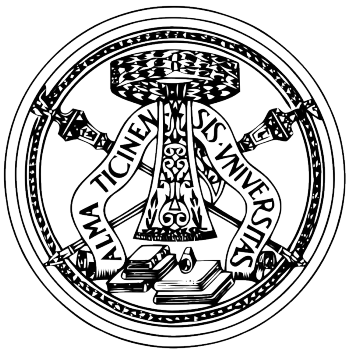


UNIVERSITY OF PAVIA

FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL, COMPUTER AND BIOMEDICAL ENGINEERING

PH.D. THESIS

Lithium-ion batteries: fast simulation, set-based state estimation and fault detection

Candidate: **Giacomo Saccani**

Supervisor: **Prof. Davide M. Raimondo**

XXXV CYCLE

Abstract Lithium-ion batteries have become in the last years a fundamental component for electrical energy storage thanks to the growing efforts in reducing fossil fuel consumption. Typical applications are portable electronics and electric/hybrid vehicles. The management of lithium-ion batteries is a challenging task handled by an appropriate device called Battery Management System (BMS). State estimation of fundamental quantities, such as State of Charge, is among the routines of the BMS. A direct application of state estimation is fault detection, where the estimated states are compared to the measurements obtained. If a discrepancy is noted, a fault might have occurred. In this work, a set-based method is proposed, based on constrained zonotopes. They are able to compute accurate enclosures and yield better results in state estimation than other set-based methods (e.g. intervals). In addition, if the battery pack is large, computations can become expensive. This works aims at solving this issue by proposing an acceleration method based on Waveform Relaxation techniques. The system of equations representing the dynamics is split in smaller subsystems that can be solved faster. Convergence to the same result of the original problem is obtained by iterating the resolution of the subsystems and applying appropriate convergence conditions.

Index of contents

List of Publications

The topics of this thesis have been published in the following articles

P.1 **Giacomo Saccani**, Gabriele Ciaramella, and Davide M. Raimondo. "A computationally efficient implementation of a battery pack electrochemical model using waveform relaxation." Journal of Energy Storage 46 (2022): 103758.

P.2 **Giacomo Saccani**, Diego Locatelli, Angelo Tottoli and Davide M. Raimondo "Model-Based Thermal Fault Detection in Li-Ion Batteries Using a Set-Based Approach." IFAC-PapersOnLine, 55(6), 329-334.

P.3 Diego Locatelli, Angelo Tottoli, **Giacomo Saccani** and Davide M. Raimondo "Thermal fault-detection in series connected Li-ion cells: a set-based approach using constrained zonotopes." 2022 IEEE Conference on Control Technology and Applications (CCTA). IEEE, 2022.

P.4 Diego Locatelli, **Giacomo Saccani**, Brenner Rego, Guillherme Raffo and Davide M Raimondo "Set-based joint state and parameter estimation of a Li-ion cell using constrained zonotopes." 2022 IEEE Vehicle Power and Propulsion Conference (VPPC). IEEE, 2022.

The paper *"Model-Based Thermal Fault Detection in Li-Ion Batteries Using a Set-Based Approach."* won the award for "Best Student Paper Award" in SAFEPROCESS 2022.

List of Figures

List of Tables

List of Abbreviations

AA Anderson Acceleration

AE Algebraic Equations

BMS Battery Management System

C-rate Charging Rate

DAE Differential Algebraic Equations

DFN Doyle-Fuller-Newman

DP Dual Polarization

ECM Equivalent Circuit Model

EM Electrochemical Models

EMF Electromotive Force

ESC External short-circuit

EV Electric Vehicle

FVM Finite Volume Method

ISC Internal short-circuit

Li-ion Lithium-ion

LS Least Squares

OC Over-charging

OCP Open-Circuit Potential

OD Over-discharging

ODE Ordinary Differential Equations

P2D Pseudo-two dimensional

PDAE Algebraic Partial Differential Equations

PDE Partial Differential Equations

SEI Solid Electrolyte Interface

SOC State of Charge

SOH State of Health

SPM Single Particle Model

SPMeT Single Particle Model with electrolyte and thermal dynamics

WR Waveform Relaxation

Part I

Introduction and Background

Chapter 1

Introduction

In the last years, lithium-ion batteries have become widespread in many commercial sectors. Nowadays, they are used as portable energy source in many devices, ranging from portable electronics to electric vehicles. The increasing diffusion of lithium-ion batteries is a consequence of the decarbonization process that many countries are trying to carry on by increasing investments in renewable sources. One of the few disadvantages of these latter is that energy production is not continuous, for example in photovoltaic and wind plants, and therefore it might not match the demand. In fact, electrical energy production from renewable power sources is characterized by spikes, as they depend from non-controllable phenomenon. Therefore batteries can be used to decouple production from the demand, in such a way that if the production exceeds the demand, electrical energy is stored in the system. Otherwise, if the demand exceeds production, it is possible to use the energy stored in the system to supply the requested power.

Lithium-ion batteries provide a possible answer to this problem thanks to their properties, such as a high energy density, good safety level and performance characteristic. These latter include a long useful life, low self-discharge rate and fast charging capabilities. In the last years, new technologies are being researched to replace lithium-ion batteries with different chemistry with better properties, such as solid state batteries, lithium-sulfur batteries and calcium batteries. However, to this date, lithium-ion is still the most used one. Managing batteries is not an easy task and in every application a device called Battery Management System is in charge of the operational aspects of the battery. One of the fundamental tasks performed by the Battery Management System is state estimation: because not all of the internal states can be measured directly, some algorithm is needed to make all the states accessible. A direct application of state estimation is fault detection, where the collected measures are compared with the estimated values and if a discrepancy is noted, a fault might have occurred. Faults are a very serious concern when dealing with lithium-ion batteries, as they can lead to explosions and electrical fires. The Battery Management System often relies on a mathematical model of the

lithium-ion cell; many different models have been developed during the years, from very simple ones to complex and complete models that describe in detail the dynamics of the electrochemical reactions. When the model is complicated and the battery pack is composed of hundreds of cells (e.g. in the case of an electric vehicle), the computations can become expensive both in terms of memory and time.

1.1 Thesis contribution

The aim of this thesis is to address the issues just presented. A complexity reduction method is proposed based on a Waveform Relaxation framework for the simulation of large battery packs: the basic idea is to divide a complex system in smaller subsystems that, if solved iteratively, return the solution of the original problem in a shorter time. In addition, appropriate acceleration techniques can be employed in order to further speed up the convergence of the method. The convergence of the method and its effectiveness are proven using two different electrochemical lithium-ion cell models.

The second part of the thesis is dedicated to state estimation and, in particular, set-based methods. A particular set representation, called constrained zonotope, is used for state estimation applied to fault detection. A single cell is considered to design the method that is afterwards extended to the case of a simple battery pack. Finally, the same state estimation framework is modified and adapted for a joint state and parameter estimation, proving that state estimation itself improves drastically when parameters and states are jointly estimated.

1.2 Thesis structure

This thesis has the following structure

- In Chapter 2 a introduction to the topic of lithium-ion batteries is given, with an historical background followed by the function principles of a lithium-ion cell and an overview of Battery Management Systems. To conclude the lithium-ion cells models used in this work are presented.

- In Chapter 3 the acceleration method for fast simulation of battery packs based on Waveform Relaxation is presented and detailed. Starting from the battery pack model, the algorithm is formulated and convergence is analyzed. Before concluding with simulation examples, an acceleration algorithm is built on top of Waveform Relaxation to further speed up simulations.

- In Chapter 4 the method presented in Chapter 3 is further improved by further refining the results with a different lithium-ion cell model

and generalizing the algorithms and the solver to fit any cell model. In addition, a full thermal model of the battery pack and surrounding elements is implemented.

- In Chapter 5, set-based state estimation with constrained zonotopes is formulated. First it is applied to fault detection of a lithium-ion cell, then the algorithm is extended for the case of battery packs. Finally joint state and parameter estimation is performed by adapting the algorithm. Results are presented comparing constrained zonotopes with other set-based methods, such as intervals and zonotopes.

- To conclude, in Chapter 6 some concluding remarks are given, together with possible future developments.

Chapter 2

Lithium-ion Batteries

The aim of the present Chapter is to give a general overview on lithium-ion batteries. First the technological development and historical background that brought to the wide usage of lithium-ion batteries is analyzed, then the working principle and electrochemical processes are detailed, followed by an introduction to Battery Management Systems. To conclude, the mathematical models representing lithium-ion batteries used in this thesis are presented.

2.1 Historical background

Li-ion batteries available today on the market are the result of decades of research and innovation. During the XVIII century, experiments were being developed regarding the conservation of electrical charge. One of the first examples is the Leiden jar, that, like a capacitor, was able to physically store an high-voltage charge and release it [Dubal et al., 2016]. The drawback of such early instruments was the the charge would release all at once, being therefore useless in practical applications. A battery instead is a device able to exploit a chemical reaction and transform its potential into electrical energy, overcoming this problem and allowing a continuous flow of current. The first prototype of such accumulator is attributed to Alessandro Volta, that built it in 1799 by intercalating zinc and copper discs, separated by cardboard soaked in a sodium chloride solution (highly conductive) [Pancaldi, 1990]. In Figure 2.1 the picture of a prototype is presented.

Following this discovery from A. Volta, other scientists tackled the problem of electrical energy storage. In 1859 Gaston Planté built the first prototype of a lead-acid battery that could be recharged instead of being one-time use. Non-rechargeable batteries are referred to as "primary", while rechargeable as "secondary". He was able to obtain this result by exploiting the reversibility of chemical reactions that could be triggered by an inverse current [Kurzweil, 2010]. The drawback was that the electrolyte was made of sulfuric acid, which is dangerous and inconvenient to use. The first "dry cell" was obtained in 1886 by Carl Gassner, who mixed plaster with ammonium chloride to obtain

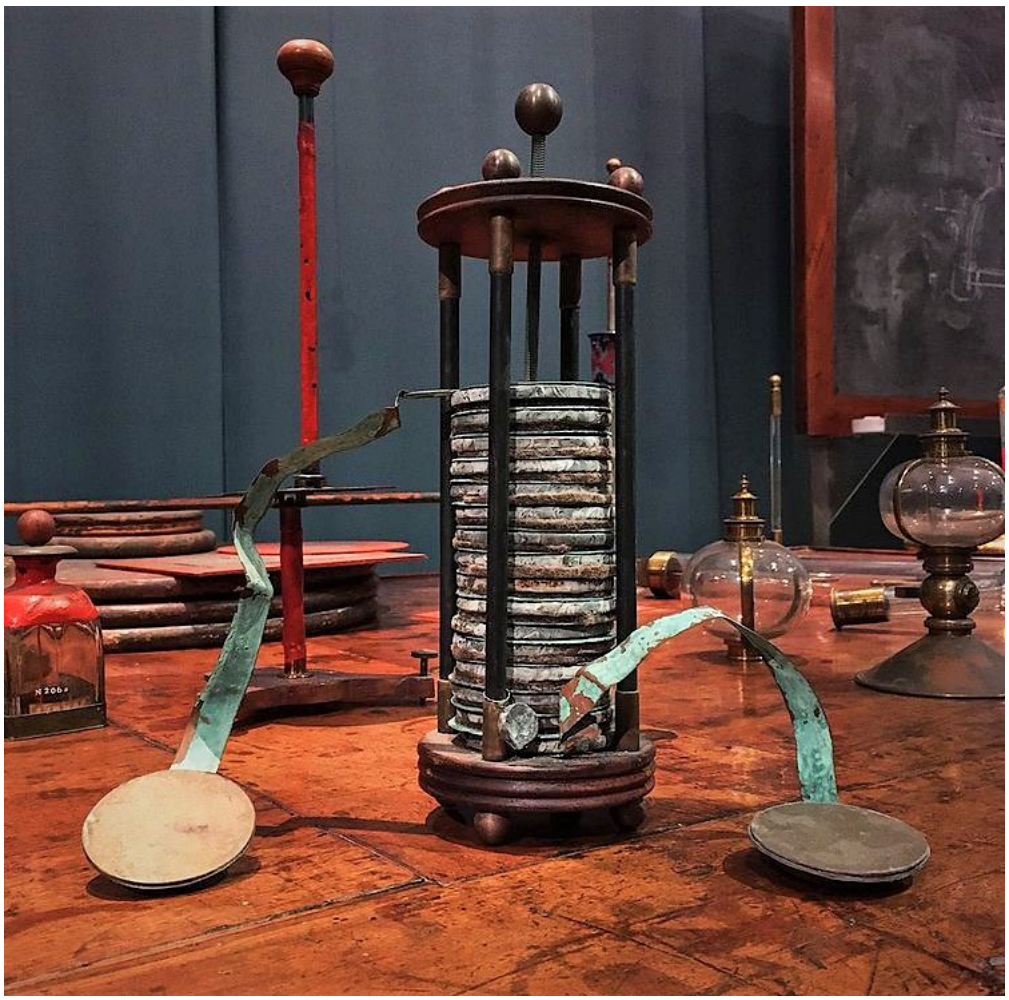


Figure 2.1: Example of the battery built by A. Volta, preserved in the Museum of Electrical Technology, Pavia, Italy.

a paste that would function as electrolyte, while the cathode and the anode were respectively made of manganese dioxide and zinc. This cell came to be known as zinc-carbon and is still available as of today [Jain et al., 2020]. In 1899 the Swedish scientist Waldemar Jungner proposed the first version of the nickel-cadmium rechargeable battery with an alkaline electrolyte (potassium hydroxide), widely used since then [Jain et al., 2020]. After this period, there

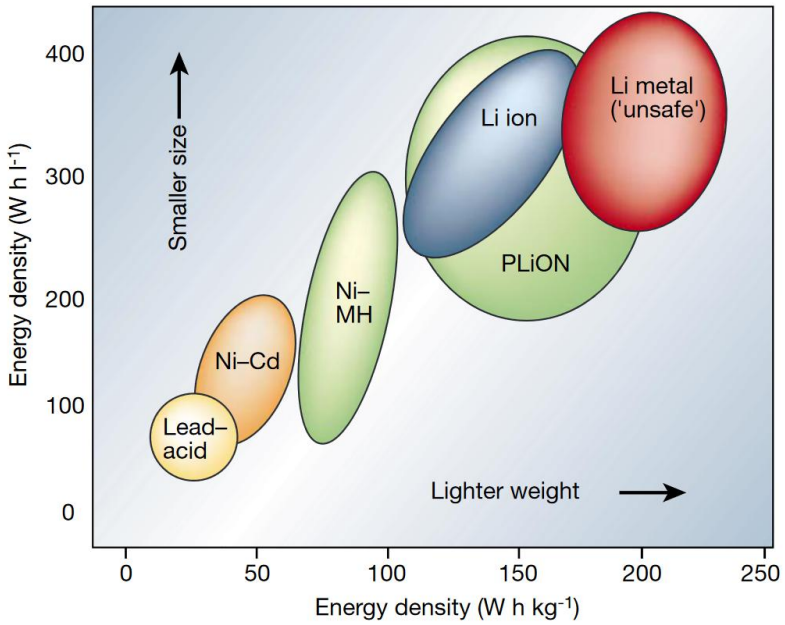


Figure 2.2: Comparison of volumetric and gravimetric energy density for different battery chemistry [Tarascon and Armand, 2001].

was not a real necessity for further improvements, as most electrical applications were satisfied by the available primary cells. It was only in the 1960s that the consumer electronics market started to grow, hence the necessity of improving portable energy sources. The main limitation was energy density: the amount of energy stored per unit of volume or weight was insufficient. Research then shifted towards the choice of lithium as base element, being the metal with the lowest density and the greatest energy-to-weight ratio. Moreover, it easily loses its most external electron, facilitating ionization.

The breakthrough came in 1974 from Stanley Whittingham, who used titanium disulfide as cathode material with a lithium anode. The porous structure of the cathode would allow it intercalate lithium ions without significant changes to its crystal structure. Still, the presence of metallic lithium would make the battery prone to catch fire. Later works from Koichi Mizushima and John Goodenough aimed at replacing the cathode with lithium cobalt oxide, that would not solve the flammability problems but was more stable and had a higher voltage [Mizushima et al., 1980]. The cause of fires in the metallic lithium anode were attributed to the formation of dendrites, which caused short-circuits. This was solved in 1987 by Akira Yoshino by using an anode

made of "soft carbon". Using this design, Sony began commercial production of rechargeable Li-ion batteries in 1991 [Scrosati, 2002].

Since the 1990s, the consumer electronics market started growing exponentially, and consequentially the request for portable power sources. Research has been focused on Li-ion batteries, due to their optimal characteristics, such as high energy density, low self-discharge rate and good safety level. In Figure 2.2 a comparison of properties (volumetric and gravimetric energy density) for different types of batteries is shown. Also of interest is the reduction in prices that happened after the widespread of Li-ion batteries. This trend is shown in Figure 2.3, where the prices per kWh are compared with the global installed capacity.

In 2019, Yoshino, Goodenough and Whittingham were awarded the Nobel Prize in Chemistry for their contribution in the development of Li-ion batteries.

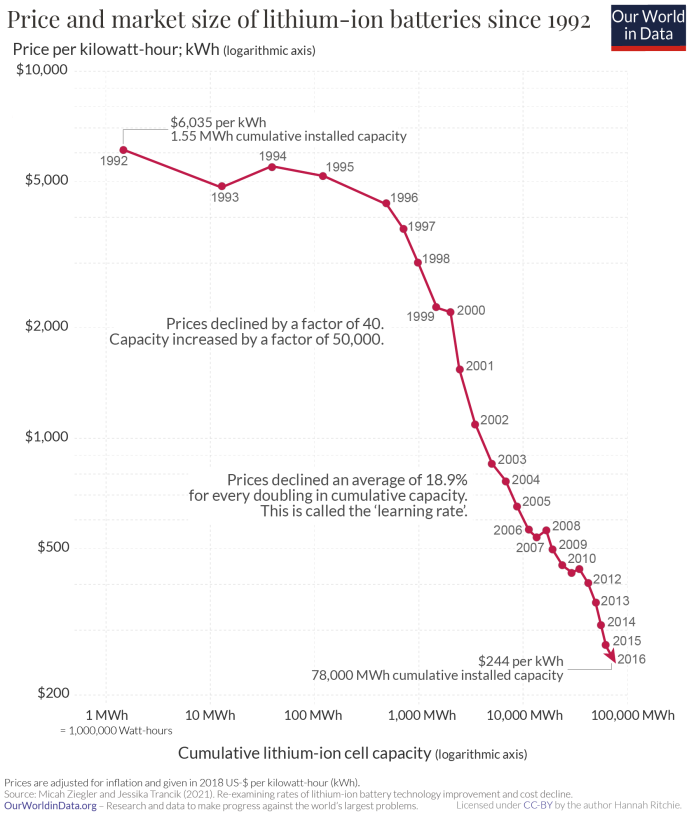


Figure 2.3: Decline in battery prices and market expansion (ourworldindata.org adapted from [Ziegler and Trancik, 2021]).

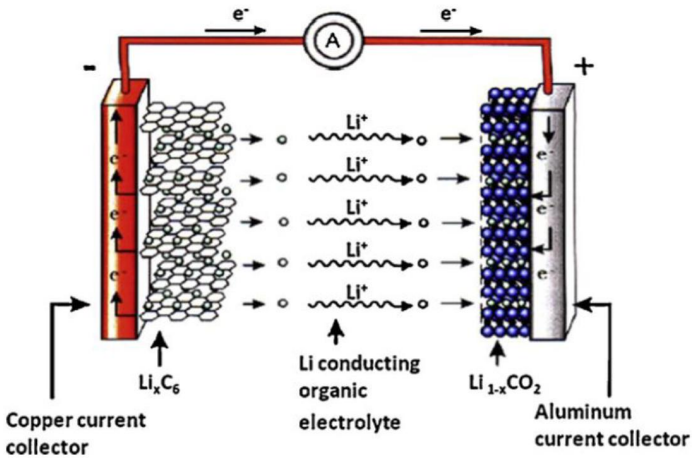


Figure 2.4: Li-ion battery cell schematic representation (discharge) [Scrosati, 2011].

