rate can be calculated in the first approximation.

$$\begin{aligned} I_4 &= \frac{152\,\mathrm{kW}}{697\,\mathrm{V}} \approx 218.01\,\mathrm{A} \\ \Longrightarrow C_4 &= \frac{I_4}{Cap_p} = 2.05\,\mathrm{h}^{-1} \end{aligned} \tag{4.6}$$

This fourth CC-phase goes until $SoC_4 \approx 0.78$

After taking these above calculated parameters for the MCC-CV algorithm, it can observed that the charging power is too high compared to the one depicted on Fastned. It is shown in Figure 4.8. The depicted Fastned-charging procedure was done until $SoC \approx 0.96$, the simulated charging procedure was done for the sake of completeness until $SoC = 0.99$. It is however only plotted until $SoC < 0.96$.

This first approach is already not a bad approximation, but can be still improved as it is shown in the following. In order to obtain more accurate results, one needs to "fine-tune" the C-rates, i.e. make a second, corrected simulation. A refined approximation for the C-rates can be applied by comparing the simulated and given charging power at **Phase 1**.

$$\begin{aligned} \frac{P_1(\text{Simulated})}{P_1(\text{Given})} &\approx \frac{262\,\text{kW}}{232\,\text{kW}} = 1.123 \\ \Longrightarrow C_1^{new} &\approx C_1/1.123 \approx 2.8 \end{aligned} \tag{4.7}$$

This means that the applied C-rates by comparing the deviation of simulated power from the "target value" can be obtained. The corrected C-rates are rounded down. It is done so because it has to be considered that when applying higher C-rates the charging voltage is higher too. The reason is that there is more loss voltage on the R-components of the EEC model as indicated in (3.1).

Taking the correction factor derived in (4.7) the new C-rates for the second try can be obtained:

$$\begin{aligned} C_1^{\text{new}} &= \frac{C_1}{1.123} \approx 2.8 \\ C_2^{\text{new}} &= \frac{C_2}{1.123} \approx 1.9 \\ C_3^{\text{new}} &= \frac{C_3}{1.123} \approx 1.7 \\ C_4^{\text{new}} &= \frac{C_4}{1.123} \approx 1.5 \\ (C_4^{\text{new}} &\text{adjusted to be a little bit lower}) \end{aligned} \tag{4.8}$$

After doing this adjustment, a second simulation can be run, all the other parameters remain the same. That yields the result depicted in Figures 4.9, 4.10 and 4.11. On the simulation output data post-processing results can be done as well, as described in 3.7 and 3.8. The post-processing results can be an indicator of the accuracy of the simulation: If they are consistent with the available manufacturer specifications, it suggests that the simulation is producing accurate results. This was observed to be the case:

- Charged amount of energy: 78.27 kWh
- Energy loss while charging 3.6 kWh
- Therefore one can calculate the amount of charged energy, stored in the battery: $78.27\,\text{kWh} - 3.6\,\text{kWh} = 74.67\,\text{kWh} \approx$ usable battery capacity according to manufacturer.

– According to EVDB, the usable battery-pack capacity of the electric vehicle Hyundai IONIQ 6 accounts 74.0 kWh [110].

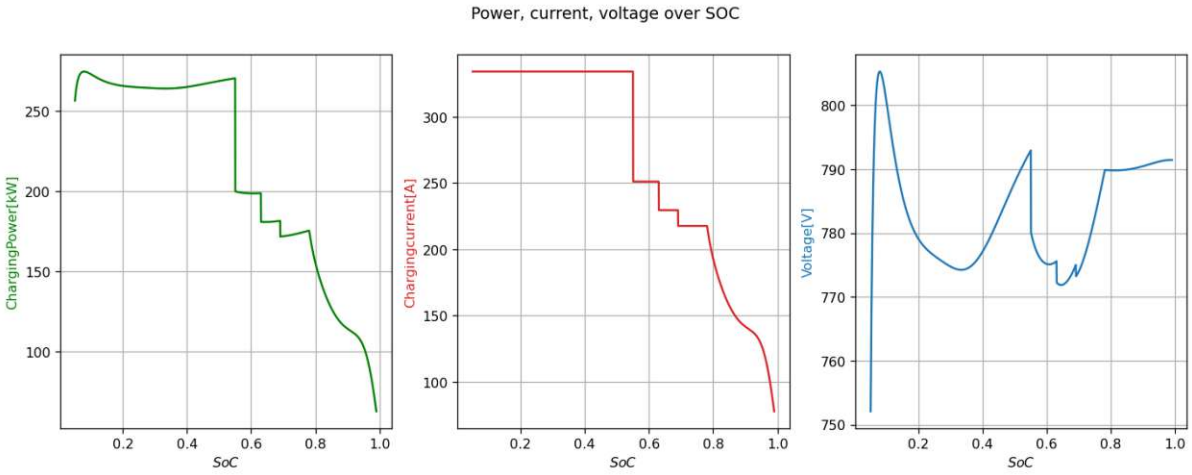


Figure 4.8: First try with calculated initial, approximate parameters. This way, the charging power is "overshoot".

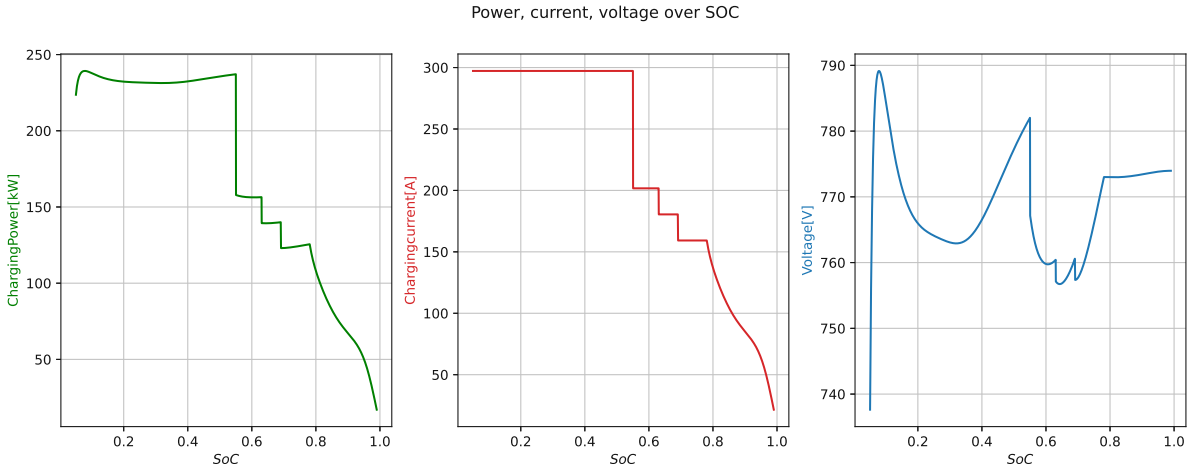


Figure 4.9: Second try with the modified initial parameters. This way, the charging power is reproduced correctly. The charging power is again re-plotted and compared against the reference curve from Fastned in Figure 4.11.