2.2 Li-ion batteries technology and working principle

In this section, the working principles of a generic Li-ion cell are summarized. Note that different elements can be used, however the functioning principle is basically the same. The three components of a Li-ion cell are the anode, cathode and separator; everything is submerged in an electrolyte. The anode is the negative electrode and is usually made of graphite, exhibiting a layered structure capable of intercalating ions. The cathode, or positive electrode, is made of a lithium-based metal (e.g lithium cobalt oxide $LiCoO_2$), while the electrolyte is usually a lithium salt solution (e.g. $LiPF_6$). The separator is an insulating material with a porous structure, used to electrically isolate the electrodes. Note that anode and cathode are not connected directly to the load but rather through current collectors usually made of copper or aluminum. The fundamental working principle is the insertion and extraction of Li-ions from the electrodes, that simultaneously creates a flow of electrons. During the charging phase, when a power supply is connected to the collectors of the cell, ions move from the cathode to the anode flowing through the separator, while electrons flow through the external electrical circuit to balance the reaction. Ions then intercalate in the layered structure of the anode and are stored. The opposite happens when a load is connected to the cell and the cell discharges its potential: ions move from the anode to rejoin the cathode. Energy is released in the external circuit that is then able to power the connected load. The discharging phase is schematized in Figure 2.4, where all the previously described elements are represented.

2.2.1 Aging mechanisms

Although Li-ion batteries are less prone to aging effects with respect to their predecessors, their lifetime is still limited by degradation phenomenons. These mainly consist of a capacity fade and film resistance grow [Yoshida et al., 2006]: Solid Electrolyte Interface (SEI) layer growth is perhaps the single most important cell degradation phenomenon and leads to the growth of the internal cell resistance. At first, when the battery is freshly produced, this is a beneficial effect, as it improves the stability between the anode and the electrolyte. During nominal operations, SEI growth continues to happen, however it can be accelerated in presence of elevated currents and temperatures. Together with SEI growth, in the active carbon elements, lithium corrosion leads to capacity fade. On the other hand, when temperatures are low, lithium metal deposits on the surface of the anode. It is known as lithium plating and it can be particularly dangerous due to the risk of dendrite formation, that could grow and puncture the separator, causing a short-circuit between anode and cathode. Degradation dynamics mainly affect the anode, however it is possible to observe in the cathode some oxidation, causing surface film formation. These are minor phenomenons and have a limited impact on the cell lifespan.

2.2.2 Cell form factor

Depending on the application, Li-ion cells can be packed with different form factors. When a cell is used in a battery pack, the objective is to minimize the inert space between cells, taking however into account thermal dynamics and heat dissipation. For applications that instead require a single cell, such as mobile devices, the cell has to be shaped accordingly to the needs of the application [Maiser, 2014]. In the following we report the most common types of cell formats, also pictured in Figure 2.5:

- **Cylindrical** - it was the first format adopted for Li-ion cells (and for other types of electrochemical accumulators) and still used today. They have the big advantage of being mechanically stable that can resist without deformations to internal pressures. The most common type of cylindrical Li-ion cell format is the so-called 18650: 18 denotes the diameter, 65 indicates the length in mm. The capacity depends strictly in the chemistry and materials used and it ranges from 1.2 Ah to 3 Ah. Cathode, anode and separator are cut in stripes and rolled in a cylindrical container. When packing this format in a battery pack, the density is low, however a cooling liquid, typically air, is able to easily circulate.

- **Prismatic** - this type of cell format was introduced recently to adapt to the new requirements of portable devices. In particular, laptops and smartphones are becoming thinner and thinner, and the battery is the main obstacle. It is clear that a cylindrical cell is not suitable for this purpose, therefore box-shaped formats were developed. These are built

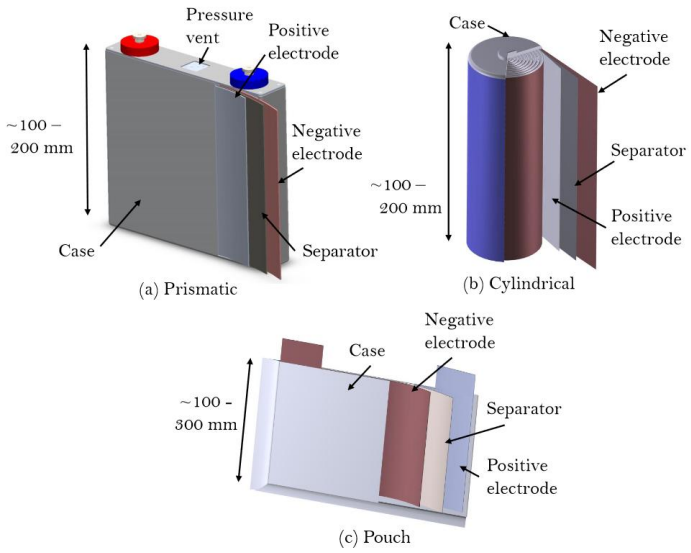


Figure 2.5: Types of the most used cell packaging [Lidbeck and Syed, 2017].

by cutting in long stripes the components and pressing them in the container. Note that in a single container can be inserted multiple cells. Mechanical stress is induced in the points where the material is bent, introducing potential problems in the electrode coating or electrolyte distribution in case of mistakes or inaccuracies. There is no standard for their format and size. For their form factor, density can be very high.

- **Pouch** - this type of cell is the one offering the highest possible density due to layering and stacking of cathodes, separators and anodes. Moreover, the external container is made of flexible foil, reducing weight. They are particularly suitable for high-power and high-energy applications. Pouch cells require careful current and temperature management to avoid swelling of the package that has no mechanical resistance.

2.3 Battery Management System

The correct management of a Li-ion battery is not a trivial task and appropriate devices have to be used to accomplish this task. The Battery Management System (BMS) serves exactly this purpose, performing various tasks. These are fundamental both to ensure a safe operation and use the batteries optimally. The limiting factor during BMS operations is the limited quantities of available measurements: from the complicated electrochemical system of a cell, only voltage, current and temperature measurements can be extracted. Moreover, charging and discharging profiles can be managed in order to reduce stress on the battery and minimize aging. The main tasks for which the BMS is responsible are:

- **SOC estimation** - SOC is a fundamental quantity in the management of a Li-ion battery: it indicates the level of charge remaining. Using an analogy, it can be seen as the fuel level in an internal combustion engine car. A definition can be given

$$\text{SOC} = \frac{C_{stored}}{C_{total}}, \tag{2.1}$$

where C_{stored} is the quantity of charge stored at any given time, while C_{total} is the maximum charge that can be stored. It is a dimensionless quantity and is usually expressed in percentage or a value in [0 1]. Note that this definition is not complete and formally correct, it is used only to give an idea of this quantity. The issues are that C_{total} is often uncertain even at the beginning of the battery lifespan and it changes over time: the complicated underlying electrochemical model makes the evaluation of C_{total} difficult.

The biggest challenge with SOC is that there is no way to measure it directly and it can only be estimated using the available measures. This is perhaps the most important task performed by the BMS as, for example, it indicates the range left on an Electric Vehicle (EV). It has been widely researched in the literature with the first proposed method being Coulomb counting: it is based on the time integral of the current flowing through the battery to compute the charge moving through the load or stored in the battery (depending if the current is positive or negative) [Ng et al., 2009, Truchot et al., 2014, Lashway and Mohammed, 2016]. A possible formula for SOC is given by

$$\text{SOC}(t) = \text{SOC}(t_0) - \frac{\upsilon_c}{C_n} \int_{t_0}^{t} I_{cell}(t)dt, \tag{2.2}$$

where t_0 is the initial time instant, υ_c is the coulombic efficiency, C_n the nominal capacity and $I_{cell}(t)$ the instantaneous current. The critical issue with this methodology are the unavoidable measurement errors on the current and the unknown initial condition for the SOC. A detailed overview of SOC estimation methods was done in [How et al., 2019]. Some of them are reported in the following. *Look-up table* methods establish a relationship between SOC and external measurable characteristics (e.g. impedance). Extensive experiments that consist in charging and discharging the battery using appropriate profiles are run beforehand to map a table, which can be later used in online estimation [Kim et al., 2012, Xing et al., 2014, Shen et al., 2017]. *Data-driven* methods rely on the use of machine learning to establish a relationship between measurable battery quantities and SOC instead of experiments. Different learning algorithms can be used, such as neural networks [Hannan et al., 2018], deep learning [Yang et al., 2019, Chemali et al., 2017], support

vector machine [Antón et al., 2013, Anton et al., 2013] and fuzzy logic [Sheng and Xiao, 2015]. *Model-based* approaches require the definition of a dynamical model of the electrochemical system; the states can be estimated using filtering algorithms such as the Kalman filter [Yu et al., 2017]. Different models can be used for the Li-ion cell: they are analyzed in detail in Section 2.4.

- **SOH estimation** - State of Health (SOH) is a quantity that indicates the health of the battery in terms of capacity lost during operations. Similarly to SOC, it is difficult to give a formal and universal definition, however conceptually it is possible to write

$$\text{SOH} = \frac{C_{cell}}{C_{total}}, \tag{2.3}$$

where C_{cell} is the total available capacity at any given moment. According to this definition, when the cell is newly manufactured, it will result SOH $= 1$ and this value will decrease as the battery is used and deteriorates. Once again, the challenge is having and accurate estimation of both C_{cell} and C_{total}, for which the same problems as before arise. Also the same methods used for SOC can be used to evaluate SOH. Although less important than SOC, accurate knowledge of SOH is important to estimate the remaining useful lifespan of the battery; usually when a battery loses $20\% - 30\%$ of its original capacity it is considered to be exhausted for the application it was designed for. However note that exhausted cells can still be reused for example in smart grids energy storage, where energy density is not of primary importance. This application is ideal for EV batteries, that originally have a very high capacity and energy density. For smaller batteries (e.g. smartphones and laptops), the most convenient disposing solution at the end of their useful life is recycling: although much work has still to be done, newly developed techniques are able to recover most of the materials in the cell [Wu et al., 2022].

- **Fault detection** - in rare occasions it might happen that the battery encounters some configurations in which continuing operations is not safe. It is the BMS's task to identify potentially dangerous situations and act accordingly. Dangerous situations might include overcurrents, over-voltage, under-voltage, over-charge, over-discharge, overtemperature and short-circuits. If a fault gets detected, the BMS can for example decide to disconnect the battery from the load (or from the charger), to prevent worst situations such as fires or explosions; in case of thermal faults instead, it can decide to act on the cooling system. Note that for high-power applications such as EVs, the presence of fault detection algorithms and procedures is mandatory and it requires the compliance with the norm ISO 26262. In addition to fault detection,

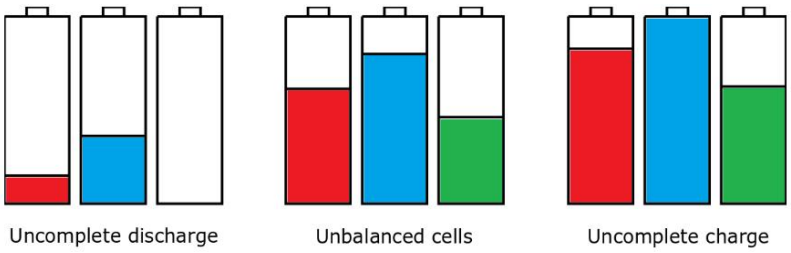


Figure 2.6: Example of unbalanced cells [Pozzi et al., 2020].

some BMS may also perform fault identification and isolation, trying to understand what type of fault has happened and isolate it to continue operations (even at reduced performance); this task however requires also actuators in the battery pack, for example to exclude a section.

- **Cell balancing** - as mentioned before, battery packs can be composed of up to thousands of cells, connected in a suitable combination of series and parallel. A mismatch in SOC or capacity of the single cells could lead to a degradation of the overall performance. For example, during discharge when a single cell reaches the threshold set for considering the battery empty, the whole battery is considered empty, even if there is still charge left in other cells (see as example Figure 2.6). The opposite can happen during charging, where the full capacity of the battery cannot be exploited. Different algorithms can be used to perform this task, refer to [Hemavathi, 2020, Pozzi et al., 2020].

- **Charge and discharge control** - from the user point of view, charging time should be as short as possible. However, having high currents and temperatures is the main source of degradation mechanisms, therefore there needs to be a trade-off between charging speed and aging. Besides aging, also safety is a concern with respect to charging currents: having high currents and voltages (and therefore powers) can be dangerous and should be carefully monitored. The BMS can provide an optimal charging profile based on the environmental conditions and taking into account other provided requirements.

In conclusion, the BMS is in charge of all the operational aspects of a battery. It is present in every application, however its functions can vary. Simpler (and cheaper) ones may only perform SOC estimation, while complete (and expensive) BMSs include all of the tasks listed above.

2.4 Li-ion cell models

The mathematical model of a Li-ion battery can be useful also for many different purposes. For instance, it can be used to simulate the battery behaviour

in different conditions: it is possible to test different charging current profiles, reproduce faults or test in a virtual environment new prototypes. All of these are useful to avoid the costs, lengthiness and eventually dangers of physical experiments on real batteries. Moreover, if any criticality arises during simulations, it is possible to focus the experiments and eventually reproduce particular situations. As explained in Section 2.3, the BMS benefits greatly from having a mathematical model available in order to improve estimation, control and diagnosis. In the literature many different models have been proposed, that can be divided mainly in two categories: equivalent circuit models and electrochemical ones. Both are described in detail in the following.

2.4.1 Equivalent Circuit Models

ECM make use of electrical components to describe the electrochemical dynamics of the Li-ion cell. Initially, ECMs were developed by performing a frequency-domain analysis of the current-voltage relationship to define an impedance-based model. Thévenin models substituted this approach [Rahimi-Eichi et al., 2013]. These are generally very simple models, with few states represented by a set of Ordinary Differential Equations (ODE). These are therefore very inexpensive to simulate and particularly suitable for real-time applications. Early developed models relied on a large linear capacitor whose voltage and accumulated charge would represent SOC and Electromotive Force (EMF).

Remark 1 *EMF is an intrinsic characteristic of the Li-ion cell and represents the terminal voltage when the battery is not connected to a load. For this reason sometimes it is also called Open-Circuit Potential (OCP).*

However, having only linear dynamics limits the capabilities and applicability of this type of models, as the relationship between SOC and EMF is nonlinear. As mentioned in Section 2.3, one possible solution would be to derive a SOC/EMF look-up table that in the ECM can be introduced as a controlled voltage source. Doing so, also allows to map the differences in the SOC/EMF relationship during charge and discharge due to the hysteresis effect [Eichi and Chow, 2012]. A complete overview of all the available ECM is given in [He et al., 2011], the ones relevant in this thesis are listed and detailed below.

- **Internal resistance model** - one of the simplest models possible, it consists only of the variable voltage source and an internal resistance R_{int}. This latter represents the ohmic losses inside the cell due to resistance in ion diffusion in the electrolyte, the SEI layer and the losses in electrodes and current collectors. The terminal voltage is given by

$$\begin{cases} \dfrac{d\mathrm{SOC}(t)}{dt} = -\dfrac{I_{cell}(t)}{3600 C_{cell}}, \\ V_{cell}(t) = V_{EMF}(\mathrm{SOC}(t)) + R_{int} I_{cell}(t). \end{cases} \tag{2.4}$$

The current flowing through the cell is denoted with I_{cell}, by convention is negative if the cell is charging. C_{cell} is the capacity of the cell indicated in Ah, thus explaining the presence of the factor 3600. Time dependency is included to eventually take into account the possibility of it changing over time due to aging effects. This model is represented in Figure 2.7.

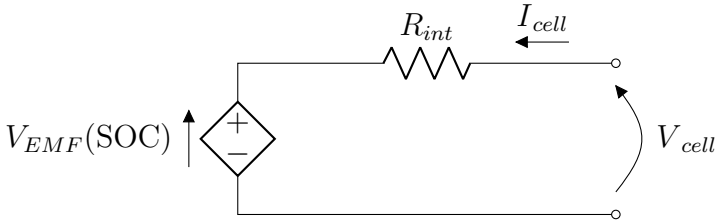


Figure 2.7: Internal resistance model.

- **Thévenin model** - in addition to the controllable voltage source and the internal resistance, in this model a pair of parallel resistor-capacitor (RC) is present; this represents the dynamical characteristics of the cell. Below in Figure 2.8 is the schematic diagram. The equations describing

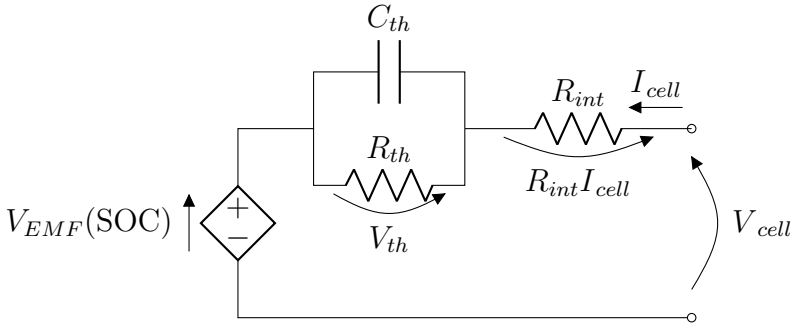


Figure 2.8: Thévenin model circuit.

the system are

$$\begin{cases} \dfrac{dV_{th}(t)}{dt} = \dfrac{-1}{R_{th}C_{th}}V_{th}(t) + \dfrac{1}{C_{th}}I_{cell}(t), \\ \dfrac{d\text{SOC}(t)}{dt} = -\dfrac{I_{cell}(t)}{3600C_{cell}}, \\ V_{cell}(t) = V_{EMF}(\text{SOC}(t)) + V_{th}(t) + R_{int}I_{cell}(t), \end{cases} \tag{2.5}$$

where R_{th} and C_{th} represent respectively the polarization resistance and a capacity used to describe the transient response during charge and discharge.

- **Dual polarization model** - to improve the performance of the Thévenin model, the Dual Polarization (DP) model was developed. Based on practical experiments, a polarization can be observed and the use of a single RC block is not able to accurately capture the difference between concentration and electrochemical polarization, especially at low and high levels of SOC (start of charge and discharge). The Thévenin model is then improved by adding a second RC parallel block. The circuital scheme for

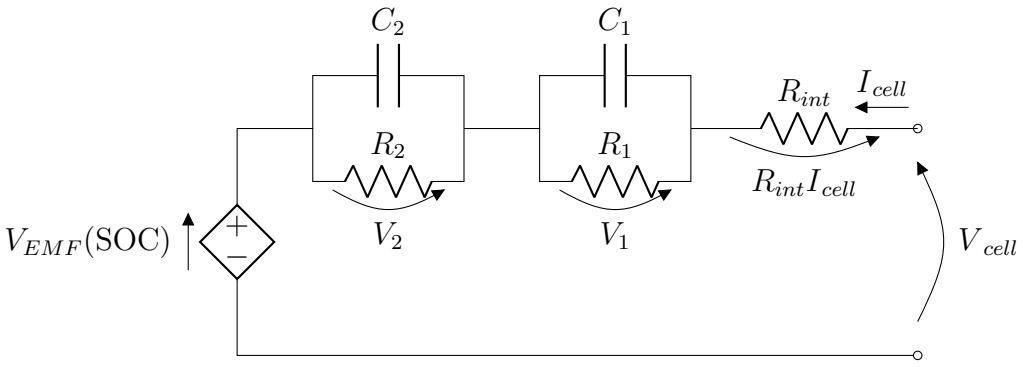


Figure 2.9: Dual polarization model circuit.

the DP model is depicted in Figure 2.9. Terminal voltage is given by

$$V_{cell}(t) = V_{EMF}(\mathrm{SOC}(t)) + V_1(t) + V_2(t) + R_{int}I_{cell}(t), \tag{2.6}$$

where $V_1(t)$ and $V_2(t)$ are the voltage drops on the RC parallel blocks, $R_{int}I_{cell}(t)$ is the ohmic voltage drop. The voltage dynamics in the RC blocks are

$$\begin{cases} \dfrac{dV_1(t)}{dt} = -\dfrac{1}{R_1C_1}V_1(t) + \dfrac{1}{C_1}I_{cell}(t), \\ \dfrac{dV_2(t)}{dt} = -\dfrac{1}{R_2C_2}V_2(t) + \dfrac{1}{C_2}I_{cell}(t). \end{cases} \tag{2.7}$$

From a practical standpoint, the one RC block represents fast polarization dynamics while the other one represents slower dynamics. SOC is calculated as

$$\frac{d\mathrm{SOC}(t)}{dt} = -\frac{I_{cell}(t)}{3600C_{cell}}. \tag{2.8}$$

In addition to electrochemical dynamics expressed with an electric circuit analogy, it is also possible to include simplified thermal and aging dynamics in the ECM. These can be used in combination with any ECM just presented.

Thermal dynamics - the considered model is a two-state lumped thermal model as presented in [Lin et al., 2014a, Hu et al., 2015]. The Li-ion cell is considered to be cylindrical, as states we consider the core temperature $T_c(t)$ and the surface temperature $T_s(t)$ as depicted in Figure 2.10. Note that these are considered to be uniform both across surface and core; in reality

this might not always be the case, as some parts of the core might be at a higher temperature than others or the surface might be exposed differently to the coolant fluid. We consider that the surface of the cell is in contact with the coolant at temperature $T_f(t)$ and they exchange heat through convection. It is not important the type of fluid that circulates around the cell, whether it is air, water or a specific-purpose fluid, the parameter that changes in this model is the thermal resistance R_u. Between the core and the surface of the cell instead heat transfers through convection and is modeled with the thermal resistance R_c. Note that in this model there are no dynamics associated to $T_f(t)$, however time dependency has been maintained to account for the case in which the temperature changes but the value is known as input. The thermal

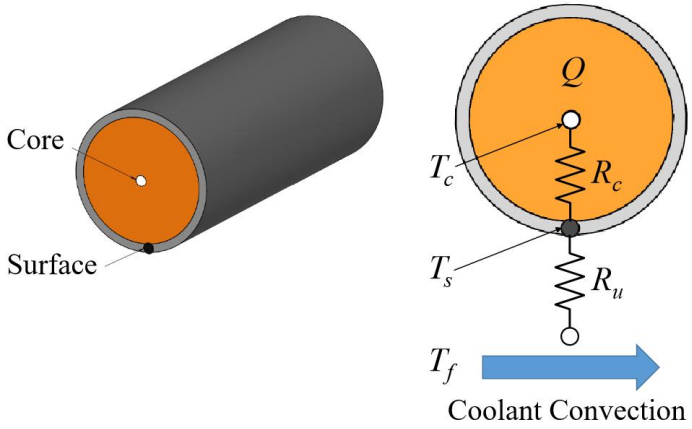


Figure 2.10: Two-state lumped thermal model [Hu et al., 2015].

dynamical model corresponds to

$$\frac{dT_c(t)}{dt} = \frac{T_s(t) - T_c(t)}{R_c C_c} + \frac{Q(t)}{C_c}, \tag{2.9}$$

$$\frac{dT_s(t)}{dt} = \frac{T_f(t) - T_s(t)}{R_u C_c} + \frac{T_s(t) - T_c(t)}{R_c C_s}, \tag{2.10}$$

where C_c is the thermal capacity of the core of the cell and C_s is the surface thermal capacity, both measured in JK^{-1}. Q is the heat generation computed using Joule's law

$$Q(t) = |I_{cell}(t)|| \left(V_{cell}(t) - V_{EMF}(\text{SOC}(t))\right)|. \tag{2.11}$$

Aging dynamics - because ECMs are very simple models, it is hard to define an accurate aging model, especially because the quantities involved are very speculative and none can be measured, in contrast to thermal dynamics where the temperature is a measurable quantity. In [Wang et al., 2011] a

formulation is provided with the aim of estimating SOH

$$\frac{d\text{SOH}(t)}{dt} = -\frac{|I_{cell}(t)|}{2N_{cycle}(c_r, T_c)C_{cell}}, \tag{2.12}$$

where $N_{cycle}(c_r, T_c)$ is the number of cycles until the end of useful life for the battery; it is function of the core temperature T_c and Charging Rate (C-rate) c_r. It is defined as

$$N_{cycle}(c_r, T_c) = \frac{3600A_{tol}(c_r, T_c)}{C_{cell}}, \tag{2.13}$$

with A_{tol} being the total discharged throughput (measured in Ah) given by

$$A_{tol}(c_r, T_c) = \left[\frac{20}{M_{exp}(c_r)\exp\left(\frac{-E_a E_a(c_r)}{RT_c(t)}\right)}\right]^{\frac{1}{z_p}}. \tag{2.14}$$

Here, it is assumed that a capacity loss of 20% indicates the depletion of the battery, therefore it will result that SOH = 0 when the battery is at 80% of the original capacity. R is the ideal gas constant, $M_{exp}(c_r)$ is a pre-exponential factor as a function of C-rate, z_p the power-law factor and $E_a(c_r)$ the activation energy in $[J/mol]$ expressed as

$$E_a(c_r) = 31700 - 370.3c_r. \tag{2.15}$$

More details on this topic can be found in [Wang et al., 2011].

Having defined electrical, thermal and aging dynamics, it is possible to combine them in a single model and observe how they are coupled together and connected together. See Figure 2.11 for reference. The input of this model

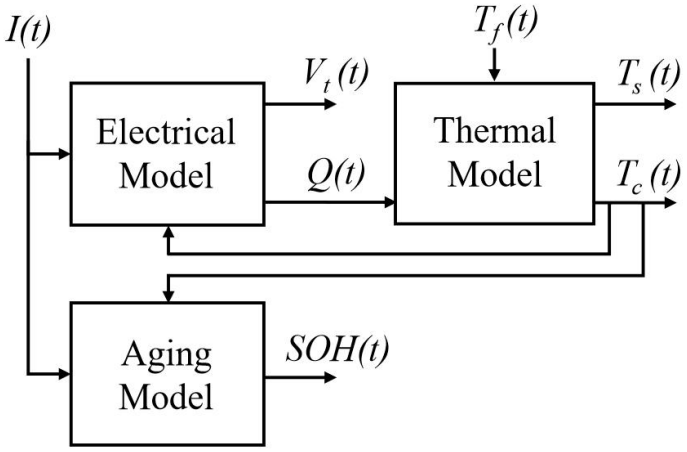


Figure 2.11: Coupling between electrical, thermal and aging dynamics in an ECM [Hu et al., 2015].

is considered to be the current flowing in the cell $I_{cell}(t)$, while the outputs are

the terminal voltage $V(t)$ and the surface temperature $T_s(t)$, typically the two measurable quantities. The cooling fluid temperature is also assumed to be known and provided. Note that any of the electrical circuits presented above can be used in this scheme. Because of its simplicity, as there are at most six ODE, it is particularly suitable for real time simulation purposes, however it lacks precision, especially in the aging model. Moreover internal dynamics of the cell are ignored. This is the reason why electrochemical models have been developed.

2.4.2 Electrochemical Models

Electrochemical Models (EM) have the greatest detail possible for a Li-ion cell. They comprise the chemical reactions happening inside the cell in great detail. The drawback of having accurate models is having to deal with high computational time and memory usage, as the dynamics include Partial Differential Equations (PDE) and Differential Algebraic Equations (DAE). EMs prove extremely useful for manufacturers during the design process to test and optimize the chemistry in a simulation environment. The first EM proposed is the Doyle-Fuller-Newman (DFN) model proposed in 1993 [Doyle et al., 1993], also known as Pseudo-two dimensional (P2D) model; it one of the mostly used models in the literature. It is also one the most complex models, requiring a high computational power to simulate. Moreover, the parameters of the DFN model are difficult to retrieve, as they either need to be physically measured with advanced instruments by taking apart the cell or estimated using appropriate techniques. To address the identifiability, observability and computational complexity issues [Moura, 2015], many works have focused on creating reduced-order models derived from the P2D, looking for a trade-off between the aforementioned issues while providing the best possible representation of the internal phenomena of the cell. Amongst these, it is worth to mention the Single Particle Model (SPM) [Santhanagopalan et al., 2006], that describes the electrodes as single particles. The SPM has a lower computational cost with respect to the DFN model, however it lacks some quantities that can be useful, such as electrolyte, thermal and aging dynamics. [Moura et al., 2016] proposes the SPMe, which includes the electrolyte dynamics; [Perez et al., 2017] instead adds to SPMe also thermal dynamics, and will be denoted as SPMeT. In the following, both the P2D model and the SPMeT are presented in detail.

- **P2D model** - this model consists of tightly coupled nonlinear Algebraic Partial Differential Equations (PDAE), used to represent the conservation of charge and mass in the Li-ion cell. The two dimensions considered for this model are a longitudinal and a radial one. Consider Figure 2.12, where a schematic representation of the cell is provided. Note that the flow of electrons and Li-ions is what happens during discharging. The cell is divided in three region along the z axis. In the electrodes, Li-ions exist in two phases: in the electrodes they are intercalated in the solid

material, represented by spheres with radius R_s (that in principle can be different for the anode and the cathode), while in the electrolyte, Li-ions exist only in a dissolved state. The radial dimension along the radius of the particles is indicated with r. In the separator instead, they exist only dissolved in the electrolyte.

The P2D model is governed by four coupled PDEs. Denoting with δ_n the length of the negative electrode and with δ_p the length of the positive electrode, it is possible to write Fick's law for Li-ion concentration in the solid phase $c_s(r, z, t)$

$$\frac{\partial c_s(r,z,t)}{\partial t} = \frac{D_s}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial c_s(r,z,t)}{\partial r}\right). \tag{2.16}$$

This relationship is valid only in the solid phase, therefore for $z \in [0, \delta_n] \cup [L_c - \delta_p, L_c]$, with L_c being the total length of the cell. Along Equation 2.16, boundary conditions also have to be defined

$$\left.\frac{\partial c_s(r,z,t)}{\partial r}\right|_{r=0} = 0, \quad \left. -D_s\frac{\partial c_s(r,z,t)}{\partial r}\right|_{r=R_s} = j_n(z,t), \tag{2.17}$$

with $j_n(z, t)$ being the net molar flux of Li-ions exiting the particle. These boundary conditions state that the concentration in the center of the particle does not change over time, while the concentration on the outside surface changes according to $j_n(z, t)$.

The Li-ion concentration in the electrolyte phase $c_e(z, t)$ for $z \in [0, L_c]$ is given by

$$\epsilon_e\frac{\partial c_e(z,t)}{\partial t} = \frac{\partial}{\partial z}\left(D_e{\epsilon_e}^{p_b}\frac{\partial c_e(z,t)}{\partial z} + \frac{3\epsilon_s\left(1-t_+^0\right)}{R_s}\right) \tag{2.18}$$

with boundary conditions

$$\left.\frac{\partial c_e(z,t)}{\partial r}\right|_{z=0} = 0, \quad \left.\frac{\partial c_e(z,t)}{\partial r}\right|_{z=L_c} = 0. \tag{2.19}$$

ϵ_e is the electrolyte phase volume fraction, D_e is the diffusion coefficient of lithium in the electrolyte phase, p_b the Bruggeman porosity exponent, ϵ_s the active material volume fraction and t_+^0 the transference number of Li-ions. Boundary conditions indicate that the electrolyte Li-ion concentration at $z = 0$ and $z = L_c$ is null.

The potential in the solid phase $\phi_s(z, t)$ is defined for $z \in [0, \delta_n] \cup [L_c - \delta_p, L_c]$ and is given by Ohm's law

$$\frac{\partial}{\partial z}\left(\sigma_c\epsilon_s\frac{\partial\phi_s(z,t)}{\partial z}\right) = \frac{3\epsilon_s F}{R_s}j_n(z,t), \tag{2.20}$$

with boundary conditions

$$\sigma_c \epsilon_s \frac{\partial \phi_s(z,t)}{\partial z}\bigg|_{z=0} = \frac{I_{cell}(t)}{A_{surf}}, \quad \sigma_c \epsilon_s \frac{\partial \phi_s(z,t)}{\partial z}\bigg|_{z=\delta_n} = 0, \tag{2.21}$$

$$\sigma_c \epsilon_s \frac{\partial \phi_s(z,t)}{\partial z}\bigg|_{z=L_c-\delta_p} = 0, \quad \sigma_c \epsilon_s \frac{\partial \phi_s(z,t)}{\partial z}\bigg|_{z=L_c} = \frac{I_{cell}(t)}{A_{surf}}, \tag{2.22}$$

that affirm that the solid phase potential derivative is null at the electrode-separator interface, while different from zero elsewhere.

The potential for the electrolyte phase $\phi_e(z,t)$ for $z \in [0, L_c]$ is given by

$$\frac{\partial}{\partial z}\left(\kappa_c {\epsilon_e}^{p_b} \frac{\partial \phi_e(z,t)}{\partial z} + \kappa_c {\epsilon_e}^{p_b} \nu_t \frac{2RT(t)}{F} \frac{\partial \ln c_e(z,t)}{\partial z}\right) = -\frac{3\epsilon_s F}{R_s} j_n(z,t), \tag{2.23}$$

where κ_c is the ionic conductivity, ν_t a thermodynamic factor and T the temperature of the Li-ion cell. The boundary conditions for $\phi_e(z,t)$ are

$$\frac{\partial \phi_e(z,t)}{\partial z}\bigg|_{z=0} = 0, \quad \frac{\partial \phi_e(z,t)}{\partial z}\bigg|_{z=L_c} = 0. \tag{2.24}$$

These state that the derivative of $\phi_e(z,t)$ is null at the extremes of the cell.

The above PDEs (2.16),(2.18),(2.20),(2.23) are coupled by a Butler-Volmer equation describing the electrochemical reaction rate $j_n(z,t)$ at the SEI interface. It equates to

$$j_n(z,t) = \frac{i_0(z,t)}{F}\left(\exp\left(\frac{\alpha_a F}{RT(t)}\eta_e(z,t)\right) - \exp\left(-\frac{\alpha_c F}{RT(t)}\eta_e(z,t)\right)\right), \tag{2.25}$$

defined only for the electrodes $z \in [0, \delta_n] \cup [L_c - \delta_p, L_c]$, while it is null in the separator $z \in [\delta_n, L_c - \delta_p]$. In (2.25), α_a and α_c are respectively the anodic and cathodic transfer coefficients, $\eta_e(z,t)$ is the electrode overpotential defined as

$$\eta_e(z,t) = \phi_s(z,t) - \phi_e(z,t) - U(z,t), \tag{2.26}$$

with $U(z,t)$ being the electrode equilibrium potential, given by a pre-defined function of the solid phase concentration at the SEI interface. Moreover, the exchange current density $i_0(z,t)$ is defined as

$$i_0(z,t) = k_0 c_e(z,t)^{\alpha_a} \left(c_s^{max} - \bar{c}_s(z,t)\right)^{\alpha_a} \bar{c}_s(z,t)^{\alpha_c}, \tag{2.27}$$

where k_0 is a kinetic constant, c_s^{max} the maximum solid Li-ion concentration and $\bar{c}_s(z,t)$ the solid concentration at the SEI interface.

At this point, it is possible to define the terminal voltage V_{cell} using the

quantities defined above

$$V_{cell}(t) = \phi_s(L_c, t) - \phi_s(0, t) - \frac{R_{cc}}{A_{surf}} I_{cell}(t). \tag{2.28}$$

In (2.28), R_{cc} is an empirical film resistance of the Li-ion cell.

Besides physical constants (e.g. F and R), many of the parameters of the DFN model have to be defined. Many works have tackled the problem of determining and estimating them [Valøen and Reimers, 2005, Georén and Lindbergh, 2004, Delacourt et al., 2011, Safari and Delacourt, 2010] and have shown that many of these properties are not constant but rather depend from the Li-ion concentration. Also to note is the fact that measuring some physical parameters is an expensive and time-consuming procedure, therefore it should be avoided as much as possible, making parameter estimation a fundamental task.

In addition to electrochemical dynamics, also thermal and aging dynamics can be defined. Note that these quantities are different from the ones defined of ECMs, because many more information are available. In particular, it is possible to define the thermal heat generation Q(t) as

$$Q(t) = Q_{cc}(t) + Q_{rev}(t) + Q_{irr}(t), \tag{2.29}$$

where $Q_{cc}(t)$ is the Ohmic heat generation, $Q_{rev}(t)$ the reversible heat generation and $Q_{irr}(t)$ the irreversible one. They are defined as

$$Q_{cc}(t) = \frac{R_{cc}}{a_{surf} V} I^2_{cell}(t), \tag{2.30}$$

$$Q_{irr}(t) = \frac{j_n(z, t)\eta_e(z, t) F a_{surf}}{L_c}, \tag{2.31}$$

$$Q_{rev}(t) = \frac{j_n(z, t) F a_{surf} S_e}{L_c}, \tag{2.32}$$

with a_{surf} being the electrodes specific interface surface area and S_e the entropy of the electrodes. Subsequently, the temperature can be defined using Q as

$$T(t) = \frac{1}{\rho_c C_c} Q(t). \tag{2.33}$$

In (2.33), ρ_c represents the density of the Li-ion cell and C_c the thermal capacity. The aging model under consideration describes the build-up of the SEI layer as follows

$$\frac{\partial R_f(t)}{\partial t} = -\frac{\tilde{v}_f}{\sigma_{sei}} j_s, \tag{2.34}$$

where R_f is the additional resistance that builds up in the SEI layer due

to aging, $\tilde{v}_f$ the cell molar volume, σ_{sei} the SEI layer conductivity and j_s the side reaction flux.

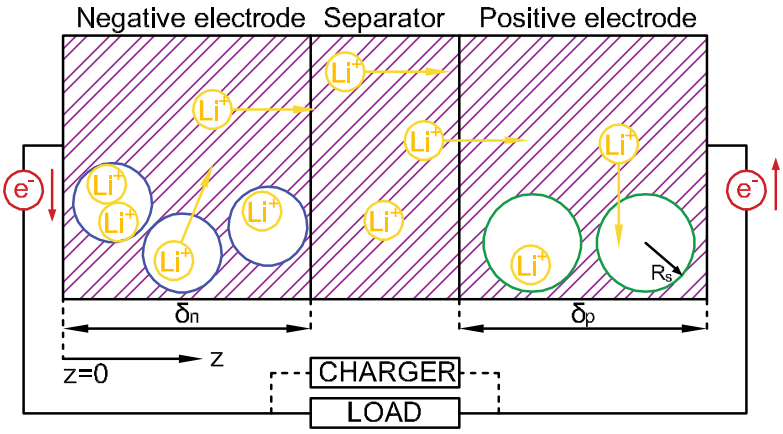


Figure 2.12: Li-ion cell schematic model during discharge.

- **SPMeT model** - this model consists of a simplification of the above P2D model. In particular, instead of modeling the electrodes as many particles, they are modeled as a single porous spherical particle. Doing so, it is possible to reduce Fick's law representing the Li-ion concentration into ODEs by exploiting a polynomial approximation for the ions concentration along the radial axis of each electrode [Subramanian et al., 2005]. Moreover, PDEs representing electrolyte dynamics are discretized using a Finite Volume Method (FVM) , as was done in [Torchio et al., 2016]. Thermal and aging dynamics are included as well.