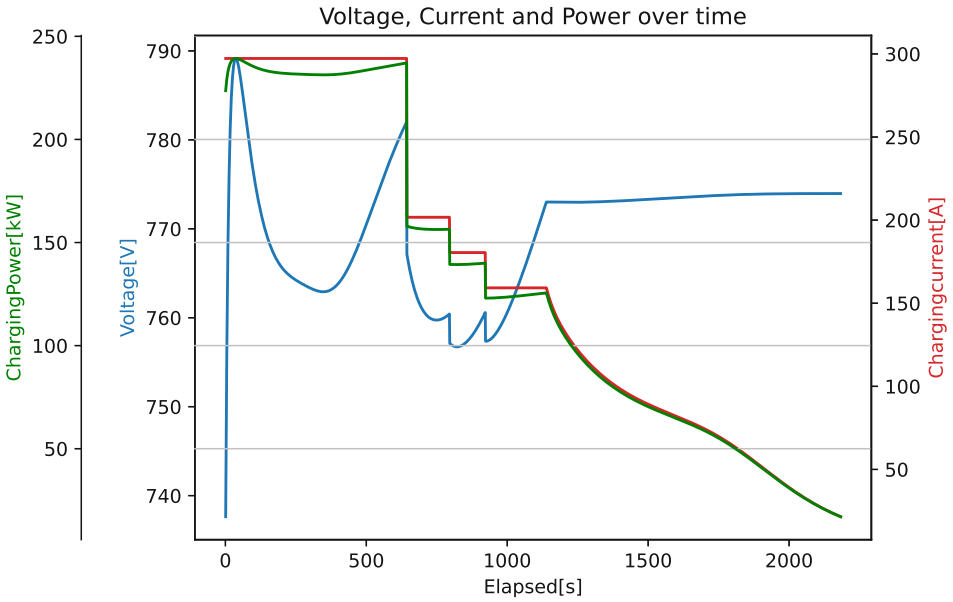


Figure 4.10: Second try with the modified initial parameters. This way, the charging power is reproduced correctly. Note that on this figure, the values are plotted over the time. Mind the difference compared to plotting over SoC.

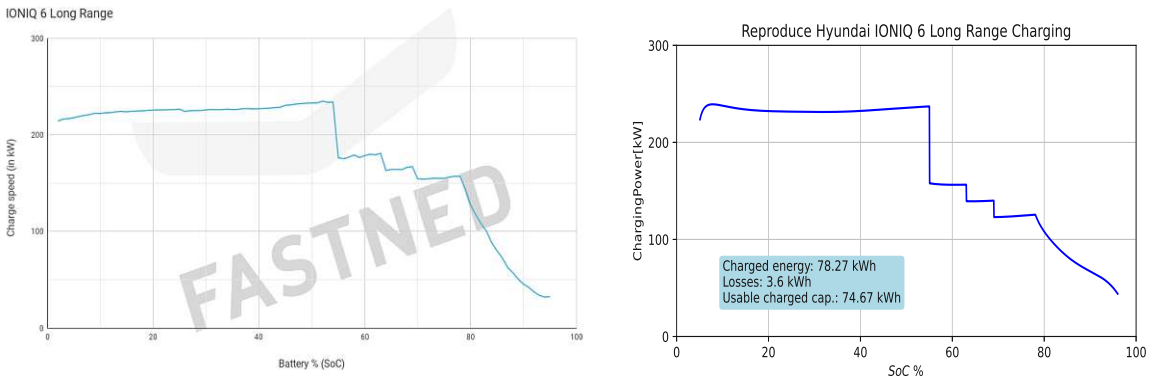


Figure 4.11: Charging curve of Hyundai IONIQ 6 Long Range provided by Fastned [111] and the reproduced, simulated curve based on publicly available data.

.

The post-processing results that give information on the time-dependent profile of charging losses are shown in Figure 4.12. These provide information of charging losses, and the profile of the accumulated charged energy into the EV's pack. They could not be compared to any publicly available profiles, as no such profiles were found.

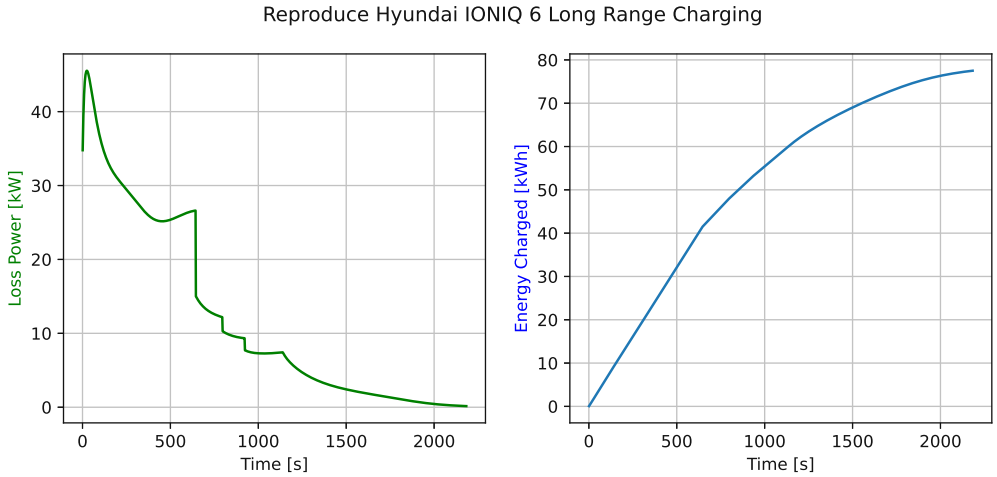


Figure 4.12: Profile of the loss power and charged amount of energy plotted over time. $t = 0$ denotes the beginning of the charging process. The simulation results are depicted.

Looking at the simulated charging voltage, current, and power profiles leaves room for discussion.