  To define the SPMeT model, start by formulating the dynamics of the spatial mean of cathode average stoichiometry $\theta_p(t)$ as

$$\frac{d\theta_p(t)}{dt} = -\frac{\Delta\theta_p}{C_n} I_{cell}(t), \tag{2.35}$$

  with $\Delta\theta_p = {\theta_p}^{100\%} - {\theta_p}^{0\%}$, respectively fully charged and discharged cathode stoichiometry. Then, $\theta_n(t)$, anode average stoichiometry, can be defined similarly

$$\frac{d\theta_n(t)}{dt} = \frac{\Delta\theta_n}{C_{cell}(t)} I_{cell}(t), \tag{2.36}$$

  with $\Delta\theta_n = {\theta_n}^{100\%} - {\theta_n}^{0\%}$, respectively fully charged and discharged anode stoichiometry. Note that, according to the model used, see [Moura et al., 2016], the cathode average stoichiometry depends only on the nominal capacity of the cell, while the anode depends on the actual capacity of the cell; this is due to the fact that only the anode are considered to be affected by aging dynamics. The volume-averaged concentration fluxes

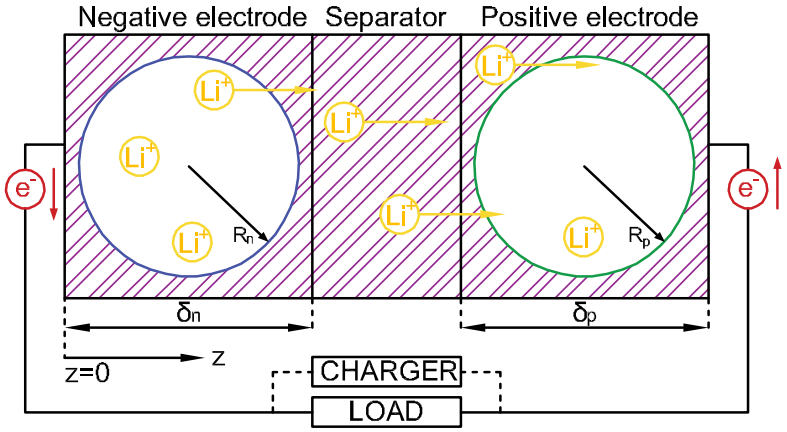


Figure 2.13: Schematic representation of the SPMeT model.

q_p and q_n are then defined

$$\frac{dq_p(t)}{dt} = -30\frac{D_s(T(t))}{{R_p}^2}q_p(t) - \frac{45\Delta\theta_p c_s^{max}}{6R_p C_{cell}(t)}I_{cell}(t), \tag{2.37}$$

$$\frac{dq_n(t)}{dt} = -30\frac{D_s(T(t))}{{R_n}^2}q_n(t) - \frac{45\Delta\theta_n c_s^{max}}{6R_n C_{cell}(t)}I_{cell}(t), \tag{2.38}$$

where R_p and R_n are the radii of the porous solid particles representing the electrodes, while $D_s(T(t))$ is the temperature-dependent solid diffusion coefficient calculated according to Arrhenius law

$$\psi(T(t)) = \psi_0 e^{\frac{-E_{a,\psi}}{RT(t)}}, \tag{2.39}$$

where ψ is a generic temperature-dependent parameter, ψ_0 its base value at a reference temperature and $E_{a,\psi}$ its correlated activation energy.

The same approach is applied to every temperature-dependent parameter. The electrodes surface stoichiometries $\theta_p^*(t)$ and $\theta_n^*(t)$ are respectively a function of $\theta_p(t)$, $q_p(t)$ and $\theta_n(t)$, $q_n(t)$

$$\theta_p^*(t) = \theta_p(t) + \frac{8R_p q_p(t)}{35c_s^{max}} - \frac{{R_p}^2\Delta\theta_p I_{cell}(t)}{105D_s(T(t))C_{cell}(t)}, \tag{2.40}$$

$$\theta_n^*(t) = \theta_n(t) + \frac{8R_n q_n(t)}{35c_s^{max}} - \frac{{R_n}^2\Delta\theta_n I_{cell}(t)}{105D_s(T(t))C_{cell}(t)}. \tag{2.41}$$

Using the above defined quantities, it is possible to calculate SOC(t)

$$\mathrm{SOC}(t) = \frac{\theta_n(t) - {\theta_n}^{0\%}}{{\theta_n}^{100\%} - {\theta_n}^{0\%}}. \tag{2.42}$$

Moving over to electrolyte dynamics, a set of PDEs describes the concen-

tration of the electrolyte $c_e(z,t)$ in the three sections of the cell [Moura et al., 2016], where z is the longitudinal variable ($0 \leq z \leq L_c$), L_c being the length of the Li-ion. Note that electrolyte concentration in the SPMeT model is defined differently in each section, electrolyte concentration in the cathode will be denoted with $c_{e,p}(z,t)$, in the separator with $c_{e,s}(z,t)$ and in the anode with $c_{e,n}(z,t)$

$$\epsilon_e \frac{\partial c_{e,p}(z,t)}{\partial t} = \frac{\partial}{\partial z}\left[{D_e}^{eff}(T(t)) \frac{\partial c_{e,p}(z,t)}{\partial z} \right] - \frac{1-t_+^0}{F A_{surf} \delta_p} I_{cell}(t), \quad (2.43)$$

$$\epsilon_e \frac{\partial c_{e,s}(z,t)}{\partial t} = \frac{\partial}{\partial z}\left[{D_e}^{eff}(T(t)) \frac{\partial c_{e,s}(z,t)}{\partial z} \right], \quad (2.44)$$

$$\epsilon_e \frac{\partial c_{e,n}(z,t)}{\partial t} = \frac{\partial}{\partial z}\left[{D_e}^{eff}(T(t)) \frac{\partial c_{e,n}(z,t)}{\partial z} \right] + \frac{1-t_+^0}{F A_{surf} \delta_n} I_{cell}(t), \quad (2.45)$$

with ${D_e}^{eff}(T(t)) = D_e(T(t)){\epsilon_e}^p$, calculated according to Equation (2.39), ϵ_e is the porosity and p Bruggeman's coefficient. The area of the cell is denoted with A_{surf}, F is Faraday's constant and t_+^0 the transference number. According to the FVM discretization proposed in [Torchio et al., 2016], each section gets divided into P non-overlapping volumes in the spatial domain along the z-axis and the system of PDEs is discretized, where the p-th volume ($p = 1, \ldots, P$) of any section has bounds $[z^{[\overline{p}]}, z^{[\underline{p}]}]$; the set of equations defined in (2.43) then becomes

$$\epsilon_e \frac{d{c_{e,p}}^{[p]}(t)}{dt} = \left[\frac{\tilde{D}_e(z,T(t))}{\Delta z_p} \frac{\partial c_{e,p}(z,t)}{\partial z} \right]\Bigg|_{z^{[\underline{p}]}}^{z^{[\overline{p}]}} - \frac{1-t_+^0}{F A_{surf} \delta_p} I_{cell}(t), \quad (2.46)$$

$$\epsilon_e \frac{d{c_{e,s}}^{[p]}(t)}{dt} = \left[\frac{\tilde{D}_e(z,T(t))}{\Delta z_s} \frac{\partial c_{e,s}(z,t)}{\partial z} \right]\Bigg|_{z^{[\underline{p}]}}^{z^{[\overline{p}]}}, \quad (2.47)$$

$$\epsilon_e \frac{d{c_{e,n}}^{[p]}(t)}{dt} = \left[\frac{\tilde{D}_e(z,T(t))}{\Delta z_n} \frac{\partial c_{e,n}(z,t)}{\partial z} \right]\Bigg|_{z^{[\underline{p}]}}^{z^{[\overline{p}]}} + \frac{1-t_+^0}{F A_{surf} \delta_n} I_{cell}(t), \quad (2.48)$$

where ${c_{e,p}}^{[p]}(t)$, ${c_{e,s}}^{[p]}(t)$ and ${c_{e,n}}^{[p]}(t)$ are the average electrolyte concentration in the p-th volume of the different sections. Moreover, Δz_p, Δz_s and Δz_n are the lengths of the single finite elements, computed as

$$\Delta z_p = \frac{\delta_p}{P}, \quad (2.49)$$

$$\Delta z_s = \frac{\delta_s}{P}, \quad (2.50)$$

$$\Delta z_n = \frac{\delta_n}{P}. \quad (2.51)$$

The number of finite volumes P can be in principle different for each

section, however here it is considered to be equal.

The electrolyte diffusion coefficients have to be adjusted in order to take into account the fact that on the border between the sections there are different materials. The harmonic mean is therefore used to calculate $\tilde{D}_e(z, T(t))$, refer to [Torchio et al., 2016] for more details.

The terminal voltage $V_{cell}(t)$ can be defined

$$V_{cell}(t) = U_p(\theta_p^*(t)) - U_n(\theta_n^*(t)) + \eta_p(t) - \eta_n(t) + \Delta\phi_e(t) + I_{cell}(t)R_f(t), \tag{2.52}$$

where $U_p(\theta_p^*(t))$ and $U_n(\theta_n^*(t))$ are the EMFs of the electrodes, calculated as a polynomial function of $\theta_p^*(t)$ and $\theta_n^*(t)$ respectively. $\eta_p(t)$ and $\eta_n(t)$ are the mean overpotentials for positive and negative electrodes, computed as

$$\eta_p(t) = \frac{2RT(t)}{F}\sinh^{-1}\left(\frac{-I_{cell}(t)}{2A_{surf}\delta_p a_p i_{0,p}(t)}\right), \tag{2.53}$$

$$\eta_n(t) = \frac{2RT(t)}{F}\sinh^{-1}\left(\frac{-I_{cell}(t)}{2A_{surf}\delta_n a_n i_{0,n}(t)}\right), \tag{2.54}$$

with a_p and a_n being the specific active surface area. The exchange current densities $i_{0,p}(t)$ and $i_{0,n}(t)$ are given by

$$i_{0,p}(t) = Fk_p(t)c_s^{max}\sqrt{c_{e,p}(t)\theta_p(t)(1-\theta_p(t))}. \tag{2.55}$$

In addition, $\Delta\phi_e(t)$, the electrolyte voltage difference, is calculated as

$$\Delta\phi_e(t) = \phi_e^{drop}(t) + \frac{2R}{F}(1-t_+^0)T(t)\ln\left(\frac{{c_{e,p}}^{[1]}(t)}{{c_{e,p}}^{[P]}(t)}\right), \tag{2.56}$$

where $\phi_e^{drop}(t)$ is the electrolyte voltage drop, approximated by

$$\phi_e^{drop}(t) \simeq -\frac{I_{cell}(t)}{2A_{surf}}\left(\phi_p(t) + 2\phi_{sep}(t) + \phi_n(t)\right), \tag{2.57}$$

in which

$$\phi_p(t) = \Delta z_p \sum_{p=1}^{P} \frac{2p-1}{{\epsilon_e}^{p_b}\kappa_e({c_{e,p}}^{[p]}(t), T(t))}, \tag{2.58}$$

$$\phi_{sep}(t) = \Delta z_s \sum_{p=1}^{P} \frac{1}{{\epsilon_e}^{p_b}\kappa_e({c_{e,s}}^{[p]}(t), T(t))}, \tag{2.59}$$

$$\phi_n(t) = \Delta z_n \sum_{p=1}^{P} \frac{2P-2p+1}{{\epsilon_e}^{p_b}\kappa_e({c_{e,n}}^{[p]}(t), T(t))}. \tag{2.60}$$

In (2.58), the function $\kappa_e(\cdot)$ indicates the temperature-dependent electrolyte conductivity for the p-th volume of each section, expressed as a nonlinear function of the electrolyte concentration.

Similarly to the DFN model, thermal dynamics are considered. In particular, the cell temperature $T(t)$ is given by

$$\frac{dT(t)}{dt} = \frac{1}{C_c}\left(Q(t) - \frac{T(t) - T_f(t)}{R_u}\right), \tag{2.61}$$

with $Q(t)$ given by

$$Q(t) = |I_{cell}(t)||V_{cell}(t) - (U_p(\theta_p^*(t)) - U_n(\theta_n^*(t)))|. \tag{2.62}$$

All the quantities used in (2.61) and (2.62) are defined above. To conclude the formulation of the SPMeT model, aging dynamics are added, representing the variation over time of the cell capacity $C_{cell}(t)$ and the growth of the SEI resistance $R_f(t)$.

$$\frac{dC_{cell}(t)}{dt} = \frac{3C_{cell}(t)}{R_p A_{surf} \Delta\theta_n c_s^{max}} j_s(t), \tag{2.63}$$

$$\frac{dR_f(t)}{dt} = -\frac{\tilde{v}_f}{\sigma_{sei}} j_s(t), \tag{2.64}$$

where the side reaction flux $j_s(t)$ is given by

$$j_s(t) = \frac{i_0(t)}{F} \exp\left(\frac{0.5F}{RT(t)}\eta_n(t)\right). \tag{2.65}$$

This concludes the overview on the Li-ion cell models used in this work.

2.4.3 Battery pack model

As mentioned previously, many applications require the use of many cells connected together in order to reach some specific requirements. Typically applications that only need a single cell are portable electronics: the power stored in a single cell is enough for the needs of e.g. a smartphone; moreover, having more cells requires more space, which is not a welcomed feature. Other applications instead, such as EVs, grid power storage and personal computers, require to store a higher power and have the physical space to allocate it. Therefore, Li-ion cells can be connected together either in series or parallel, depending on the needs of the specific application. Connecting cells together in series increases the total voltage at the terminals of the battery pack, while a parallel connection increases instead the total capacity. Very often, a combination of the two connections is used, with parts of the battery packs are connected in parallel and are referred to as *modules*. The modules are then

connected together in series.

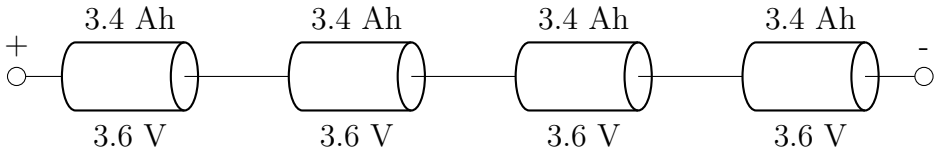


Figure 2.14: Example of series connected Li-ion cells.

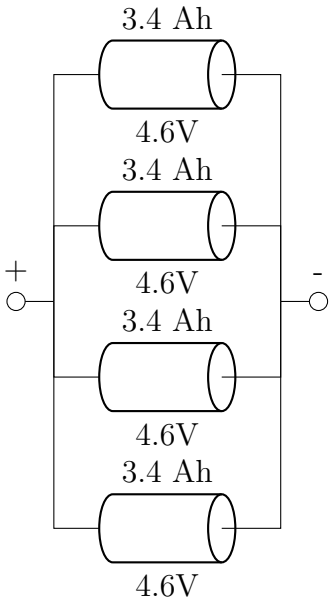


Figure 2.15: Example of parallel connected Li-ion cells.

For example, in Figure 2.14, four series-connected cells are represented. Each cell is considered to have a capacity of 3.4 Ah and a terminal voltage of $V = 3.6\ V$. The terminal voltage of this battery pack will be $V = 14.4\ V$. In Figure 2.15 instead four cells are shown connected in parallel; in this case, the terminal voltage will be equal to $V = 4.6\ V$, while the total capacity 13.6 Ah. Finally, in Figure 2.16, an hybrid combination with 3 modules of 4 parallel cells each, connected in series, is presented. This battery pack has a terminal voltage of $V = 13.8\ V$ and a capacity of 40.8 Ah.

In order to create a model of a battery pack, consider Kirchhoff's laws: for series connected cells, such as the ones depicted in Figure 2.14, it will result that the current entering the battery pack (considered to be the input of the system), will be the same in each cell and the voltage is the sum of the voltages of the single cells. Consider a battery pack composed of N series connected cells and indicate with $\mathbf{I}_h$ the current flowing in the h-th cell, with

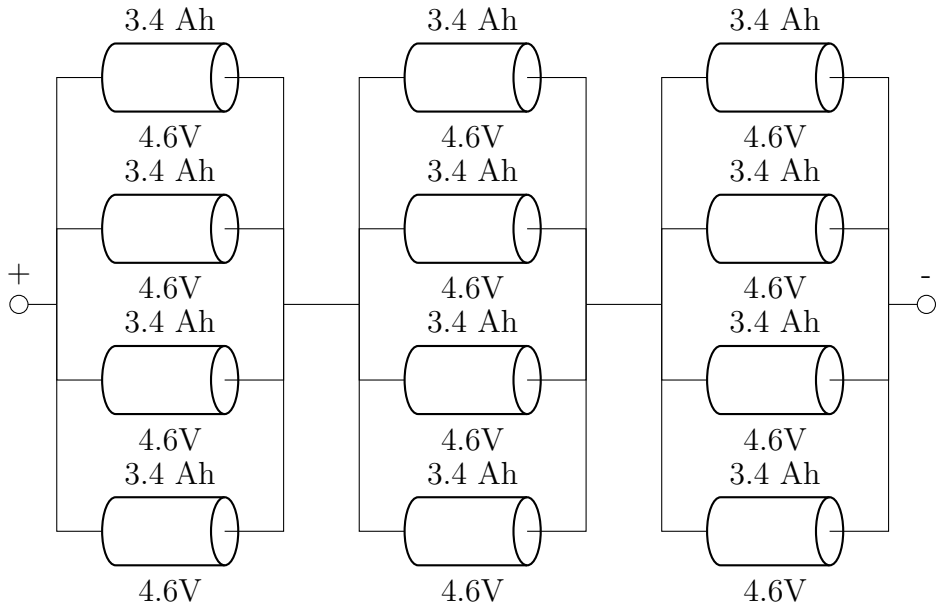


Figure 2.16: Example of series/parallel connected Li-ion cells.

$h = 1, \dots, N$. Cell terminal voltage instead is indicated with $\mathbf{V}_h$. Moreover, denote with $\mathbf{I}_{tot}$ the total current flowing in the battery pack.

Remark 2 *The bold notation for quantities like current and voltage will be used from now on to denote quantities defined over the full horizon considered, regardless of the fact that the model used is in continuous time or in discrete time, which will be specified every time.*

The equations governing a series-connected battery pack are the following

$$\begin{cases} \mathbf{I}_1 = \mathbf{I}_2 = \cdots = \mathbf{I}_N = \mathbf{I}_{tot}, \\ \mathbf{V}_t = \mathbf{V}_1 + \mathbf{V}_2 + \cdots + \mathbf{V}_N. \end{cases} \tag{2.66}$$

Note that the equations defined in (2.66), if the current is considered to be the input, are not regarded as algebraic constraints, as the dynamics of the cells are not coupled together.

Instead, for the parallel connected battery pack, Kirchhoff's laws create an algebraic coupling between the cells. Consider Figure 2.17, where it is clear that the input current $\mathbf{I}_{tot}$ splits into the N branches of the battery pack. Moreover, the voltage across each cell has to be equal. The system of equations is equal to

$$\begin{cases} \sum_{h=1}^{N} \mathbf{I}_h - \mathbf{I}_{tot} = 0, \\ \mathbf{V}_h - \mathbf{V}_{h-1} = 0, \quad \forall\, h \in \{2, \dots, N\}. \end{cases} \tag{2.67}$$

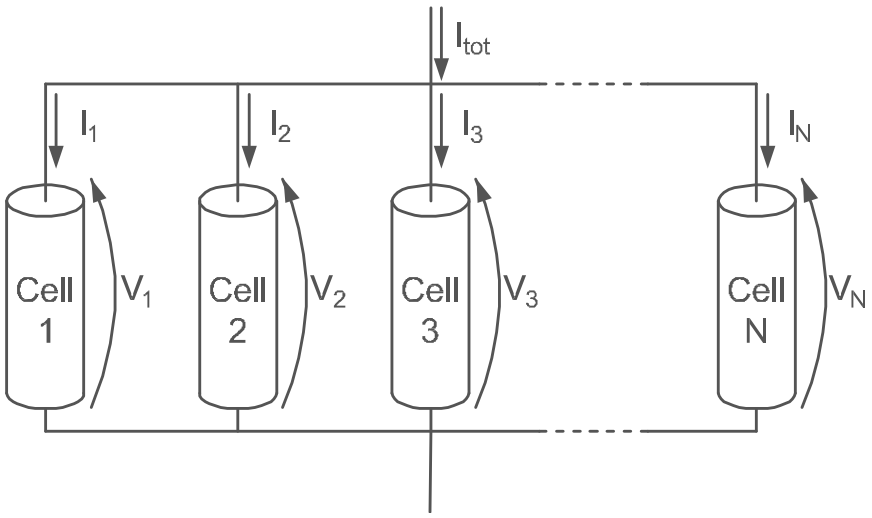


Figure 2.17: Parallel connection between Li-ion cells.

If the cells are exactly identical to one another, the current will split evenly among the N branches and the solution of this algebraic constraint would be trivial. However, this is almost never the case and, as discussed previously, cells are even slightly different from each other. This has the effect of unbalancing the electrical impedance of the cells, with the current splitting unevenly among them. The voltage however has to stay exactly equal at the terminals of each cell. The algebraic boundaries create therefore a very strong coupling among cells, that cannot be ignored. Combining the ODEs given by the dynamics of the single cells with the Algebraic Equations (AE)s of the boundaries, the system of equations will be a DAE system.

For the hybrid configuration in Figure 2.16, the current entering each module is equal to the input current of the battery pack $\mathbf{I}_{tot}$. The total voltage of the battery pack is then the sum of the voltages on the single modules connected in series.