- The charging power is mainly governed by the charging current. Note the same shape of the "ChargingPower" and "ChargingCurrent" curves.
- The simulation tool calculates the charging voltage that an EVSE *should* deliver in order to fulfill the requirements posed by the different charging algorithms. In this case they are MCC and CV. It seems to be the case for the supply equipment at which was used to record the charging process of the real car, since:
  1. The recorded and simulated "ChargingPower" curves are in accordance
  2. The charged amount of energy corresponds to the energy-related battery capacity of the car battery pack according to manufacturer.
- The charging voltage corresponds to the battery voltage plus some offset. This offset in this thesis is by definition 2%. Note the equation (3.66) and the accompanying evaluation. Therefore, by looking at the curve "Voltage[V]" not only the delivered voltage by the EVSE, but also the shape of the battery pack voltage is known. This is namely exactly the battery voltage and the added offset of two percent. The deviations of the observed, depicted charging-voltage profile from the well-known open circuit-like ones from manufacturer datasheets are evaluated in the following.
  - The initial huge voltage jump is due to the fact, that one applies a strong initial current to the battery. This leads to a voltage jump on its internal resistance.

  - Note that the reason behind such jumps builds the basics of pulse discharge measurements. They could be observed in the practice when performing pulse discharge measurements, see previous section on validation.
  - One would expect the battery voltage to rise with SoC. Therefore at the first glance it might be surprising that in the range $SoC < 0.35$ it is decreasing. Considering the discussed model properties again, it can be explained why it is so. The big jump in the battery voltage due to the initially applied strong current relaxes. The relaxation procedure is governed by the transient voltage equation, therefore the voltage decreases with an exponential-like profile.
  - After having already relaxed at $SoC \approx 0.35$ (this is after about 400 second, see Figures 4.9 and 4.10) the voltage arises again. This is because after relaxation the battery voltage is mainly governed again by the open-circuit voltage part. It is already discussed that V_{oc} grows with the State of Charge.

- Voltage drops can be observed after switching from one CC phase to a next one. This results again in declining battery voltage despite a higher SoC value. The reason is that suddenly a big difference in the battery current ΔI appears. This leads to a voltage jump ΔV due to the internal resistance, according to the Ohmic relation $\Delta V = R \cdot \Delta I$

- Plotting over SoC might give a distorted image on charging. This becomes evident when comparing the plots in the Figures 4.9 and 4.10. When plotting over SoC the battery-protecting CV phase might seem very short. However, considering its length with respect to time, it gets clear why is it unfeasible to have long continuous voltage stages during charging:
  - CV-phase happens between $SoC = 0.78$ and $SoC = 0.99$. This corresponds to about 21% of the amount of charge that is put into the car's battery.
  - However, the CV-phase takes almost half of the time of the charging process, almost 1000 seconds.

- The complicatedly-seeming voltage profile explains why battery management systems have a "hard job" when estimating the State of Charge based on the measurable battery terminal voltage. On the other hand, it justifies why both the approaches SoC and voltage based transition criteria are implemented in the simulation framework. Note again that in case of simulation, one does not have to bother about obtaining the SoC based on the battery voltage. This fact facilitates working with the simulation model.

5 Discussion and outlooks

In this chapter, the implemented environment is assessed and the results of the performed simulations and laboratory tests are analysed. Recommendations for possible improvements are given. Finally, there is a discussion on feasible applications of the developed model tools.

5.1 Recapitulation

After briefly introducing general aspects on battery modeling, -management systems, -degradation and EV technology, the development of a complete framework for the characterization and simulation of battery packs and conventional EV-charging technologies was introduced. The novelty of the method is based on an innovative combination of multiple, already well-known and accepted aspects of physics, battery modeling, numerics, mathematics and computing.

The presented method showed up an excellent performance in comparison with experiments and publicly available data, see Chapter 4. Simulations are easily reproducible by simply saving the respective configuration files.

5.2 Model analysis

5.2.1 Physical model analysis: Thevenin model

The Thevenin model is a well established and proven approach in the battery community. This is due to its simplicity, good understability, and visualization potential. Another important quality is that it does not require extensive computational resources and coupled partial differential equations that need to be solved in order to generate simulation output. The evaluation of this electric-circuit based approach is topic of 2.5.2 and 3.2.

The property of being simple and resource-saving is especially true for the subset of Thevenin models used in this work: It is the case of using one RC element to model the transient part of the battery voltage when responding to currents. Its accuracy and effectiveness have been demonstrated in this work.

5.2.2 Solver analysis: Explicit Euler and Finite Difference method

The first-order explicit forward Euler method is one of the simplest numerical methods for solving ordinary differential equations. Although there are more advanced and faster convergence methods [14], they are more computationally expensive and require more memory. The explicit Euler procedure, on the other hand, provides a simple approach to develop discretized equations that can be solved even in the case of simulation of fast

charging algorithms, see 3.6.

The simplest numerical approximation to differential equations can be derived by replacing the differentiation by a discrete difference quotient. In case of the simulation tool it is done with respect to the variable t (time).

This leads to the finite difference method that is widely used in engineering applications due to its simplicity easy implementation. It is to be noted, that when using this method on some data y_d, it is a requirement that y_d has to be continuous. It is fulfilled in real-case scenarios when battery packs are charged or discharged, making this simple method suitable for the simulation framework.

The validation procedure has shown that the used simple solvers work with sufficient accuracy for solving the ordinary differential equations arising from the Thevenin model and charging boundary conditions. When they are combined with the parametrization methods presented in this work, they even lead to excellent results (sub-per mille average error!) see Chapter 4.

5.2.3 Numerical analysis: Adaptive polynomial fitting

A good rule of thumb in numerics is that it is desirable to work with smooth functions. The traditional method of using lookup tables to determine the functional dependence of $SoC - X$ does not satisfy this condition, because it is only C^0 continuous. On the other hand, polynomials have the "best" smoothness property by being infinitely often continuously differentiable.

Another important property when fitting with polynomials is the convergence behavior as discussed in 3.3. Moreover, performing coefficient determination for polynomials by the least-squares method on experimental data has a guaranteed success in case of having distinct ordinate points. This is not the case for the coefficients of other non-linear models presented in the literature, see 2.5.2.

The last remaining issue of considering deviations between individual samples belonging to the same cell type is addressed by using adaptive polynomial fitting. It avoids focusing too much on the properties of a single sample, which would lead to a loss of generality. Meanwhile, it still allows to apply a polynomial degree as high as possible, thus leading to good convergence, as discussed in 3.3.

5.3 Software analysis

The models are implemented in a Python software package. Python is a good compromise between ease of handling and computational speed. Since the implementation uses optimized data structures in form of numpy arrays the run-time of the software is kept very low (approximately 2 seconds even for around 10^5 simulation steps even so that I/O operations are not yet optimized). NumPy is one of the most popular Python tools used by developers and data scientists to support large-scale computing. It provides libraries and techniques to work with arrays and matrices, which allows this high performance: It is supported by code written in fast languages like C, C++, and Fortran [113]. Due to its C support, NumPy is not subject to Python's computational restrictions. Optimization is

partly achieved by the fact that all numpy arrays are homogeneous: Every item takes up the same size block of memory, and all blocks are interpreted in exactly the same way [114].