2.5 Conclusions

In this Section, the preliminaries for this work have been given. First, an overview of the historical background of the technological development of Li-ion batteries is provided. Then the functioning principles, aging mechanisms and cell form factors were discussed. Battery Management Systems (BMS) were analyzed their usefulness proven. Finally, to conclude, Li-ion cell models have been presented, from the simple and fast Equivalent Circuit Models (ECM), to the complete and complex Electrochemical Models (EM), that describe in detail the chemical reactions in the Li-ion cell. Single cells can then be combined together to form battery packs with a higher output voltage and higher total capacity; in the case of a parallel connection, this makes the system

of equations representing the battery pack, a DAE system. When simulating a large system, the computational burden can become an issue. This gets addressed in the next Section.

Part II

Fast simulation of Li-ion battery packs

Chapter 3

Fast simulation of SPMeT-based battery packs

The contents of this Chapter are based on publication **P.1**

Electrochemical Models (EMs) are very expensive to simulate in terms of time spent and memory used, that increases with the precision and complexity of the model [Kemper et al., 2015]. This issue becomes more evident and impacting when dealing with a battery pack. Moreover, as explained in Section 2.4.3, the dynamical model of the battery pack consists of a DAE system of equations. In the literature, several works tackle the problem of reducing the complexity of EMs [Prada et al., 2012, Forman et al., 2010, Ohlberger et al., 2016, Xia et al., 2017b], try to balance the trade-off between complexity and accuracy [Khalik et al., 2021a, Gopalakrishnan and Offer, 2021] or consider numerical methods for the efficient simulation of single cells [Northrop et al., 2011, Bizeray et al., 2013, Cai and White, 2012, Han et al., 2021]. The use of computational reduction is extremely important for battery simulation purposes, parameter identification, state estimation and management. In the parameter identification of a P2D battery model, for example, several approaches have been developed in order to reduce the complexity of the identification procedure, see e.g. [Chun et al., 2020] and [Kim et al., 2019], that respectively use a long short-term neural network and a Bayesian neural network. Regarding state estimation instead several schemes have been proposed, such as the use of Kalman filtering [Jiang et al., 2021, Wang et al., 2019], a particle filter [Wang et al., 2018] or a multiple parameter optimization [Cao et al., 2019].

This part of the work is focused on the numerical simulation of an EM of a battery pack in a fast and efficient way, that significantly improves the simulation scalability for large systems. In order to speed up the simulation, the proposed method is to divide the system of equations in smaller subproblems using the WR method. Decomposing the system has the effect of reducing the overall complexity, because solving small subsystems, that contain fewer

equations than the original problem, is more efficient than solving a large one [Lelarasmee, 1982]. These have to be solved iteratively in order to reach convergence to the result of the original problem [White et al., 1985]. In addition, if these subsystems are built in such a way that they are independent, it is possible to solve them in parallel. WR methods are used to efficiently simulate large systems of equations. They were introduced to analyze complex integrated circuits in the time domain, hence the term *waveform*, referring to the signals [Lelarasmee et al., 1982]. As modern applications required more complex and larger integrated circuits, the need for a simulation technique that would be more efficient than classical and sequential integration methods is exaggerated. First introduced for ODE problems [Sand and Burrage, 1998], WR methods were extended to DAE systems, as explained and proven in [Crow and Ilić, 1994]. This methodology has been applied to many different areas such as power converters [Maciejewski et al., 2017], reaction-diffusion systems [Gander, 1999], electromagnetic systems [Gander and Ruehli, 2010] and others [Chang, 1989, Chang, 1990, Moayedi et al., 2013]. WR methods have never been applied to battery pack simulations. Moreover, note that WR is not a simple parallelization technique. Indeed, due to the algebraic constraints between parallel cells, an iterative scheme is required in order to guarantee the convergence to the solution of the original problem.

Starting from a standard WR approach, the modifications needed to obtain a convergent iterative method for the simulation of accurate EMs will be explained: first of all the necessity of having overlapping subsystems. Then the idea of a correction step is introduced. Finally numerical techniques to speed up the convergence of the WR method are analyzed: in particular the focus will be on a technique called Anderson Acceleration [Anderson, 1965], that is built on top of the modified WR method and uses the results provided from previous iterations to obtain a faster convergence. Note that the methodology proposed here is general and can be used in principle to simulate any EM, providing a higher efficiency when the model complexity increases. Besides, this could be useful as an add-on to existent simulators, such as LIONSIMBA [Torchio et al., 2016], BEST [Latz and Zausch, 2011, Less et al., 2012], MULTIBAT [Feinauer et al., 2019], PyBaMM [Sulzer et al., 2021], DandeLiion [Korotkin et al., 2021] and TOOFAB [Khalik et al., 2021a], to extend their single cell capabilities. As a proof of concept, the proposed approach has been tested first on a SPMeT with aging dynamics (thermal dynamics are considered only between the cells and the coolant). In particular, a detailed analysis has been conducted on battery packs with an increasing number of parallel cells. The results show that the method we developed is very effective in simulating large systems. When the system reaches a certain number of cells, we gain a considerable advantage in computation times.

For example in the case of a Tesla Model S battery level, composed of 74 cells in parallel, a reduction of computation times to a third using a 6 core processor was obtained. In addition, note that the precision of the obtained

result is not affected while reducing computation times. Subsequently, the same methodology will be applied to a DFN model with improved thermal dynamics, that completely consider thermal exchange between the cells of a battery pack.

3.1 Battery pack model and cell parameters

In this first part of the work, a parallel connection between cells is considered. Moreover, thermal dynamics regulating heat exchange between the cells of the battery pack are neglected; they will be added in Chapter 4. The reason for this choice is to reduce the degree of complication of the proposed method: first it is formulated and proven in its simplest form, then additions are made to have the best possible representation of a battery pack. The ultimate objective of this work is to create the framework for a toolbox able to simulate efficiently (in terms of time) a large battery pack. In addition, it is taken into consideration the fact that adding series-connected cells would not add any complication from the perspective of constraints. Only a set of differential equations would be added without any modification needed from a methodology point of view. For this reason, in order to prove the effectiveness of the proposed methodology, only a parallel connection is chosen.

Consider the model presented in Section 2.4.2, and in particular the SP-MeT model. The equations defining the states of the system are given by (2.35),(2.36),(2.37),(2.46),(2.61),(2.63) and (2.64). From these equations it is possible to define the vector containing the states of the system

$$\mathbf{x}_{cell}(t) = [\theta_p(t)\ \theta_n(t)\ q_p(t)\ q_n(t)\ \mathbf{c}_{e,p}(t)\ \mathbf{c}_{e,s}(t)\ \mathbf{c}_{e,n}(t)\ T(t)\ C_{cell}(t)\ R_f(t)]\,, \tag{3.1}$$

having defined

$$\mathbf{c}_{e,p}(t) = \left[{c_{e,p}}^{[1]}(t), {c_{e,p}}^{[2]}(t), \dots, {c_{e,p}}^{[P]}(t)\right], \tag{3.2}$$

$$\mathbf{c}_{e,s}(t) = \left[{c_{e,s}}^{[1]}(t), {c_{e,s}}^{[2]}(t), \dots, {c_{e,s}}^{[P]}(t)\right], \tag{3.3}$$

$$\mathbf{c}_{e,n}(t) = \left[{c_{e,n}}^{[1]}(t), {c_{e,n}}^{[2]}(t), \dots, {c_{e,n}}^{[P]}(t)\right]. \tag{3.4}$$

The set of differential equations representing the dynamics of a single Li-ion cell can be comprised ad

$$\dot{\mathbf{x}}_{cell} = f_{cell}\left(\mathbf{x}_{cell}(t), I_{cell}(t)\right), \tag{3.5}$$

with $\mathbf{x}_{cell}(t) \in \mathbb{R}^{7+3P}$, $I_{cell}(t) \in \mathbb{R}$ and $f_{cell} : \mathbb{R}^{7+3P} \times \mathbb{R} \to \mathbb{R}^{7+3P}$. Also, recall the algebraic boundaries given by Kirchhoff's laws governing the parallel connection given in (2.67).

The complete DAE system can be formulated and is given in Equation (3.6), where the voltage $V_h(t)$ of each cell is computed according to Equation (2.52). The initial conditions $\hat{\mathbf{x}}_h$, for $h = 1, \dots, N$, are needed to make the

problem solvable.

$$\begin{cases} \dot{\mathbf{x}}_1(t) = f_1\left(\mathbf{x}_1(t), I_1(t)\right), \\ \quad \vdots \\ \dot{\mathbf{x}}_N(t) = f_N\left(\mathbf{x}_N(t), I_N(t)\right), \\ \mathbf{x}_1(0) = \hat{\mathbf{x}}_1, \\ \quad \vdots \\ \mathbf{x}_N(0) = \hat{\mathbf{x}}_N, \\ \sum_{h=1}^{N} I_h(t) - I_{tot}(t) = 0, \\ V_1(t) - V_2(t) = 0, \\ \quad \vdots \\ V_{N-1}(t) - V_N(t) = 0, \end{cases} \tag{3.6}$$

The states and the currents can be aggregated into vectors

$$\mathbf{X}(t) = \left[\mathbf{x}_1(t), \mathbf{x}_2(t), \ldots, \mathbf{x}_N(t)\right]^\top, \tag{3.7a}$$

$$\mathbf{I}(t) = \left[I_1(t), I_2(t), \ldots, I_N(t)\right]^\top, \tag{3.7b}$$

with $\mathbf{X}(t) \in \mathbb{R}^{N(7+3P)}$ and $\mathbf{I}(t) \in \mathbb{R}^N$. The vector $\mathbf{X}(t)$ collects the states of every cell contained in the system, while $\mathbf{I}(t)$ includes the algebraic variables, in this case the currents. System (3.6) can be expressed using a condensed form

$$\begin{cases} \dot{\mathbf{X}}(t) = F\left(\mathbf{X}(t), \mathbf{I}(t)\right), \\ 0 = G\left(\mathbf{X}(t), \mathbf{I}(t), I_{tot}(t)\right), \\ \mathbf{X}(0) = \hat{\mathbf{X}}, \end{cases} \tag{3.8}$$

with $F : \mathbb{R}^{7+3P} \times \mathbb{R}^N \to \mathbb{R}^{7+3P}$, $G : \mathbb{R}^{7+3P} \times \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$. The function $F\left(\mathbf{X}(t), \mathbf{I}(t)\right)$ comprises all the equations describing the model presented in Section (2.4.3) for each cell, while $G\left(\mathbf{X}(t), \mathbf{I}(t), I_{tot}(t)\right)$ contains the algebraic constraints previously obtained in Equation (2.67), with $I_{tot}(t)$ as the input current for the entire battery pack.

All the parameters used in the SPMeT model and defined in the previous Section and refer to a Kokam SLPB 75106100 cell and are in accordance with [Ecker et al., 2015c, Ecker et al., 2015a]. They are reported in Table 3.1. These parameters were found experimentally by taking measures of a real cell. From a manufacturing standpoint, it is impossible to create cells that are exactly equal one another, so to simulate this fact, a variance was introduced on some of the parameters. The value for this variance was taken from the literature, see for reference [Rumpf et al., 2018, Baumhöfer et al., 2014], and the quantities to which is applied are indicated with an asterisk in Table 3.1. Note that not all the parameters have a variance applied to them, while in practice also these

parameters may exhibit differences. This was done, once again, according to [Ecker et al., 2015c, Ecker et al., 2015a]. In addition, the focus of this work is to prove the effectiveness of the methodology simulating cells that are different from each other; the fact that some parameters are constant does not affect the properties of the proposed scheme. For what concerns the effects the variance has on the model, we have to recall once again Kirchhoff's law: in a parallel connection, if the cells are identical and assuming initial conditions equal for all of them, the current would split evenly among them, making the DAE problem trivial. Otherwise, the current would divide itself in the N branches based on the value of the cells' parameters.

Table 3.1: SPMeT model cell parameters.

Parameter	Value	Unit	Parameter	Value	Unit
${\theta_p}^{100\%}$	0.26	–	${\theta_p}^{0\%}$	0.86	–
${\theta_n}^{100\%}$	0.75	–	${\theta_n}^{0\%}$	0.04	–
$c_{s,p}^{max}$	48580	$\frac{mol}{m^3}$	$c_{s,n}^{max}$	31920	$\frac{mol}{m^3}$
${R_p}^{(*)}$	6.49×10^{-6}	m	${R_n}^{(*)}$	8.7×10^{-6}	m
A_{surf}	0.4121	m^2	${\delta_p}^{(*)}$	54.5×10^{-6}	m
${\delta_s}^{(*)}$	19×10^{-6}	m	${\delta_n}^{(*)}$	73.7×10^{-6}	m
t_+^0	0.26	–	${\epsilon_e}^{(*)}$	0.296	–
$\tilde{v}_f$	73×10^{-3}	$\frac{kg}{mol}$	σ_{sei}	2500	$\frac{kg}{m^3}$

Given a finite horizon $[0, t_f]$, the system of equations (3.8) can be solved using, for example, the built-in Matlab® function *ode15s*, which is suitable to solve stiff ODE and DAE problems using implicit integration. In the following we rely on *ode15s*, but any other integrator can be used.

Initial conditions need to be provided in order for the problem to be solvable and they are assumed to be equal for all the cells, $\hat{\mathbf{x}}_1 = \hat{\mathbf{x}}_2 = ... = \hat{\mathbf{x}}_N$. Without loss of generality, their values are defined at $t = 0$ as follows

$$\hat{\mathbf{x}}_h(0) = [0.83\ 0.1\ 0\ 0\ 2000, \ldots, 2000\ 27000\ 0.01\ 310]^\top, \quad \forall\, h = 1, \ldots, N. \tag{3.9}$$

Simulating the problem corresponding to (3.8) as a whole, will be referred to as *centralized* method, opposed to the proposed method of dividing the system into smaller subproblems, to reduce their complexity. In the next Section, the WR approach is presented and detailed.

3.2 Waveform relaxation

First, the WR approach is described in detail with all the steps needed to obtain a method for the simulation of an EM-based battery pack that is faster than the centralized one. After, some examples of its efficiency and precision will be presented. In fact, the purpose of this part of the work is to obtain

a method that has the same solution of the centralized one, while having a faster simulation time. The convergence of WR methods has been extensively proven in the literature (see e.g. [Gander et al., 1999]). A formal proof of convergence will not be provided in this work, however some specific examples will be given.

3.2.1 Waveform relaxation algorithm

Given a battery pack composed of N cells, let us consider the set of indices $\mathcal{I} := \{1, \dots, N\} \subset \mathbb{N}$, where each index corresponds to a cell. We decompose $\mathcal{I}$ in $a \in \mathbb{N}$ subsets (or *subdomains*) denoted by $\langle j \rangle \subset \mathcal{I}$, $j = 1, \dots, a$, such that $\cup_{j=1}^{a} \langle j \rangle = \mathcal{I}$. We assume that each subdomain contains ordered and successive indices and has cardinality s, that is $\text{card}\, \langle j \rangle = s$ for $j = 1, \dots, a$. The cardinality s is called *length of the subdomains*. Moreover, we assume that

$$\begin{aligned} \text{card}\left(\langle j \rangle \cap \langle j+1 \rangle\right) &= q \text{ for } j = 1, \dots, N-1, \\ \text{card}\left(\langle j \rangle \cap \langle \tilde{j} \rangle\right) &= 0 \text{ for } \tilde{j} < j-1,\ \tilde{j} > j+1, \end{aligned} \tag{3.10}$$

where the positive integer q is called *overlap*. Hence, for $q > 0$, the decom-

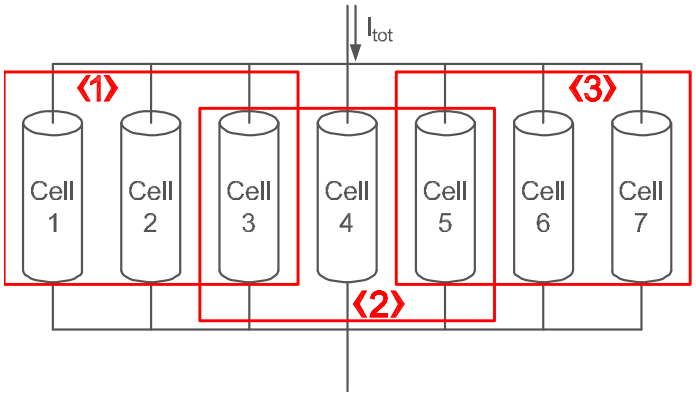


Figure 3.1: Example of overlapping subsystems in a case with $N = 7$, $s = 3$ and $q = 1$.

position $\mathcal{I} = \cup_{j=1}^{a} \langle j \rangle$ is said to be an *overlapping decomposition* and it clearly results that each subdomain $\langle j \rangle$ overlaps with only the two neighboring subdomains $\langle j-1 \rangle$ and $\langle j+1 \rangle$. As an example, consider a battery pack of 7 cells as the one depicted in Figure 3.1. This pack corresponds to the index set $\mathcal{I} = \{1, \dots, 7\}$. A possible overlapping decomposition of $\mathcal{I}$ is the one shown in red in Figure 3.1, where the subdomains are

$$\langle 1 \rangle = \{1, 2, 3\},\ \langle 2 \rangle = \{3, 4, 5\},\ \langle 3 \rangle = \{5, 6, 7\}, \tag{3.11}$$

with an overlap $q = 1$ and a length $s = 3$. The relationship between a, s, q and N is given by

$$N = a\,(s - q) + q. \tag{3.12}$$

Note that, in general, the only given value is N, while the other quantities are free parameters of choice. Now, denote by $\langle j \rangle, \ell$ the ℓ-th index (or cell) of the j-th subdomain. It is then possible to rewrite the current equilibrium $\sum_{h=1}^{N} I_h(t) - I_{tot}(t) = 0$, defined in (3.6), as

$$\sum_{\ell=1}^{s} I_{\langle j \rangle,\ell}(t) + I_{rest,\langle j \rangle}(t) - I_{tot}(t) = 0, \tag{3.13}$$

for $j = 1, \ldots, a$, where

$$I_{rest,\langle j \rangle}(t) = \sum_{h \in \mathcal{I} \setminus \langle j \rangle} I_h(t). \tag{3.14}$$

The term $I_{rest,\langle j \rangle}(t)$ represents the sum of the currents in all cells of the whole battery pack excluding the ones corresponding to the subdomain $\langle j \rangle$. At this point, an important remark is needed: the definition of $I_{rest,\langle j \rangle}(t)$ allows one to avoid accounting for the same currents multiple times. In other words, the objective is to take and sum every current only once, even if the subsystems are overlapping. Even though this is formally equivalent to the constraint on the sum of the currents in system (3.6), this formulation allows to make the subsystems independent from each other, with the only condition of having $I_{rest,\langle j \rangle}(t)$ known. It will be shown that the WR method is iterative, and this value will be taken from the results of previous iterations. Using the notation just presented, it is possible to reformulate the DAE system (3.6) in an equivalent *domain decomposition* form:

$$\begin{cases} \dot{\mathbf{x}}_{\langle j \rangle,1}(t) = f_{\langle j \rangle,1}\left(\mathbf{x}_{\langle j \rangle,1}(t), I_{\langle j \rangle,1}(t)\right), \\ \quad \vdots \\ \dot{\mathbf{x}}_{\langle j \rangle,s}(t) = f_{\langle j \rangle,s}\left(\mathbf{x}_{\langle j \rangle,s}(t), I_{\langle j \rangle,s}(t)\right), \\ \mathbf{x}_{\langle j \rangle,1}(0) = \hat{\mathbf{x}}_{\langle j \rangle,1}, \\ \quad \vdots \\ \mathbf{x}_{\langle j \rangle,s}(0) = \hat{\mathbf{x}}_{\langle j \rangle,s}, \\ \sum_{\ell=1}^{s} I_{\langle j \rangle,\ell}(t) + I_{rest,\langle j \rangle}(t) - I_{tot}(t) = 0, \\ V_{\langle j \rangle,1}(t) - V_{\langle j \rangle,2}(t) = 0, \\ \quad \vdots \\ V_{\langle j \rangle,s-1}(t) - V_{\langle j \rangle,s}(t) = 0, \end{cases} \tag{3.15}$$

for $j = 1, \ldots, a$.

A compact formulation can be provided for this system:

$$\begin{cases} \dot{\mathbf{X}}_{\langle j \rangle}(t) = F_{\langle j \rangle}\left(\mathbf{X}_{\langle j \rangle}(t), \mathbf{I}_{\langle j \rangle}(t)\right), \\ 0 = G_{\langle j \rangle}\left(\mathbf{X}_{\langle j \rangle}(t), \mathbf{I}_{\langle j \rangle}(t), I_{rest,\langle j \rangle}(t), I_{tot}(t)\right), \\ \mathbf{X}_{\langle j \rangle}(0) = \hat{\mathbf{X}}_{\langle j \rangle}, \end{cases} \tag{3.16}$$

for $j = 1, \ldots, a$, where the vectors $\mathbf{X}_{\langle j \rangle}(t)$ and $\mathbf{I}_{\langle j \rangle}(t)$ are defined as

$$\mathbf{X}_{\langle j \rangle}(t) = \left[\mathbf{x}_{\langle j \rangle,1}(t), \ldots, \mathbf{x}_{\langle j \rangle,s}(t)\right]^\top, \tag{3.17a}$$

$$\mathbf{I}_{\langle j \rangle}(t) = \left[I_{\langle j \rangle,1}(t), \ldots, I_{\langle j \rangle,s}(t)\right]^\top. \tag{3.17b}$$

Note that the function $F_{\langle j \rangle} : \mathbb{R}^{s(7+3P)} \times \mathbb{R}^s \to \mathbb{R}^{s(7+3P)}$ has as arguments only the states and the currents of subsystem j, while $G_{\langle j \rangle} : \mathbb{R}^{s(7+3P)} \times \mathbb{R}^s \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^s$ has as arguments the states and the currents of the j-th subsystem, the current $I_{rest,\langle j \rangle}(t)$ and the total input current $I_{tot}(t)$. Notice also that $G_{\langle j \rangle}$ is different from the function G defined in Equation (3.8): in Equation (3.15) the current constraint has been rewritten to distinguish the currents belonging to the subsystem from all the other ones. The objective of this subdivision is to create independent subsystems that can be solved in parallel and, if their resolution is iterated, to obtain a result that will converge to the solution of the original problem (3.6). The parallel resolution is possible because of the particular structure created for the subsystems and in particular, thanks to the definition of I_{rest}. Moreover, because these subsystems are smaller and therefore in principle faster to solve than (3.6), it will be shown that this type of resolution can be more time efficient than the centralized method.

At this point, it is crucial to discuss the importance of having an overlapping decomposition. Clearly, the first intuitive approach would be to consider no overlap (i.e. $q = 0$), to reduce the number of cells in each subdomain that are present multiple times in different subsystems, thus reducing complexity. Doing so, the total number of cells solved in the subsystems would be minimal. This is however not possible because it would result in a *domain decomposition* system not equivalent to the main problem (3.6).

An example can be used to explain this issue. Consider the system of 7 cells of Figure 3.1. The corresponding set $\mathcal{I} = \{1, \ldots, 7\}$ can be decomposed in the two non-overlapping subdomains $\langle 1 \rangle = \{1, 2, 3\}$ and $\langle 2 \rangle = \{4, 5, 6, 7\}$, with $q = \text{ card}\, \langle 1 \rangle \cap \langle 2 \rangle = 0$. Without loss of generality and in order to re-use the example of Figure 3.1, the assumption of having card $\langle j \rangle = s$ for $j = 1, \ldots, a$ was neglected. The first subproblem corresponding to $\langle 1 \rangle$ is characterized by the algebraic constraints for the equilibrium of the voltages

$$\begin{cases} V_1(t) - V_2(t) = 0, \\ V_2(t) - V_3(t) = 0, \end{cases} \tag{3.18}$$

while the second subproblem corresponding to $\langle 2 \rangle$ has the voltage constraints

$$\begin{cases} V_4(t) - V_5(t) = 0, \\ V_5(t) - V_6(t) = 0, \\ V_6(t) - V_7(t) = 0. \end{cases} \tag{3.19}$$

It is then clear that the constraint

$$V_3(t) - V_4(t) = 0 \tag{3.20}$$

is missing in both sets (3.18) and (3.19). Therefore, by solving the two subsystems corresponding to $\langle 1 \rangle$ and $\langle 2 \rangle$, it is not guaranteed that the constraint (3.20) is satisfied, even though the subsystems $\langle 1 \rangle$ and $\langle 2 \rangle$ are coupled in terms of the sum of the currents, i.e. Equation (3.13). Depending on the different properties of the single cells, it can happen that the obtained current distribution leads to voltages $V_3(t)$ and $V_4(t)$ that violate (3.20). On the other hand, if one enlarges by one cell the first subdomain and considers an overlapping decomposition $\langle 1 \rangle = \{1, 2, 3, 4\}$ and $\langle 2 \rangle = \{4, 5, 6, 7\}$, then the voltage of a cell appears in both subsystems. In the considered example we will have that $V_4(t)$ appears as $V_3(t) - V_4(t)$ in $\langle 1 \rangle$ and $V_4(t) - V_5(t)$ in $\langle 2 \rangle$. By doing so, all the algebraic constraints are enforced in all of the subsystems. However note that this is not sufficient in order to guarantee that the solution of a *domain decomposition* based method coincides with the solution of the original problem (3.6) (i.e. obtain convergence).

For further discussions regarding the choice of the overlap in the field of domain decomposition methods, refer to [Gander, 1999, Gander and Ruehli, 2004, Gander, 2008, Gander and Ruehli, 2010] and references therein.

It was mentioned previously that the resolution of subsystems needs to be iterated. Before introducing the iterative algorithm, convergence conditions need to be discussed and analyzed. From the *domain decomposition* systems (3.16) it is possible to derive a Jacobi-type WR iteration. Considering an iteration k, for a given $\mathbf{I}^{k-1}_{\langle j \rangle}(t)$, $j = 1, \ldots, a$ computed in the previous iterations (or an initial guess, for the first one), the WR iteration is computed by solving (in parallel) for $\mathbf{X}^k_{\langle j \rangle}(t)$ and $\mathbf{I}^k_{\langle j \rangle}(t)$, the subproblems

$$\begin{cases} \dot{\mathbf{X}}^k_{\langle j \rangle}(t) = F_{\langle j \rangle}\left(\mathbf{X}^k_{\langle j \rangle}(t), \widetilde{\mathbf{I}}^k_{\langle j \rangle}(t)\right), \\ 0 = G_{\langle j \rangle}\left(\mathbf{X}^k_{\langle j \rangle}(t), \widetilde{\mathbf{I}}^k_{\langle j \rangle}(t), I^{k-1}_{rest,\langle j \rangle}(t), I_{tot}(t)\right), \\ \mathbf{X}^k_{\langle j \rangle}(0) = \hat{\mathbf{X}}_{\langle j \rangle}, \\ \widetilde{\mathbf{I}}^k_{\langle j \rangle}(t) = D_{\langle j \rangle}\mathbf{I}^k_{\langle j \rangle}(t) + (\mathbb{I} - D_{\langle j \rangle})\mathbf{I}^{k-1}_{\langle j \rangle}(t), \end{cases} \tag{3.21}$$

for $k = 1, 2, \ldots$, where $\mathbb{I}$ is the $s \times s$ identity matrix and $D \in \mathbb{R}^{s \times s}$ is a diagonal

matrix having diagonal entries equal to

$$D_{\langle j\rangle}(\ell,\ell) := \begin{cases} \omega_D & \text{if } \ell \in \{1,\dots,q\} \cup \{s-q+1,\dots,s\} \text{ and } j = 2\dots,a-1, \\ \omega_D & \text{if } \ell \in \{s-q+1,\dots,s\} \text{ and } j = 1, \\ \omega_D & \text{if } \ell \in \{1,\dots,q\} \text{ and } j = a, \\ 1 & \text{otherwise.} \end{cases} \tag{3.22}$$

Here, $\omega_D \in [0,1]$ is a *relaxation parameter* that can affect the convergence speed of the WR method. Notice that ω_D is applied only to the currents of the overlap and the definition of $D_{\langle j\rangle}$ takes into account that the first and last subsystem do not overlap. The systems defined in (3.21) are DAE problems of the same form of the full DAE system (3.8), but they have much smaller dimension since they are defined on $s \ll N$ cells. Notice how the coupling among the subsystems is obtained via the currents $I^{k-1}_{rest,\langle j\rangle}(t)$, which are computed using $I^{k-1}_{\langle j\rangle,\ell}(t)$, at the $(k-1)$-th iteration according to Equation (3.14). Moreover, since the subsystems (3.21) have the same form of (3.8), one can use exactly the same numerical solver (e.g., the built-in Matlab® function *ode15s*).

Now a few words about the choice of the relaxation parameter ω_D in Equations (3.21) and (3.22). If $\omega_D = 0$ is chosen, then the currents in the overlap do not account for values coming from the previous iterations and the method defined by Equation (3.21) can not converge to a correct solution. This happens because the values of $I^{k}_{\langle j\rangle,\ell}(t)$ never get updated, the values $I^{k-1}_{\langle j\rangle,\ell}(t)$ are used, and therefore the currents are always equal to the initial conditions. The opposite, and intuitive, choice would be to assign $\omega_D = 1$, which leads to a full update of the currents in the overlap. This typical choice, which corresponds to a standard WR method, does not necessarily lead to a convergent method. For similar discussions in the context of domain decomposition methods for stationary equation we refer to [Ciaramella et al., 2019]. If the standard WR relaxation iteration (with $\omega_D = 1$) does not converge, one can try to reduce the value of ω_D to relax (or damp) the iterations and obtain convergence. However, even if convergence is obtained by a small ω_D, the iteration converges generally quite slowly. Only after introducing an appropriate *correction step*, the positive effect of ω_D on the convergence of our WR method becomes significant.

Obtaining convergence is a crucial step of the proposed WR method: during the first tests of the WR method it was observed a divergent behaviour. This issue that was encountered when using the WR iteration (3.21) is discussed at this point. The algebraic constraint on the current needs to be corrected along the iterations. Since the goal is to guarantee that the current constraint (3.13) is satisfied, the sum

$$I^{k}_{sum}(t) = \sum_{j=1}^{a}\sum_{\ell=1}^{s} D_{\langle j\rangle}(\ell,\ell) I^{k}_{\langle j\rangle,\ell}(t) \qquad \forall t \in [0,t_f]\,, \tag{3.23}$$

with $D_{\langle j \rangle}(\ell, \ell)$ defined in (3.22), should become closer to the value of $I_{tot}(t)$, i.e. the given input, as iterations go on. Notice that the value of $I^k_{sum}(t)$ is computed in such a way that the currents belonging to the cells of the overlap are not accounted twice. Instead they are weighed in such a way that half is taken from the first subsystem and the second half from the second subsystem.

Remark 3 *This is done by choosing $\omega_D = 0.5$. Any other value for ω_D can be chosen, however Equation (3.23) would have to be adjusted accordingly.*

However, in the performed tests, $I^k_{sum}(t)$ was computed at the end of each iteration and observed a lack of convergence was observed: the sum of the currents $I^k_{sum}(t)$ would oscillate heavily around the correct value $I_{tot}(t)$ and the oscillations become larger and larger along the iterations, as shown in Figure 3.2. In this figure, the system and the subdivision are the same of Figure 3.1, where the input was set to a constant value $I_{tot}(t) = -7.5A \ \forall t \in [0, 100]$. What it is possible to observe is an over-compensation: at each iteration every subsystem tries to "correct" the small error in the value of $I^k_{sum}(t)$, generated in the previous iteration and contained in the term $I^{k-1}_{rest,\langle j \rangle}(t)$. However because the subsystems are all solved at the same time, every one of them adds its own correction to compensate for the over or under estimation of $I^{k-1}_{rest,\langle j \rangle}(t)$. When one looks at the updated value of $I^k_{sum}(t)$, all these corrections sum up, creating an even bigger error. Hence, the iterations diverge and the WR method fails.

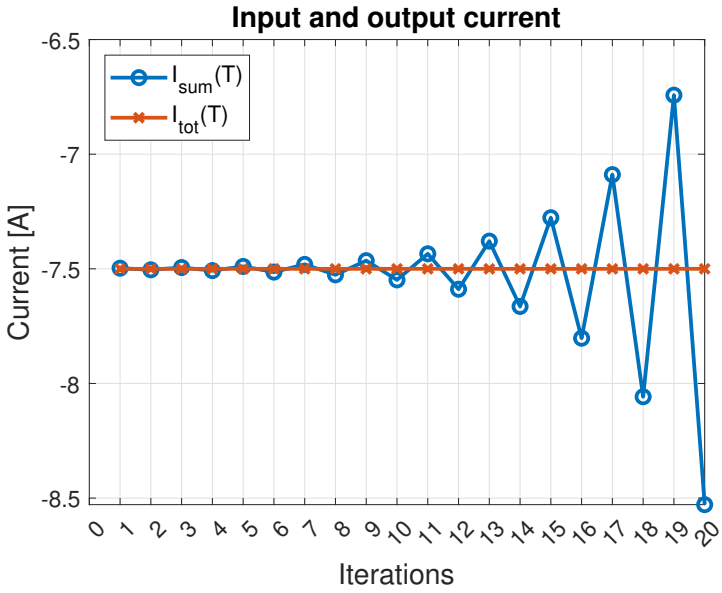


Figure 3.2: Convergence of $I_{sum}(t_f)$ to $I_{tot}(t_f)$ without a correction step applied.

The problem essentially lies in the lack of communication among the subsystems: each subsystem can exchange only current information with its two neighbors. To solve this problem one could introduce a so-called second-level

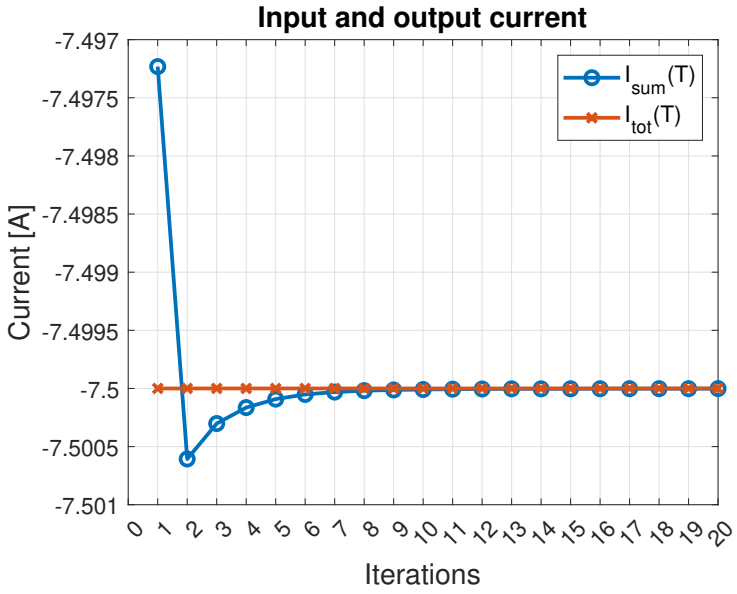


Figure 3.3: Convergence of $I_{sum}(t_f)$ to $I_{tot}(t_f)$ with the correction step applied.

(or coarse) correction that generally consists in a modification of the result of the WR iterate; see, e.g., [Ciaramella et al., 2019, Ciaramella and Gander, 2017, Chaouqui et al., 2018] and references therein. However, the definition of an appropriate correction step is a complicated and delicate task, which depends heavily on the class of problems under consideration. For the battery pack problem considered here, an appropriate correction step is designed by reinforcing the algebraic current constraint at the end of each iteration. This allows to always start the successive iteration with a feasible current distribution. At iteration k, the sum $I^k_{sum}(t)$ is computed and calculate how far $I^k_{sum}(t)$ is from $I_{tot}(t)$, that is

$$I^k_{diff}(t) = I_{tot}(t) - I^k_{sum}(t) \qquad \forall t \in [0, t_f]\,, \tag{3.24}$$

and divide this quantity by the number of cells N

$$\delta_c{}^k(t) = \frac{I^k_{diff}(t)}{N} \qquad \forall t \in [0, t_f]\,. \tag{3.25}$$

The functions $\delta_c{}^k(t)$ are used to complete the *correction step*

$$\begin{aligned} I^{*,k}_{\langle j\rangle,\ell}(t) = I^k_{\langle j\rangle,\ell}(t) + \delta_c{}^k(t) \quad & \text{for } j = 1, \dots, a \\ & \text{for } \ell = 1, \dots, s \\ & \forall t \in [0, t_f]\,, \end{aligned} \tag{3.26}$$

where the currents $I^{*,k}_{\langle j\rangle,\ell}(t)$, $j = 1, \dots, a$ and $\ell = 1, \dots, s$, satisfy the current

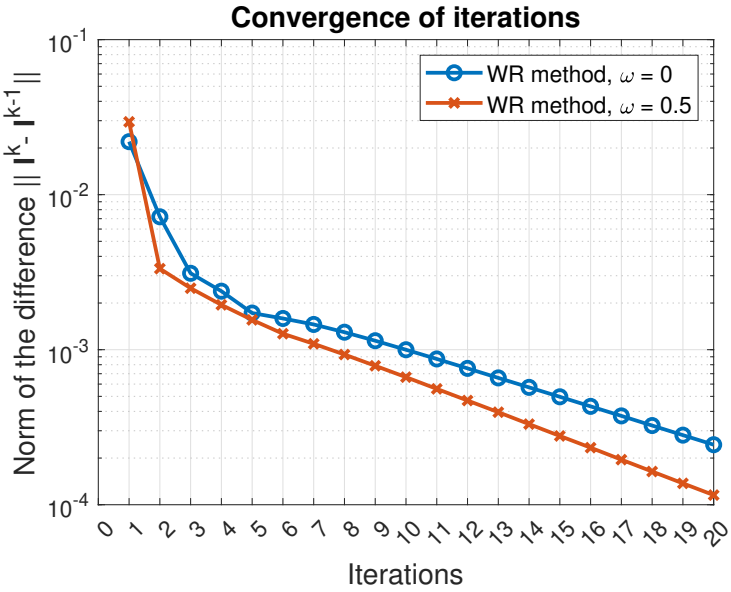


Figure 3.4: Comparison between the convergence of the WR method with different values of ω_D.

constraint defined in (3.23)

$$\sum_{j=1}^{a}\sum_{\ell=1}^{s} D_{\langle j\rangle}(\ell,\ell) I^{*,k}_{\langle j\rangle,\ell}(t) = I_{tot}(t) \tag{3.27}$$

for all $t \in [0, t_f]$.

Hence, these currents represent a feasible distribution and we can use them in place of $I^k_{\langle j\rangle,\ell}(t)$, to perform the next $(k+1)$-th iteration. In this way, the values returned from the k-th iteration, used as input of iteration $k+1$, respect the algebraic constraints. With this adjustment, the convergence of the method is obtained, as shown in Figure 3.3, for the same conditions of Figure 3.2.

It is now possible to test the method for different values of ω_D, comprising the *correction step*. The considered example is the same of Figure 3.3, with $N = 7$, $s = 3$ and $q = 1$. The blue line of Figure 3.4 corresponds to $\omega_D = 1$, i.e. the case where the overlap currents are fully updated, and the red line corresponds to $\omega_D = 0.5$. The use of the parameter ω_D $(0 < \omega_D \leq 1)$ has the effect of retaining more information throughout iterations, making them more efficient. The improvement is not radical, however it is interesting to see that without changing the structure of the problem and its computational complexity, a faster convergence is obtained.

The full WR strategy for the simulation of a battery pack is presented in Algorithm 1. The inputs needed for this algorithm are the total current $I_{tot}(t)$ for the considered interval, an initial guess for the currents $\mathbf{I}^0(t)$ over the considered horizon and the states $\mathbf{X}(0)$ at the initial time instant. Moreover, also the parameter ω_D previously described is needed, a tolerance ϵ_{wr}

Algorithm 1 (WR Jacobi-type algorithm)

Require: $I_{tot}(t), \mathbf{I}^0(t)\ \forall\ t \in [0, t_f]$, $\mathbf{X}(0)$, a parameter $\omega_D \in (0, 1]$, a tolerance ϵ_{wr} and k_{max}.