The other reason for the software of this work being fast is based on the well-reasoned choice of the physical basic model, polynomial interpolation and ordinary differential equation- and finite difference solver that are discussed in the previous subsections. When using them, the idea of temporal locality as discussed in the first Chapter is being exploited: The software temporarily switches the values of the pre-allocated variables representing the previous and current parameters/observables.

The Python implementation with the high performance data-types has another advantage: Python is open-source, portable and works computer-architecture independent. It allows the codes being versatile and portable.

5.4 Lab setup analysis

To parameterize the first-order Thevenin model a pulse discharge measurement procedure in 3.2 and 3.3 was established. It requires specific lab equipment. However, once it is available the procedure delivers many data-points within one batch. With the appropriate measuring instrument the procedure can be done in a completely automatized manner, therefore the procedure is optimized in terms of human time resource.

5.5 Charging algorithm simulator analysis

Based on the literature, conditions are implemented that have to be fulfilled according to the different charging modes. Only the most common (commercially observed) algorithms are implemented: CC, CV, MCC, BC (voltage and current based) and their arbitrary combinations with transition criteria based either on SoC or voltage thresholds. The modular nature of the software and the modeling approach, based on solving differential equations governed by the physical battery model, allow this set of algorithms to be easily extended with future ones.

5.6 Limitations, and recommendations

As it has been shown, modeling the battery behavior with the first-order Thevenin model and the numerical solvers/functional $SoC - X$ dependencies established in this work gives very accurate results, see Chapter 4 for validation. However, even better accuracy could be achieved by adding more RC elements to the equivalent circuit model [83], [90], [97]. This would not change much in the applied equations to be solved: It would result in additional transient voltage values $v_2, v_3 \ldots$, depending on how many additional RC pairs are added. It is questionable whether this is a necessary step at all, considering that the average error of the method is currently below 0.01%. see Table 4.1.

Another aspect which is not captured by the presented models is the degradation and change of the battery due to cycling and calendar aging. This would lead to a change of the X-values in the equivalent circuit depending on the number of cycles, the calendar

age of the cell/pack and the operating conditions. Considering all these aspects could lead to arbitrarily complicated equation systems. A good approach to avoid the need to implement analytical expressions that capture the above effects could be to perform measurements on used cells and obtain their degraded X values this way. To consider calendar-based degradation and cycle-based degradation separately, a fit (e.g., polynomial) could be applied to

1. Cells cycled many times within a very short calendar time.
2. Cells which have been in storage without cycling for a long period of calendar time.

Temperature dependency is implemented indirectly by allowing $SoC - X$ data to be captured at various ambient temperatures and stored in the database. When running the simulation, the user can select the desired temperature. This could be improved by establishing models for scaling the X values as a function of temperature. This could be represented by a temperature scaling function $\alpha_T(T)$ that depends only on T. A reference temperature T_0 (following the literature 25 °C could be defined as $\alpha_T(T_0). = 1$). Therefore $X(T)$ would be given as $X(T) = X \cdot \alpha_T(T)$. This would not change the applied equations, since they are independent of temperature. This means that this separation ansatz would not require any modification except the extension of the software with modules that implement α_T.

5.7 Outlooks

Due to its modular structure the developed software is well suitable for extensions and an integration into other frameworks. The presented work offers a highly efficient approach for characterizing arbitrary rechargeable batteries and simulating their transient terminal voltage under various load conditions. It enables a flexible scale up of cells and charging scenarios according to users' wishes.

It is intended that the models will be further developed by experts at the Austrian Institute of Technology, including the author. It is planned to

- Assist in testing and evaluating newly developed battery cells
- Simulate hybrid power plant components
- Form the basis of an electric vehicle emulation tool. This would allow realistic testing of EVSE devices as well as hardware-in-the-loop network testing.
- With certain modifications, the models will be extended to simulate fuel cells. This would improve the design and simulation of modern green energy based power plants.

In addition, further validation and extension of the software tool is planned based on the measurement of voltage and current profiles of electric vehicles during charging. For this purpose, the author and experts from the Austrian Institute of Technology GmbH are currently building a "man-in-the-middle" measurement device which is plugged between the charger and the car. The tests will be carried out on the 50 kW EVSE of the SmartEST laboratory as well as on public high-power charging stations with a variety of electric cars.

It is intended to extend the SQLite database of the software written by the author with parameterization data of many battery cells, including high-power ones of electric vehicles. For this purpose, the method developed in this thesis will be used. The measurements will be conducted in the battery test laboratory of the Austrian Institute of Technology GmbH. The parameterization of a sufficiently large variety of cell types will allow even more accurate simulation results for a wide range of application cases. For the aforementioned network planning purposes, simulations with different cell types can be performed to suggest the optimal choice for the respective storage system.

On the innovative aspects of this work the author and her supervisors from the Austrian Institute of Technology GmbH have submitted a manuscript to the 2024 Annual Conference of the IEEE Industrial Electronics Society that has been accepted for publication. The author will present these aspects on the conference that is held in November 2024, Chicago, Illinois. Further publications and examinations on the extensions, use cases and development of the models are intended. Especially incorporating temperature and aging/ degradation effects into the existing framework is intended. It will be based on the ideas in 5.6 and already existing approaches developed at the Austrian Institute of Technology GmbH.

It can be concluded that the developed simulation framework has the potential to be used for various high performance applications such as battery emulation or grid simulation. It will serve as a valuable tool in advancing battery technology and energy systems through continuous research and development.

Bibliography

[1] International Energy Agency, "Electricity 2024: Executive summary", International Energy Agency (IEA), Tech. Rep., 2024. [Online]. Available: https://www.iea.org/reports/electricity-2024/executive-summary.

[2] International Energy Agency, "Net zero emissions by 2050 scenario (nze)", International Energy Agency (IEA), Tech. Rep., 2024. [Online]. Available: https://www.iea.org/reports/global-energy-and-climate-model/net-zero-emissions-by-2050-scenario-nze#abstract.

[3] D. o. E. United Nations and S. D. Social Affairs, "The 17 goals", United Nations, Tech. Rep., 2023. [Online]. Available: https://sdgs.un.org/goals.

[4] T. I. P. on Climate Change, "Climate change 2022: Mitigation of climate change", The Intergovernmental Panel on Climate Change (IPCC), Tech. Rep., 2022. [Online]. Available: https://www.ipcc.ch/report/sixth-assessment-report-working-group-3/.

[5] C. Brunner, G. Deac, S. Braun, and C. Zöphel, "The future need for flexibility and the impact of fluctuating renewable power generation", *Renewable Energy*, vol. 149, pp. 1314–1324, 2020.

[6] P. Rullo, L. Braccia, P. Luppi, D. Zumoffen, and D. Feroldi, "Integration of sizing and energy management based on economic predictive control for standalone hybrid renewable energy systems", *Renewable energy*, vol. 140, pp. 436–451, 2019.

[7] A. Sangswang and M. Konghirun, "Optimal strategies in home energy management system integrating solar power, energy storage, and vehicle-to-grid for grid support and energy efficiency", *IEEE Transactions on Industry Applications*, vol. 56, no. 5, pp. 5716–5728, 2020.

[8] International Energy Agency, "Outlook for battery demand and supply", International Energy Agency (IEA), Tech. Rep., 2024. [Online]. Available: https://www.iea.org/reports/batteries-and-secure-energy-transitions/outlook-for-battery-demand-and-supply#abstract.

[9] International Energy Agency, "Global ev outlook 2024", International Energy Agency (IEA), Tech. Rep., 2024. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2024.

[10] F. Mohammadi and M. Saif, "A comprehensive overview of electric vehicle batteries market", *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 3, p. 100 127, 2023.

[11] C. Maier, C. Pfeiffer, and M. Millendorfer, "A comparative analysis of perceived drivers and barriers for electric vehicle usage among users and non-users in austria", *Available at SSRN 4015252*,

[12] "APP Metrics for Intel® Microprocessors", Intel Corporation. (), [Online]. Available: https://www.intel.com/content/dam/support/us/en/documents/processors/APP-for-Intel-Core-Processors.pdf (visited on 05/15/2024).

[13] B. Jacob, D. Wang, and S. Ng, *Memory systems: cache, DRAM, disk*. Morgan Kaufmann, 2010.

[14] T. Goudon, *Mathematics for Modeling and Scientific Computing*, eng, 1st. Newark: John Wiley Sons, Incorporated, 2016, ISBN: 1848219881.

[15] M. A. Rosen and A. Farsi, *Battery technology : from fundamentals to thermal behavior and management*, eng. London, England: Academic Press, 2023, ISBN: 0443188637.

[16] Statista. "Global battery market size, by technology". (), [Online]. Available: https://www.statista.com/statistics/1339880/global-battery-market-size-by-technology/ (visited on 04/04/2024).

[17] R. K. Gupta, *Solid State Batteries Volume 1.* (ACS Symposium), eng. Washington, DC: American Chemical Society, 2022, ISBN: 0841297673.

[18] "USABC", United States Advanced Battery Consortium LLC (USABC). (), [Online]. Available: https://uscar.org/usabc/ (visited on 05/16/2024).

[19] " QC/T 743-2006 PDF in English", ChineseStandard. (), [Online]. Available: https://www.chinesestandard.net/PDF.aspx/QCT743-2006 (visited on 05/16/2024).

[20] M. Ecker, J. B. Gerschler, J. Vogel, *et al.*, "Development of a lifetime prediction model for lithium-ion batteries based on extended accelerated aging test data", *Journal of Power Sources*, vol. 215, pp. 248–257, 2012.

[21] X. Tan, A. Vezzini, Y. Fan, N. Khare, Y. Xu, and L. Wei, *Battery Management System and Its Applications*. John Wiley and Sons, 2022.

[22] S. Saxena, C. Le Floch, J. MacDonald, and S. Moura, "Quantifying ev battery end-of-life through analysis of travel needs with vehicle powertrain models", *Journal of Power Sources*, vol. 282, pp. 265–276, 2015, ISSN: 0378-7753. DOI: https://doi.org/10.1016/j.jpowsour.2015.01.072. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775315000841.

[23] A. Guha and A. Patra, "State of health estimation of lithium-ion batteries using capacity fade and internal resistance growth models", *IEEE Transactions on Transportation Electrification*, vol. 4, no. 1, pp. 135–146, 2017.

[24] J. Kim, J. Oh, and H. Lee, "Review on battery thermal management system for electric vehicles", *Applied thermal engineering*, vol. 149, pp. 192–212, 2019.

[25] E. Schuster, C. Ziebert, A. Melcher, M. Rohde, and H. J. Seifert, "Thermal behavior and electrochemical heat generation in a commercial 40 ah lithium ion pouch cell", *Journal of Power Sources*, vol. 286, pp. 580–589, 2015.

[26] A. Tomaszewska, Z. Chu, X. Feng, *et al.*, "Lithium-ion battery fast charging: A review", *eTransportation*, vol. 1, p. 100 011, 2019, ISSN: 2590-1168. DOI: https://doi.org/10.1016/j.etran.2019.100011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590116819300116.

[27] M. Fleckenstein, O. Bohlen, M. A. Roscher, and B. Bäker, "Current density and state of charge inhomogeneities in li-ion battery cells with lifepo4 as cathode material due to temperature gradients", *Journal of Power Sources*, vol. 196, no. 10, pp. 4769–4778, 2011, ISSN: 0378-7753. DOI: https://doi.org/10.1016/j.jpowsour.2011.01.043. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775311001558.

[28] W. Song, M. Chen, F. Bai, S. Lin, Y. Chen, and Z. Feng, "Non-uniform effect on the thermal/aging performance of lithium-ion pouch battery", *Applied Thermal Engineering*, vol. 128, pp. 1165–1174, 2018, ISSN: 1359-4311. DOI: https://doi.org/10.1016/j.applthermaleng.2017.09.090. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359431117350986.

[29] B. Bose, A. Garg, B. Panigrahi, and J. Kim, "Study on li-ion battery fast charging strategies: Review, challenges and proposed charging framework", *Journal of Energy Storage*, vol. 55, p. 105 507, 2022.

[30] T. Waldmann, M. Wilka, M. Kasper, M. Fleischhammer, and M. Wohlfahrt-Mehrens, "Temperature dependent ageing mechanisms in lithium-ion batteries – a post-mortem study", *Journal of Power Sources*, vol. 262, pp. 129–135, 2014, ISSN: 0378-7753. DOI: https://doi.org/10.1016/j.jpowsour.2014.03.112. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775314004352.

[31] A. Barré, B. Deguilhem, S. Grolleau, M. Gérard, F. Suard, and D. Riu, "A review on lithium-ion battery ageing mechanisms and estimations for automotive applications", *Journal of Power Sources*, vol. 241, pp. 680–689, 2013, ISSN: 0378-7753. DOI: https://doi.org/10.1016/j.jpowsour.2013.05.040. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775313008185.