1: Set $k \leftarrow 1$ and $\mathbf{I}^1 \leftarrow$ inf.
2: **while** $||\mathbf{I}^k - \mathbf{I}^{k-1}|| > \epsilon_{wr}$ and $k < k_{max}$ **do**
3: **for all** $j \in \{1, \dots, a\}$ **do**
4: Compute $I^{k-1}_{rest,\langle j\rangle}$ using (3.14).
5: Solve (3.21) on $[0, t_f]$ to get $\mathbf{X}^k_{\langle j\rangle}$ and $\mathbf{I}^k_{\langle j\rangle}$.
6: **end for**
7: Compute $I^{*,k}_{\langle j\rangle,\ell}$ for $j = 1, \dots, a$ and $\ell = 1, \dots, s$ using (3.25) and (3.26).
8: Set $I^k_{\langle j\rangle,\ell} \leftarrow I^{*,k}_{\langle j\rangle,\ell}$ for $j = 1, \dots, a$ and $\ell = 1, \dots, s$.
9: Set $k \leftarrow k + 1$.
10: **end while**

(an arbitrarily small value), used as a criteria to stop the iterations, and a maximum value of iterations k_{max}, to avoid infinite loops. The parameters of the cells have also to be provided. Having no information on the distribution of the currents in the parallel branches of the battery pack, the initial guess is obtained by dividing $I_{tot}(t)$ by N, assuming that the input current divides itself equally among the parallels. The algorithm then starts an iterating cycle, in which every subsystem gets solved for its variables on a given time interval, while all the quantities not included in the subsystem (i.e. $I^{k-1}_{rest,\langle j\rangle}$) are taken from the previous iteration. The cycle returns the values of $I^k_{\langle j\rangle}(t)$ at every time instant for every subsystem, that will be used in the successive iterates to calculate $I^{k-1}_{rest,\langle j\rangle}(t)$. This gets repeated until the current modulus of the difference between successive iterates gets smaller than a certain threshold ϵ_{wr}, or the number of iterations k exceeds the maximum allowed k_{max}.

For what concerns the precision of the proposed method, it has been tested by comparing the current distribution (i.e. the current flowing in each of the parallel branches) obtained by the centralized method, denoted with I_{centr}, with the one obtained by the WR approach, denoted with I_{WR}. In order to know how precise I_{WR} is with respect to I_{centr} (that will be taken as the reference solution), compute the absolute error as

$$E_{abs} = ||I_{centr} - I_{WR}||$$

and the relative error as

$$E_{rel} = \frac{||I_{centr} - I_{WR}||}{||I_{centr}||}.$$

For the example presented above, with $N = 7$, $s = 3$, $q = 1$, a time horizon of $[0, 100]$ and $I_{tot} = -7.5A$, the absolute error is $E_{abs} = 8.98 \times 10^{-5}$ and the relative error is $E_{rel} = 1.023 \times 10^{-5}$. Keep in mind that the solution provided

by the WR method depends on the threshold set for stopping the iterations (indicated in the algorithm with ϵ_{wr} and here set to $\epsilon_{wr} = 10^{-6}$). Therefore, even if in some cases the accuracy of the solution appears to be low, one could lower the threshold, reducing the absolute and relative errors.

Remark 4 *Convergence of the problem to the true global solution is guaranteed if the problem is well posed (i.e. has a unique solution) and the method converges. In our particular case, the succession generated by the method converges (for continuity of the considered functions), therefore it converges also to the true global solution of the problem.*

In order to further improve convergence speed of the methodology presented, the use of an acceleration scheme is presented in the following, built on top of the WR iteration.

3.3 Anderson Acceleration

The WR method presented in the previous section can be further improved by a numerical strategy called AA [Anderson, 1965], which is used to accelerate fixed-point iterations. The WR method is exactly a fixed-point iteration. Given a current distribution $\mathbf{I}^{k-1}$, the new value $\mathbf{I}^k$ is computed by one cycle of the while loop in Algorithm (1). In a compact notation, this single iteration can be written as

$$\mathbf{I}^k = S(\mathbf{I}^{k-1}),$$

where the function S is known implicitly if Algorithm (1) is available in the form of a routine.

Denote by $\mathbf{I}^b \in \mathbb{R}^N$ the approximations to a solution $\mathbf{I}$ (i.e. the current distribution in the battery pack) obtained at iterations $b = 1, \ldots, k$, and by

$$\gamma^b := S(\mathbf{I}^b) - \mathbf{I}^b, \tag{3.28}$$

the corresponding residuals. Let $m \in \mathbb{N}^+$ and define $m_k = \min(m, k)$. According to the AA strategy, the new approximation $\mathbf{I}^{k+1}$ is computed as

$$\mathbf{I}^{k+1} = \sum_{b=0}^{m_k} \alpha_b^k S(\mathbf{I}^{k-m_k+b}), \tag{3.29}$$

where the coefficients α_b^k for $b = 0, \ldots, m_k$ are obtained as the solution to the following minimization problem

$$\min_{\alpha_0^k, \ldots, \alpha_{m_k}^k} \left|\left| \sum_{b=0}^{m_k} \alpha_b^k \gamma^{k-m_k+b} \right|\right| \quad \text{s.t.} \sum_{b=0}^{m_k} \alpha_b^k = 1. \tag{3.30}$$

In other words, what we are doing is extending the previously explained concept of using ω_D to weight the results coming from previous iterations, to take

account also for all the previously obtained current distributions. Many algorithms and different implementations of the Anderson Acceleration method are available in the literature and they are fully described in [Walker, 2011], depending on the norm used in Equation (3.30).

Algorithm (2) shows a possible implementation of the Anderson Acceleration. Notice that iterations require the storage of the history of residuals of

Algorithm 2 Anderson Acceleration algorithm

Require: Fixed-point operator S, initial guess $\mathbf{I}^0$, m, k_{max}, ϵ_{aa}.
1: Set $k \leftarrow 1$, $\mathbf{I}^1 = S(\mathbf{I}^0)$.
2: **while** $k < k_{max}$ and $||\mathbf{I}^k - \mathbf{I}^{k-1}|| > \epsilon_{aa}$ **do**
3: Set $m_k \leftarrow \min(m, k)$.
4: Compute $s_b = S(\mathbf{I}^k)$.
5: Set $\gamma^k \leftarrow s_b - \mathbf{I}^k$.
6: Solve the minimization problem $||\sum_{b=0}^{m_k} \alpha_b^k \gamma^{k-m_k+b}||$ s.t. $\sum_{b=0}^{m_k} \alpha_b^k = 1$.
7: Set $\mathbf{I}^{k+1} \leftarrow \sum_{b=0}^{m_k} \alpha_b^k s_{k-m_k+b}$.
8: Set $k \leftarrow k + 1$.
9: **end while**

depth at most $m + 1$. This fact is crucial when the dimension N is large. In this case one has to consider small values of m to reduce the amount of information stored during the iterations.

A crucial step in the Anderson strategy is the solution of the minimization problem (3.30), where any norm could be used. It was decided to use the Euclidean norm, so the first-order sufficient optimality condition for the minimization problem is a linear system [Giorgi and Guerraggio, 1994]. Different approaches are considered to solve (3.30); see, e.g., [Gander et al., 2014]. An efficient implementation of Anderson algorithm is proposed in [Walker, 2011], where (3.30) is transformed into an unconstrained problem. To do so, one defines

$$\Delta\gamma^b = \gamma^{b+1} - \gamma^b, \quad \forall b, \tag{3.31}$$

and set

$$\tilde{\mathbf{\Gamma}}_k = [\Delta\gamma^{k-m_k}, \ldots, \Delta\gamma^{k-1}]. \tag{3.32}$$

Then the Least Squares (LS) problem (3.30) is equivalent to

$$\min_{\eta=[\eta_0,\ldots,\eta_{m_k-1}]^\top} \|\gamma^k - \tilde{\mathbf{\Gamma}}_k \eta\|_2. \tag{3.33}$$

Notice that the coefficients α_b in (3.30) and η_b are related by $\alpha_0 = \eta_0$, $\alpha_b = \eta_b - \eta_{b-1}$ for $1 \leq b \leq m_k - 1$ and $\alpha_{m_k} = 1 - \eta_{m_k-1}$. The unconstrained LS problem (3.33) leads to a modified form of Anderson acceleration given in Algorithm (3), where the LS solution is denoted by $\eta^k = \left[\eta_0^k, \ldots, \eta_{m_k-1}^k\right]^\top \in \mathbb{R}^{m_k}$, and it

results

$$\begin{aligned}\mathbf{I}^{k+1} &= S(\mathbf{I}^k) - \sum_{b=0}^{m_k-1} \eta_b^k [S(\mathbf{I}^{k-m_k+b-1}) - S(\mathbf{I}^{k-m_k+b})] \\ &= S(\mathbf{I}^k) - \mathcal{S}_k \eta^k,\end{aligned} \tag{3.34}$$

where $\mathcal{S}_k = [\Delta S_{k-m_k}, \ldots, \Delta S_{k-1}] \in \mathbb{R}^{m_k \times m_k}$ with $\Delta S_b = S(\mathbf{I}^{b+1}) - S(\mathbf{I}^b)$ for each b.

Algorithm 3 Modified Anderson Acceleration

Require: Fixed-point operator S, initial guess $\mathbf{I}^0$, m, k_{max}, ϵ_{aa}.
1: Set $k \leftarrow 1$, $\mathbf{I}^1 = S(\mathbf{I}^0)$, $\gamma^0 = \mathbf{I}^1 - \mathbf{I}^0$ and $s_0 \leftarrow \mathbf{I}^1$.
2: **while** $k < k_{max}$ and $||\gamma^k|| > \epsilon_{aa}$ **do**
3: Set $m_k \leftarrow \min(m, k)$.
4: Compute $s_k = S(\mathbf{I}^k)$.
5: Set $\gamma^k \leftarrow s_k - \mathbf{I}^k$.
6: Set $\Delta\gamma^{k-1} \leftarrow \gamma^k - \gamma^{k-1}$ and $\tilde{\mathbf{\Gamma}}_k \leftarrow [\Delta\gamma^{k-m_k}, \ldots, \Delta\gamma^{k-1}]$.
7: Solve the LS problem $\eta^k = [\eta_0^k, \ldots, \eta_{m_k-1}^k]^\top = \text{argmin}_\eta ||\gamma^k - \tilde{\mathbf{\Gamma}}_k \eta||_2$.
8: Set $\Delta S_{k-1} \leftarrow s_k - s_{k-1}$ and $\mathcal{S}_k \leftarrow [\Delta S_{k-m_k}, \ldots, \Delta S_{k-1}]$.
9: Set $\mathbf{I}^{k+1} \leftarrow s_k - \mathcal{S}_k \eta^k$.
10: Set $k \leftarrow k + 1$.
11: **end while**

As the algorithm proceeds, the successive LS problems in step 7 are solved by a QR-factorization. The details can be found in [Walker, 2011, Walker and Ni, 2011].

Remark 5 *In particular, one first decomposes $\widetilde{G}_k = Q_k R_k$ with a factorization "economy-size". Then the solution γ^k is obtained as $\gamma^k = R_k^{-1}(Q_k^\top \mathbf{g}^k)$. In this step, the QR-factorization can be computed efficiently. In fact, suppose that the factorization $\widetilde{G}_k = Q_k R_k$ is known and that one has to compute the decomposition $\widetilde{G}_{k+1} = Q_{k+1} R_{k+1}$. Since $\widetilde{G}_{k+1}$ is obtained by adding one column to $\widetilde{G}_k$, that is $\widetilde{G}_{k+1} = [\widetilde{G}_k, \Delta\mathbf{g}^k]$, then the factors Q_{k+1} and R_{k+1} can be obtained by adding one column to Q_k and R_k, respectively. This can be achieved by a single modified Gram-Schmidt sweep.*

It is important to notice that the Anderson Acceleration strategy uses the WR Algorithm 1 as a black-box. Therefore, each time the fixed-point function S is applied to a current distribution, this corresponds to one iteration of the WR Algorithm 1. This means that, if the index m_k does not get too large (as observed for the class of problems considered in this work), the effort required by one iteration of the Anderson strategy is essentially equal to the one corresponding to one iteration of the WR Algorithm (1). More details on these algorithms and their convergence proofs can be found in [Walker, 2011, Walker and Ni, 2011, Ciaramella and Fabrini, 2021]. The improvement

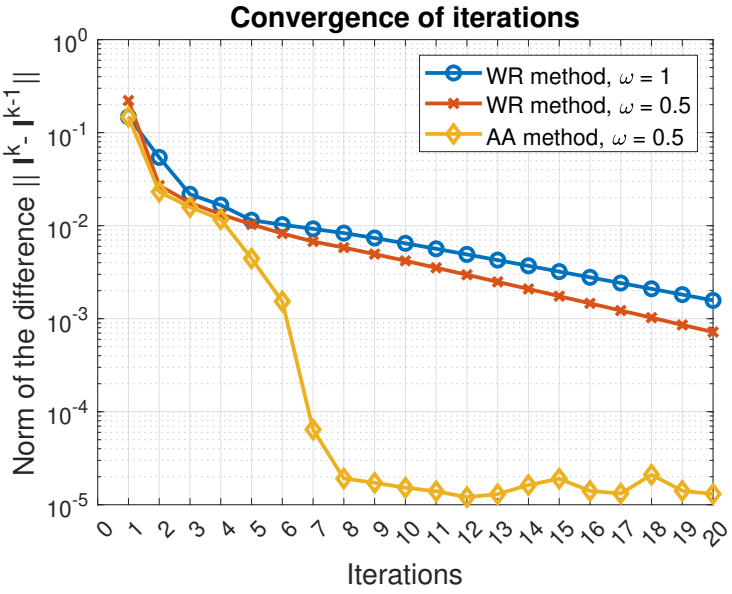


Figure 3.5: Comparison of the convergence for all the presented methods, WR with different values of ω_D and AA.

brought by the AAA method is tremendous and can be appreciated in Figure 3.5, where it is compared with the previously described methods. It is possible to see that for the first iterations it has a similar behavior, but then the gradient of descent becomes much steeper. The stagnation present at the end is due to the tolerance used by the numerical integrator. It was noted that reducing the tolerance, the value at which stagnation is reached becomes lower.

3.4 Simulation results and discussion

In this section it will be shown that the WR method improved with AA can be useful to speed up the simulation of a battery pack with respect to a centralized method

Remark 6 *All the simulations presented here were carried out on a Windows 10 machine with 8 Gbytes of RAM and an Intel® Core™ i7-9750H 2.60 GHz 6-core processor.*

In all the following examples, without loss of generality, the time horizon is set to $[0, 100]$. The value of the total current $I_{tot}(t)$ was chosen proportional to the number of cells in the system and constant $\forall t$ in the considered time interval. The threshold for stopping iterations is set to $\epsilon_{aa} = 10^{-6}$.

The first simulations that are presented in Figure 3.6, refer to the centralized approach and show the scaling of computation times with an increasing number of parallel connected cells. It is possible to note that the scaling is exponential, with a steeper growth for the cases with higher values of P (i.e. the

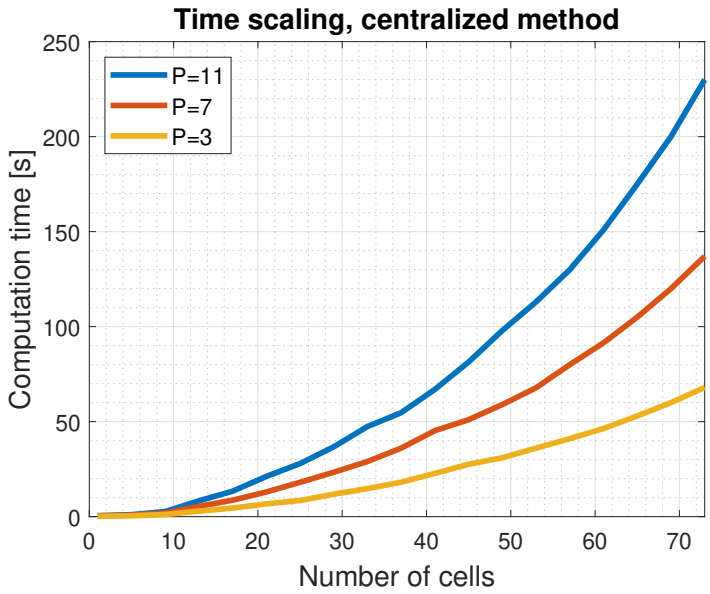


Figure 3.6: Time scaling for a centralized simulation with different values of P with an increasing number of parallel cells.

number of finite volumes used for the discretization of the electrolyte dynamics equation). The number of discretization volumes P was chosen in such a way that further increasing its value would not result in any sensible difference in the behaviour of the other states and of the output voltage[1]. Note that the centralized method uses an algorithm (*ode15*) that is serial and does not benefit of the parallel capabilities of the processor. This was verified by forcing the centralized method to run on a single processor and comparing it with the computation times of simulations where it was allowed to freely allocate resources. No difference was noted. This has been further verified in Figure 3.7, where it is possible to note that for the single subsystem, the computation time is exactly equal for all the curves (i.e. simulating the same system using a different number of processors).

The proposed algorithm has the advantage, with respect to the centralized approach, of solving smaller subproblems in parallel, as $s \ll N$. However, in order for the WR method to be effective, the drawback of having to iterate the resolution of the subproblems to obtain the same current distribution of the original problem has to be taken into account. As explained previously, the algorithm relies on the parallel capabilities of the processor: having up to 6 cores available means that up to 6 subsystems can be solved simultaneously[2]. The advantage of having a parallel algorithm can be appreciated in Figure 3.7, where a single iteration of the WR method is simulated using a different number of processors, having fixed the subsystem size and the overlap ($s = 5$,

[1]Proven experimentally by simulating the same system with different values of P.

[2]If the number of subsystems is greater than the number of cores, they are solved in groups in a sequential way.

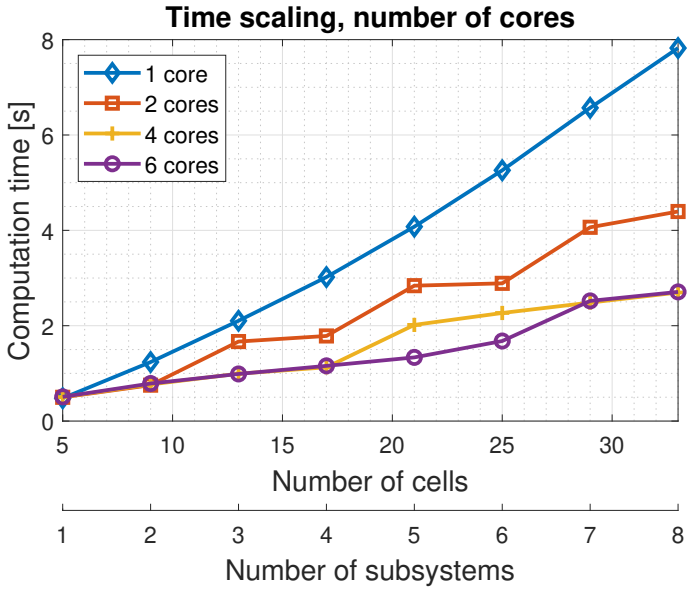


Figure 3.7: Scaling of simulation time for the resolution of a single iteration with increasing number of available cores in the processor. The size of the subsystems is fixed to $s = 5$.

$q = 1$). Note that on the x-axis there is the number of cells N and the number of subsystems a. The blue line represents the single core performance and it is possible to see that it is significantly higher than the others. The other lines show a step behaviour, where there are almost horizontal traits interspersed with steeper increases in computation times. These latter are present when the number of subsystems a becomes equal to a multiple of the number of available cores. As example take the red line, corresponding to a simulation performed with two cores. Every addition of two subsystems there is a step up in the computation time. The same holds for the other lines. Note that the ideal behaviour would exhibit a constant simulation time until the number of subsystems is equal to the number of processors. This does not happen because of parallelization overhead.

Figure 3.7 demonstrates the advantage of using a parallel algorithm, as computation times are largely reduced. In this example, a single iteration of the method was simulated gradually increasing the number of subsystems present, while their size was set to $s = 5$. Potentially, by increasing the number of cores, one would be able to extend the length of the almost horizontal traits, further reducing the computation times for large systems. Note that the choice of P here is not important: increasing or decreasing it would not affect the relative position between the lines.