[32] S. J. An, J. Li, C. Daniel, D. Mohanty, S. Nagpure, and D. L. Wood, "The state of understanding of the lithium-ion-battery graphite solid electrolyte interphase (sei) and its relationship to formation cycling", *Carbon*, vol. 105, pp. 52–76, 2016, ISSN: 0008-6223. DOI: https://doi.org/10.1016/j.carbon.2016.04.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0008622316302676.

[33] S. Yan and A. C. Marschilok, "Conversion-type electrodes for rechargeable lithium based batteries: Case studies of iron based conversion materials for lithium-ion batteries and molybdenum disulfides for lithium-sulfur batteries", in *Encyclopedia of Energy Storage*, L. F. Cabeza, Ed., Oxforpd: Elsevier, 2022, pp. 36–46, ISBN: 978-0-12-819730-1. DOI: https://doi.org/10.1016/B978-0-12-819723-3.00116-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128197233001165.

[34] "A look inside your battery: Watching the dendrites grow". (2020), [Online]. Available: https://www.batterypoweronline.com/news/a-look-inside-your-battery-watching-the-dendrites-grow/ (visited on 06/13/2024).

[35] Y. Shen, S. Wang, H. Li, K. Wang, and K. Jiang, "An overview on in situ/operando battery sensing methodology through thermal and stress measurements", *Journal of Energy Storage*, vol. 64, p. 107 164, 2023.

[36] B. Suthar, V. Ramadesigan, S. De, R. D. Braatz, and V. R. Subramanian, "Optimal charging profiles for mechanically constrained lithium-ion batteries", *Physical Chemistry Chemical Physics*, vol. 16, no. 1, pp. 277–287, 2014.

[37] "Battery Efficiency", PVsyst 7 Help. (), [Online]. Available: `https://www.pvsyst.com/help/battery_efficiency.htm` (visited on 05/20/2024).

[38] BM für Verkehr, Bau und Stadtentwicklung (BRD). "What is the capacity of the battery? ritar power". (2013), [Online]. Available: `https://www.ritarpower.com/service/158.html`.

[39] J. Lee, Y. Kim, and H. Cha, "A new battery parameter identification considering current, soc and peukert's effect for hybrid electric vehicles", in *2011 IEEE Energy Conversion Congress and Exposition*, IEEE, 2011, pp. 1489–1494.

[40] S. U. Jeon, J.-W. Park, B.-K. Kang, and H.-J. Lee, "Study on battery charging strategy of electric vehicles considering battery capacity", *IEEE Access*, vol. 9, pp. 89 757–89 767, 2021.

[41] I. E. Commission, *Electric vehicle conductive charging system–part 1: General requirements*, 2017.

[42] S. Palanisamy, S. Chenniappan, and S. Padmanaban, *Fast-Charging Infrastructure for Electric and Hybrid Electric Vehicles: Methods for Large-Scale Penetration into Electric Distribution Networks*, eng, 1st ed. Newark: Wiley, 2023, ISBN: 1119987741.

[43] "Charging for all", Tesla. (2024), [Online]. Available: `https://www.tesla.com/NACS` (visited on 05/20/2024).

[44] J. Schmutzler, C. A. Andersen, and C. Wietfeld, "Evaluation of ocpp and iec 61850 for smart charging electric vehicles", eng, *World electric vehicle journal*, vol. 6, no. 4, pp. 863–874, 2013, ISSN: 2032-6653.

[45] M. Nöhrer, "Flexible co-simulationsumgebung zum testen von ladeinfrastrukturen für elektromobilität", Ph.D. dissertation, 2014.

[46] F. Rücker, I. Schoeneberger, T. Wilmschen, *et al.*, "Self-sufficiency and charger constraints of prosumer households with vehicle-to-home strategies", *Applied Energy*, vol. 317, p. 119 060, 2022.

[47] B. Vaidya and H. T. Mouftah, "Deployment of secure ev charging system using open charge point protocol", in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, IEEE, 2018, pp. 922–927.

[48] "The complete ocpp guide". (), [Online]. Available: `https://www.ampeco.com/guides/complete-ocpp-guide/?utm_medium=cpc&utm_source=google&utm_campaign=Search-EU&utm_adgroup=OCPP-B&utm_content=OCPP-B&utm_term=ocpp%202.0&_gl=1*1kf6gsz*_up*MQ..&gclid=Cj0KCQjwir2xBhC_ARIsAMTXk87xtjK2PAvD2IQhSz79VtnQRSZGfpaAk0iEhEcXIGfbOkhWB7V52UUaAjxxEALw_wcB#6-the-functional-blocks-of-ocpp` (visited on 04/29/2024).

[49] O. C. Alliance, "Wwcp ocpp", Open Charge Alliance, Tech. Rep., 2024. [Online]. Available: `https://github.com/OpenChargingCloud/WWCP_OCPP`.

[50] D. Ramasamy, "Possible hardware architectures for power line communication in automotive v2g applications", *Journal of The Institution of Engineers (India): Series B*, vol. 104, no. 3, pp. 813–819, 2023.

[51] M. U. Tahir, A. Sangwongwanich, D.-I. Stroe, and F. Blaabjerg, "Overview of multi-stage charging strategies for li-ion batteries", *Journal of Energy Chemistry*, vol. 84, pp. 228–241, 2023, ISSN: 2095-4956. DOI: https://doi.org/10.1016/j.jechem.2023.05.023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2095495623003091.

[52] N. Wassiliadis, J. Schneider, A. Frank, *et al.*, "Review of fast charging strategies for lithium-ion battery systems and their applicability for battery electric vehicles", *Journal of energy storage*, vol. 44, p. 103 306, 2021.

[53] S. S. Zhang, "The effect of the charging protocol on the cycle life of a li-ion battery", *Journal of Power Sources*, vol. 161, no. 2, pp. 1385–1391, 2006, ISSN: 0378-7753. DOI: https://doi.org/10.1016/j.jpowsour.2006.06.040. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775306011839.

[54] Y. Li, K. Li, Y. Xie, J. Liu, C. Fu, and B. Liu, "Optimized charging of lithium-ion battery for electric vehicles: Adaptive multistage constant current–constant voltage charging strategy", *Renewable Energy*, vol. 146, pp. 2688–2699, 2020, ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2019.08.077. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148119312649.

[55] A. B. Khan and W. Choi, "Optimal charge pattern for the high-performance multistage constant current charge method for the li-ion batteries", *IEEE Transactions on Energy Conversion*, vol. 33, no. 3, pp. 1132–1140, 2018. DOI: 10.1109/TEC.2018.2801381.

[56] L. Patnaik, A. V. J. S. Praneeth, and S. S. Williamson, "A closed-loop constant-temperature constant-voltage charging technique to reduce charge time of lithium-ion batteries", *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1059–1067, 2019. DOI: 10.1109/TIE.2018.2833038.

[57] M. Sabarimuthu, N. Senthilnathan, and M. Kamalesh, "Multi-stage constant current–constant voltage under constant temperature (mscc-cv-ct) charging technique for lithium-ion batteries in light weight electric vehicles (evs)", *Electrical Engineering*, vol. 105, no. 6, pp. 4289–4309, 2023.

[58] U. S. Kim, J. Yi, C. B. Shin, T. Han, and S. Park, "Modeling the thermal behaviors of a lithium-ion battery during constant-power discharge and charge operations", *Journal of The Electrochemical Society*, vol. 160, no. 6, A990, 2013.

[59] J. M. Amanor-Boadu and A. Guiseppi-Elie, "Improved performance of li-ion polymer batteries through improved pulse charging algorithm", *Applied Sciences*, vol. 10, no. 3, p. 895, 2020.

[60] S. Li, Q. Wu, D. Zhang, *et al.*, "Effects of pulse charging on the performances of lithium-ion batteries", *Nano Energy*, vol. 56, pp. 555–562, 2019.

[61] M. Song and S.-Y. Choe, "Fast and safe charging method suppressing side reaction and lithium deposition reaction in lithium ion battery", *Journal of Power Sources*, vol. 436, p. 226 835, 2019.

[62] X. Huang, Y. Li, A. B. Acharya, *et al.*, "A review of pulsed current technique for lithium-ion batteries", *Energies*, vol. 13, no. 10, p. 2458, 2020.

[63] H. Fang, C. Depcik, and V. Lvovich, "Optimal pulse-modulated lithium-ion battery charging: Algorithms and simulation", *Journal of Energy Storage*, vol. 15, pp. 359–367, 2018.

[64] S. Wang, K. Kuang, X. Han, Z. Chu, L. Lu, and M. Ouyang, "A model-based continuous differentiable current charging approach for electric vehicles in direct current microgrids", *Journal of Power Sources*, vol. 482, p. 229 019, 2021, ISSN: 0378-7753. DOI: https://doi.org/10.1016/j.jpowsour.2020.229019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775320313161.

[65] Z. Guo, B. Y. Liaw, X. Qiu, L. Gao, and C. Zhang, "Optimal charging method for lithium ion batteries using a universal voltage protocol accommodating aging", *Journal of Power Sources*, vol. 274, pp. 957–964, 2015.

[66] A. Abdollahi, X. Han, G. Avvari, *et al.*, "Optimal battery charging, part i: Minimizing time-to-charge, energy loss, and temperature rise for ocv-resistance battery model", *Journal of Power Sources*, vol. 303, pp. 388–398, 2016, ISSN: 0378-7753. DOI: https://doi.org/10.1016/j.jpowsour.2015.02.075. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775315003092.

[67] F. Support. "Vehicles - fastned charging". (), [Online]. Available: https://support.fastned.nl/hc/en-gb/sections/4428932764573-Vehicles (visited on 04/29/2024).

[68] I. (2023). "Global ev outlook 2023, iea, paris". (), [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2023 (visited on 04/29/2024).

[69] M. S. Houache, C.-H. Yim, Z. Karkar, and Y. Abu-Lebdeh, "On the current and future outlook of battery chemistries for electric vehicles—mini review", *Batteries*, vol. 8, no. 7, p. 70, 2022.

[70] P. Nzereogu, A. Omah, F. Ezema, E. Iwuoha, and A. Nwanya, "Anode materials for lithium-ion batteries: A review", *Applied Surface Science Advances*, vol. 9, p. 100 233, 2022, ISSN: 2666-5239. DOI: https://doi.org/10.1016/j.apsadv.2022.100233. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666523922000253.

[71] X. Liu, K. Li, and X. Li, "The electrochemical performance and applications of several popular lithium-ion batteries for electric vehicles-a review", in *Advances in Green Energy Systems and Smart Grid: First International Conference on Intelligent Manufacturing and Internet of Things and 5th International Conference on Computing for Sustainable Energy and Environment, IMIOT and ICSEE 2018, Chongqing, China, September 21-23, 2018, Proceedings, Part III 5*, Springer, 2018, pp. 201–213.

[72] S.-P. Chen, D. Lv, J. Chen, Y.-H. Zhang, and F.-N. Shi, "Review on defects and modification methods of lifepo4 cathode material for lithium-ion batteries", *Energy & Fuels*, vol. 36, no. 3, pp. 1232–1251, 2022.

[73] Y. Horowitz, C. Schmidt, D.-h. Yoon, *et al.*, "Between liquid and all solid: A prospect on electrolyte future in lithium-ion batteries for electric vehicles", *Energy Technology*, vol. 8, no. 11, p. 2 000 580, 2020.

[74] E. Database. "Electric vehicle database". (), [Online]. Available: ev-database.org (visited on 04/29/2024).

[75] E. Database. "Renault zoe ze50 r135". (2024), [Online]. Available: https://ev-database.org/car/1205/Renault-Zoe-ZE50-R135.

[76] "Possible upgrades with ncm 712 battery cells from lg chem", PushEVs. (2020), [Online]. Available: https://pushevs.com/2020/04/23/possible-upgrades-with-ncm-712-battery-cells-from-lg-chem/ (visited on 05/23/2024).

[77] L. E. Solution. "Cell solutions for powerful ev performance". (2024), [Online]. Available: https://www.lgensol.com/assets/file/202312_LGES_cell_spec_en.pdf.

[78] Z. Lyu and R. Gao, "A model-based and data-driven joint method for state-of-health estimation of lithium-ion battery in electric vehicles", *International Journal of Energy Research*, vol. 43, no. 14, pp. 7956–7969, 2019.

[79] X. Li, C. Yuan, and Z. Wang, "State of health estimation for li-ion battery via partial incremental capacity analysis based on support vector regression", *Energy*, vol. 203, p. 117 852, 2020.

[80] Y. Ma, P. Duan, Y. Sun, and H. Chen, "Equalization of lithium-ion battery pack based on fuzzy logic control in electric vehicle", *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6762–6771, 2018. DOI: 10.1109/TIE.2018.2795578.

[81] MathWorks. "Kalman". (2024), [Online]. Available: https://de.mathworks.com/help/control/ref/ss.kalman.html.

[82] R. R. Labbe. "Kalmanfilter". (2024), [Online]. Available: https://filterpy.readthedocs.io/en/latest/kalman/KalmanFilter.html.

[83] A. A.-H. Hussein and I. Batarseh, "An overview of generic battery models", in *2011 IEEE Power and Energy Society General Meeting*, IEEE, 2011, pp. 1–6.

[84] A. Fotouhi, D. J. Auger, K. Propp, S. Longo, and M. Wild, "A review on electric vehicle battery modelling: From lithium-ion toward lithium–sulphur", *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 1008–1021, 2016.

[85] S. Tamilselvi, S. Gunasundari, N. Karuppiah, *et al.*, "A review on battery modelling techniques", *Sustainability*, vol. 13, no. 18, p. 10 042, 2021.

[86] G. Fan, X. Li, and R. Zhang, "Global sensitivity analysis on temperature-dependent parameters of a reduced-order electrochemical model and robust state-of-charge estimation at different temperatures", *Energy*, vol. 223, p. 120 024, 2021.

[87] S. E. Li, B. Wang, H. Peng, and X. Hu, "An electrochemistry-based impedance model for lithium-ion batteries", *Journal of Power Sources*, vol. 258, pp. 9–18, 2014.

[88] M. Chen and G. A. Rincon-Mora, "Accurate electrical battery model capable of predicting runtime and iv performance", *IEEE transactions on energy conversion*, vol. 21, no. 2, pp. 504–511, 2006.

[89] M. d. F. B. Binelo, A. T. Z. R. Sausen, P. S. Sausen, and M. O. Binelo, "Mathematical modeling and parameter estimation of battery lifetime using a combined electrical model and a genetic algorithm", *TEMA (São Carlos)*, vol. 20, pp. 149–167, 2019.

[90] M.-K. Tran, A. DaCosta, A. Mevawalla, S. Panchal, and M. Fowler, "Comparative study of equivalent circuit models performance in four common lithium-ion batteries: Lfp, nmc, lmo, nca", *Batteries*, vol. 7, no. 3, p. 51, 2021.

[91] M. A. Roscher and D. U. Sauer, "Dynamic electric behavior and open-circuit-voltage modeling of lifepo4-based lithium ion secondary batteries", *Journal of Power Sources*, vol. 196, no. 1, pp. 331–336, 2011.

[92] S. J. Navas, G. C. González, F. Pino, and J. Guerra, "Modelling li-ion batteries using equivalent circuits for renewable energy applications", *Energy Reports*, vol. 9, pp. 4456–4465, 2023, ISSN: 2352-4847. DOI: https://doi.org/10.1016/j.egyr.2023.03.103. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484723003414.

[93] C. Seitl, "Development and implementation of a battery emulator for power hardware-in-the-loop simulations", Ph.D. dissertation, 2014.

[94] M. Zheng, B. Qi, and X. Du, "Dynamic model for characteristics of li-ion battery on electric vehicle", in *2009 4th IEEE Conference on Industrial Electronics and Applications*, 2009, pp. 2867–2871. DOI: 10.1109/ICIEA.2009.5138733.

[95] L. Lam, P. Bauer, and E. Kelder, "A practical circuit-based model for li-ion battery cells in electric vehicle applications", in *2011 IEEE 33rd International Telecommunications Energy Conference (INTELEC)*, IEEE, 2011, pp. 1–9.

[96] C. R. Lashway and O. A. Mohammed, "Adaptive battery management and parameter estimation through physics-based modeling and experimental verification", *IEEE Transactions on Transportation Electrification*, vol. 2, no. 4, pp. 454–464, 2016. DOI: 10.1109/TTE.2016.2558843.

[97] A. Rahmoun and H. Biechl, "Modelling of li-ion batteries using equivalent circuit diagrams", *Przeglad Elektrotechniczny*, vol. 88, no. 7, pp. 152–156, 2012.

[98] O. Arregi, E. Agirrezabala, U. Iraola, *et al.*, "A new equivalent circuit model parametrization methodology based on current pulse tests for different battery technologies", *Energies*, vol. 14, no. 21, p. 7255, 2021.

[99] "Scipy.optimize.curve$_{f}it$", The SciPy community. (2008), [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html (visited on 05/25/2024).

[100] C. Balaji, "Curve fitting", eng, in *Thermal System Design and Optimization*, Switzerland: Springer International Publishing AG, 2021, pp. 69–128, ISBN: 3030590453.

[101] N. Developers. "Numpy.polyfit". (2024), [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html.

[102] "Ieee standard for floating-point arithmetic", *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84, 2019. DOI: 10.1109/IEEESTD.2019.8766229.

[103] J. Stoer, R. Bulirsch, R. Bartels, W. Gautschi, and C. Witzgall, *Introduction to numerical analysis*. Springer, 1980, vol. 2.

[104] MathWorks. "Battery (table-based)". (), [Online]. Available: https://de.mathworks.com/help/sps/ref/batterytablebased.html (visited on 04/19/2024).

[105] D. M. Hawkins, "The problem of overfitting", *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[106] P. H. Notten, J. O. het Veld, and J. Van Beek, "Boostcharging li-ion batteries: A challenging new charging concept", *Journal of Power Sources*, vol. 145, no. 1, pp. 89–94, 2005.

[107] MathWorks. "Estimation equivalent circuit battery". (), [Online]. Available: %7Bhttps://de.mathworks.com/help/autoblks/ref/estimationequivalentcircuitbattery.html%7D (visited on 04/19/2024).

[108] MathWorks. "Ode45". (), [Online]. Available: https://de.mathworks.com/help/matlab/ref/ode45.html (visited on 04/19/2024).

[109] L. F. Shampine and M. W. Reichelt, "The matlab ode suite", *SIAM journal on scientific computing*, vol. 18, no. 1, pp. 1–22, 1997.

[110] E. Database. "Hyundai ioniq 6 77.4 kwh - full specs and price". (), [Online]. Available: https://ev-database.org/de/pkw/1718/Hyundai-IONIQ-6-774-kWh (visited on 04/23/2024).

[111] F. Support. "Hyundai electric vehicles charging - fastned support". (), [Online]. Available: https://support.fastned.nl/hc/en-gb/articles/4405121276945-Hyundai (visited on 04/23/2024).

[112] "Generate parameter data for equivalent circuit battery block". (), [Online]. Available: https://de.mathworks.com/help/autoblks/ug/generate-parameter-data-for-estimations-circuit-battery-block.html#References (visited on 04/24/2024).

[113] S. Yegulalp. "What is numpy? faster array and matrix math in python". (2024), [Online]. Available: https://www.infoworld.com/article/3712923/what-is-numpy-faster-array-and-matrix-math-in-python.html (visited on 06/10/2024).

[114] "Array objects", NumPy Developers. (2008), [Online]. Available: https://numpy.org/doc/stable/reference/arrays.html (visited on 06/10/2024).