It is therefore clear that the most benefits are obtained when the number of subsystems is equal to the number of cores. Fixing then $a = 6$ and assuming

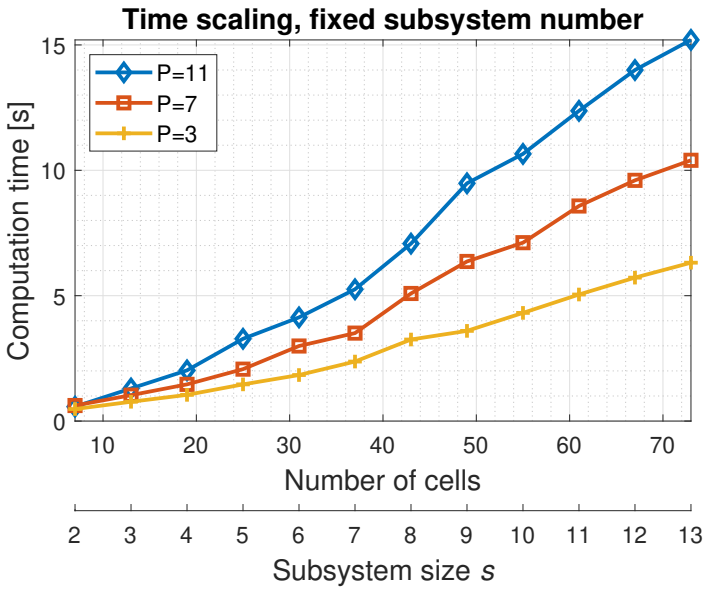


Figure 3.8: Scaling of computation times for the single iteration having fixed $a = 6$ and $q = 1$.

N as known, it is possible to calculate the subsystem size s as

$$s = \frac{N - q}{a} + q, \tag{3.35}$$

with q chosen in such a way that $s \in \mathbb{N}$. With these assumptions, Figure 3.8 shows the scaling of the single WR iteration computation time for different values of P. Note that in this figure, the x-axis still has the number of cells N, but differently from the previous one, it has the subsystem size s. Comparing Figure 3.8 with Figure 3.7, it is possible to note that scaling has become almost linear. To conclude and summarize, the optimal subsystem size s can be chosen in such a way that the number of subsystems a is equal to the number of available processors. For what concerns the choice of the overlap q, its optimal value is always 1, as it cannot be reduced to 0, otherwise convergence would be lost; on the other hand, increasing its value would not provide any benefit to the convergence speed, only increasing the computational burden.

Remark 7 *Memory usage can be a concern in some situations, to ensure that the proposed method is not greedy with respect to a centralized method. In the presented case, it depends exclusively from the value of the overlap q. In fact, the difference between the proposed WR method and a centralized one in terms of memory usage is only in the number of cells that get solved multiple times in the overlap. The choice of q in terms of convergence was previously discussed and it was noted that the optimal choice was having $q = 1$ and it could not be reduced further. This choice is optimal also in terms of memory size, as it reduces the memory needed to the essential minimum.*

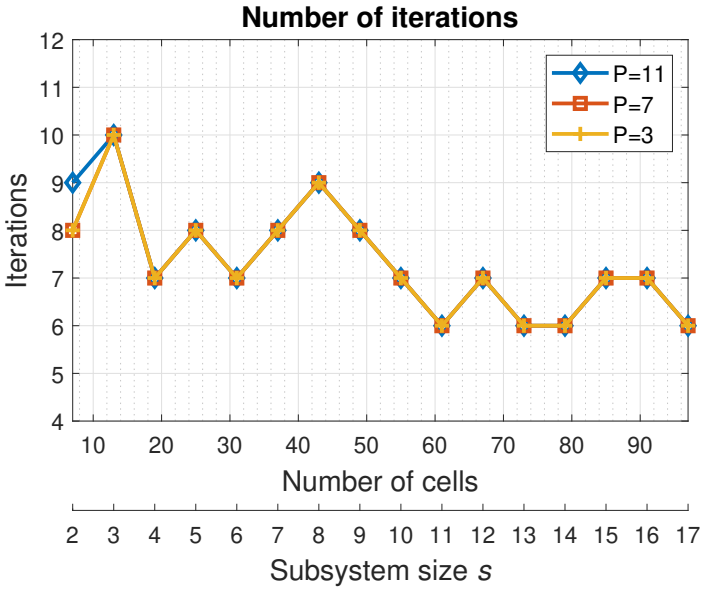


Figure 3.9: Number of iterations as a function of N and s for different values of P.

Now that a reference value for the time cost of each iteration has been set, the focus can move to the analysis of the simulation results for the complete WR method. In particular, it is interesting to analyze the number of iterations of the AA algorithm needed to reach convergence with the desired tolerance. The first observation, looking at Figure 3.9, is that the number of iterations does not grow with system size, instead it stabilizes around 6-7 iterations. The second one is that the number of iterations does not depend on the complexity of the system; this is a very important point, because even though the cost of iterations increases with system size, their number does not seem to be affected. This implies that the method is more efficient as the complexity of the system grows. Finally, in Figure 3.10 a direct comparison between the centralized resolution and the WR method improved with AA is presented for different values of P. What it is possible to see here is that, for small systems, the centralized approach works better in terms of computation times. This is due to the need of iterating the resolution: if the system is small enough there is no advantage in making it smaller. However when the number of cells increases, the proposed method behaves better, leading to reduced computation times. Moreover, it is possible to note that the crossover point, where the WR method becomes better than the centralized approach, shifts to the left when the complexity of the system increases. This proves the effectiveness of the proposed method. Considering a practical example of a Tesla Model S level with 74 cells connected in parallel [Bruen and Marco, 2016], we have that with the proposed method, simulation times would be almost a third of a centralized method in the case with $P = 11$. Instead, having $P = 3$ allows us to halve the simulation time. This was done using 6 cores. Note that

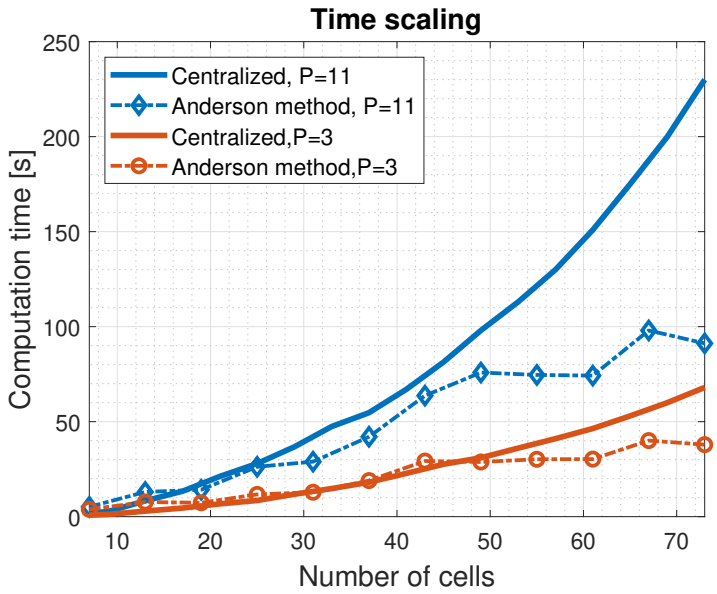


Figure 3.10: Comparison between the centralized method computation times (continuous lines) and the WR method computation times (dotted lines).

there is still room for improvement: if more cores are used for the simulation, this would allow to further reduce the total time. Moreover, we can see that as the model gets more complicated (in this case P was used to increase the number of equations in the system as a proof of concept), the proposed method becomes more efficient. This means that using more complex models (such as DFN, for example), or adding more dynamics, such as complete battery pack temperature dynamics, the advantage would be more evident.

3.5 Conclusions

In this part of the work the use of the Waveform Relaxation method to speed up the simulation of the battery pack was presented and detailed. Convergence could not be obtained by applying a standard Waveform Relaxation, so an appropriate correction step had to be designed. In order to further decrease the simulation time, the Anderson Acceleration strategy was also applied. The results demonstrate the effectiveness of the proposed method, the precision in obtaining the same result of the centralized approach and the convergence to the true global solution. In addition, they can also serve as a proof of concept for other observations: if the simulations are carried out on a system with more processor cores, the time scaling would be even better, as it would be able to solve more subsystems at the same time. Moreover it was shown that this method performs better as the computational complexity grows (in this case increased it by using P). In the next Section, a more complex model is used to further validate the results obtained here. In addition, complete thermal

dynamics are added, that increase both the complexity and the coupling of the dynamical model.

Chapter 4

Fast simulation of DFN-based battery packs with thermal dynamics

This Chapter can be considered an extension of Chapter 3: first of all, instead of using a default Matlab solver (such as *ode15i*), a custom solver was developed, capable of simulating a parallel-connected battery pack. It is based on a quasi-Newton algorithm. This solver is also very flexible, meaning that any cell model can be used as long as input and output are formatted respecting some criteria. This is an upgrade over the previous work because the *ode15i* solver required the implementation of the cell model in a particular syntax, which made it flexible but inconvenient. The results presented here are based on the cell model implemented in TOOFAB [Khalik et al., 2021a], a freely downloadable toolbox that includes a DFN model with some simplifications. In addition, a thermal model for the entire battery pack was included: heat exchange between cells, cooling plates and coolant fluid was modeled in [van de Ven et al., 2022], where measures of heat exchange coefficients have been taken experimentally. The inclusion of this thermal model required modifications to the WR algorithm because besides the current flowing in the parallel branches, also the temperature is an independent variable. Moreover, the results are updated and extended with new observations and considerations.

4.1 Battery pack model and cell parameters

The electrical model considered in this work relies is a simplified version of the previously presented DFN model. In particular, it is the implementation used in TOOFAB (TOOlbox for FAst Battery simulation), freely available at: https://github.com/Zuan-Khalik/TOOFAB. Since no major modification was made with respect to the original work [Khalik et al., 2021a, Khalik et al., 2021b], only some important information is reported here, for the full details please refer to [Khalik et al., 2021a, Khalik et al., 2021b]. Moreover, the

full formulation of the DFN model was provided in Section 2.4.2, so only the simplified model is reported here. Still referring to Figure 2.12, the equations representing the electrical dynamical model are the following ones. Note that here time and space dependency have been neglected where possible for clarity of notation. The solid-phase Li-ion concentration is denoted by c_s and governed by the differential equation

$$\frac{\partial c_s}{\partial t} = \frac{D_s}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial c_s}{\partial r}\right), \tag{4.1}$$

with boundary conditions

$$\left.\frac{\partial c_s}{\partial r}\right|_{r=0} = 0, \quad \left.-D_s\frac{\partial c_s}{\partial r}\right|_{r=R_s} = j_n. \tag{4.2}$$

The solid phase diffusion is approximated with a two-parameter polynomial expression, assuming that the concentration profile within a single particle has the shape of a parabola over r. This has the advantage of making the solid phase Li-ion concentration c_s, that is the only variable depending from r, a function of only z and t (the dependence from z is implicit in j_n). This results in

$$\frac{dc_s}{dt} = -\frac{3}{R_s}j_n; \quad \bar{c_s} = c_s - \frac{R_s}{5D_s}j_n. \tag{4.3}$$

Moreover, c_s, is considered to be constant over z. The electrolyte phase Li-ion concentration c_e is governed by

$$\epsilon_e\frac{\partial c_e}{\partial t} = \frac{\partial}{\partial z}\left(D_e{\epsilon_e}^{p_b}\frac{\partial c_e}{\partial z} + \frac{3\epsilon_s\left(1-t_+^{0\,0}\right)}{R_s}\right) \tag{4.4}$$

with boundary conditions

$$\left.\frac{\partial c_e}{\partial r}\right|_{z=0} = \left.\frac{\partial c_e}{\partial r}\right|_{z=L_c} = 0. \tag{4.5}$$

The solid phase potential ϕ_s is defined as

$$\frac{\partial}{\partial z}\left(\sigma_c\epsilon_s\frac{\partial\phi_s}{\partial z}\right) = \frac{3\epsilon_s F}{R_s}j_n, \tag{4.6}$$

with boundary conditions

$$\left.\sigma_c\epsilon_s\frac{\partial\phi_s}{\partial z}\right|_{z=0} = \left.\sigma_c\epsilon_s\frac{\partial\phi_s}{\partial z}\right|_{z=L_c} = \frac{I}{A_{surf}}, \ \left.\frac{\partial\phi_s}{\partial z}\right|_{z=\delta_n} = \left.\frac{\partial\phi_s}{\partial z}\right|_{z=L_c-\delta_n} = 0, \tag{4.7}$$

while the electrolyte phase potential ϕ_e is

$$\frac{\partial}{\partial z}\left(\kappa_c {\epsilon_e}^{p_b}\frac{\partial \phi_e}{\partial z} + \kappa_c {\epsilon_e}^{p_b}\nu_t \frac{2RT}{F}\frac{\partial \ln c_e}{\partial z}\right) = -\frac{3\epsilon_s F}{R_s} j_n, \tag{4.8}$$

the boundary conditions being

$$\left.\frac{\partial \phi_e}{\partial z}\right|_{z=0} = \left.\frac{\partial \phi_e}{\partial z}\right|_{z=L_c} = 0. \tag{4.9}$$

Butler-Volmer kinetics are given by

$$j_n = \frac{i_0}{F}\left(\exp\left(\frac{\alpha_a F}{RT}\eta_e\right) - \exp\left(-\frac{\alpha_c F}{RT}\eta_e\right)\right), \tag{4.10}$$

that can be linearized around the origin with respect to η_e, by observing that $\frac{F}{RT} >> |\eta_e|$. Moreover, note that $\alpha_a + \alpha_c = 1$, which allows to rewrite Butler-Volmer kinetics as

$$j_n = \frac{i_0}{RT}\left(\phi_s - \phi_e - U\right). \tag{4.11}$$

Exchange current density and electrode over-potential are expressed respectively as

$$i_0 = k_0 {c_e}^{\alpha_a}\left({c_s}^{\max} - \bar{c_s}\right)^{\alpha_a}{\bar{c_s}}^{\alpha_c}, \tag{4.12}$$

$$\eta_e = \phi_s - \bar{\phi_e} - U, \tag{4.13}$$

where $\bar{\phi_e}$ is the electrode mean electrolyte potential.

In addition to simplifications listed above, concentration-dependent parameters (κ_c, D_e, D_s, ν_t and $c_s/{c_s}^{\max}$) are approximated using a zero-order Taylor approximation; the evaluation point is chosen either dynamically (based on the previous time sample, after discretization), or as a constant value, effectively making these parameters constant over space and time.

Based on the quantities up to this point, it is possible to define the output of the system. Terminal voltage is calculated as

$$V = \phi_s(L_c) - \phi_s(0) - \frac{R_f}{A_{surf}} I. \tag{4.14}$$

The input of the system is the current flowing in the cell, where a positive I indicates charging. All the quantities and constants used in the previous equations are defined in Table 4.1. Besides physical constants, the majority of the DFN parameters have to be determined. As proven widely in literature [Valøen and Reimers, 2005], many of them are not constant and in fact depend on the concentration of Li-ions. The concentration-dependent parameters and their formulation have been taken from [Ecker et al., 2015c, Ecker et al., 2015b]. The thermal and aging models are unchanged from what was

A_{surf}	Active electrode area	t	Time
c_e	Li-ion electrolyte concentration	T	Cell temperature
c_s	Li-ion solid concentration	t_+^0	Transference number
$\bar{c_s}$	Concentration at the SEI	U	Electrode equilibrium potential
${c_s}^{\max}$	Maximum concentration	V	Terminal voltage
D_e	Electrolyte diffusion coefficient	z	Position across cell
D_s	Solid diffusion coefficient	α_a	Anodic charge-transfer coefficient
F	Faraday's constant	α_c	Cathodic charge-transfer coefficient
I	Cell current	δ_n	Negative electrode thickness
i_0	Exchange current density	δ_p	Positive electrode thickness
j_n	Net molar flux of Li-ions	ϵ_s	Active particles volume fraction
k_0	Kinetic constant	ϵ_e	Electrolyte volume fraction
L_c	Cell thickness	η_e	Electrode overpotential
p_b	Bruggeman porosity exponent	κ_c	Ionic conductivity
r	Radial position across particle	ν_t	Thermodynamic factor
R	Universal gas constant	σ_c	Electrical conductivity
R_f	Empirical film resistance	ϕ_e	Electrolyte phase potential
R_s	Radius of active material	ϕ_s	Solid phase potential

Table 4.1: List of symbols and constants used to define the DFN model.

previously presented for the single cell, refer to Equations (2.33) and (2.34).

4.1.1 Discretization

Taking into account the described equations, their respective boundary conditions and the simplifications made, the whole model is a set of DAEs. In order to reduce the system to a set of AEs, the discretization method presented in [Xia et al., 2017b] is applied; the only difference is the electrolyte discretization, where the FVM used in [Torchio et al., 2016] is implemented. The details of discretization are not presented in this part of the work for compactness; the interested reader can find the full formulation in [Khalik et al., 2021a, Xia et al., 2017b, Torchio et al., 2016]. Moreover, TOOFAB is used as a black box to simulate the single cell and the proposed method is built on top. After discretization, the system of DAEs will be in the form of a set of AEs, that is solved using a modified Newton's method. The states of the system are

$$\mathbf{x} = \left[c_s \; c_e \; \phi_s \; \phi_e\right]^\top, \tag{4.15}$$

that are the concentrations in respectively the solid and electrolyte phases and the potentials. In a similar way, define the input of the system, that is the current $\mathbf{u} = I$. The output is defined instead as the terminal voltage, heat generation and temperature

$$\mathbf{y} = \left[V \; Q \; T\right]^\top. \tag{4.16}$$

Given an initial time instant $t_0 = 0$, a final time instant t_f and a sampling time t_s, it is possible to write the discrete-time model of the single cell for each time instant k_t

$$\begin{cases} \mathbf{x}(k_t + 1) = f\left(\mathbf{x}(k_t), \mathbf{u}(k_t)\right), & k_t \geq 1 \\ \mathbf{y}(k_t) = g(\mathbf{x}(k_t), \mathbf{u}(k_t)), & k_t \geq 0. \end{cases} \tag{4.17}$$

Note that f comprises all the discretized model equations (Equations (2.16) to (2.26)), while g defines the output of the system and contains the cell terminal voltage V, as defined in (2.28), heat generation Q and cell temperature T, given respectively by (2.29) and (2.33). To clarify the notation used in the following, bold letters will be used to denote variables at every time instant of the considered horizon, e.g. $\mathbf{V}_1 = [V_1(t = t_0)\ V_1(t = t_s)\ V_1(t = 2t_s) \dots V_1(t = t_f)]^\top$. The choice of the sampling time is a typical one of having $t_s = 1\ s$. For all the simulations carried out for this work, this was enough to avoid convergence problems. The number of samples in the considered horizon will be denoted with t_t and calculated as

$$t_t = \frac{t_f - t_0}{t_s}. \tag{4.18}$$

Consider now a parallel-connected battery pack composed of N; note how the relationship governing the parallel connection, holds also in discrete time

$$\begin{cases} \sum_{h=1}^{N} \mathbf{I}_h - \mathbf{I}_{tot} = 0, \\ \mathbf{V}_h - \mathbf{V}_{h-1} = 0, \quad \forall\ h \in \{2, \dots, N\}. \end{cases} \tag{4.19}$$

$\mathbf{I}_{tot}$ is considered to be the known input of the system and is defined over $[t_0, t_f]$ at every sample time t_s. Having discretized the dynamics of the system, the resulting problem, combined with the algebraic boundaries, is a set of AEs.

4.1.2 Battery pack thermal model

A full thermal model is implemented, taking into account conduction, convection and radiating heat transport between cells and the surrounding elements. The thermal model presented in this Section is the one presented in [van de Ven et al., 2022]. The topology of the battery pack under consideration is represented in Figure 4.1. In this example, it is possible see 9 cells that produce heat (in orange), each one having 6 neighbours, either other active cells or insulation material considered to be at ambient temperature T_f. Consider the purple section line, centered around cells 3 and 6. In Figure 4.2 it is possible to see the thermal conductivity that connect it to other elements: q_{cc} is the cell-to-cell thermal conductivity, q_{ca} is the cell to ambient thermal conductivity; top and bottom plate are represented in yellow and are connected to cell 6 through q_{cp}, these plates are then thermally connected to the external ambient

respectively through q_{ta} and q_{ba}. Given the h-th cell, its dynamics are given

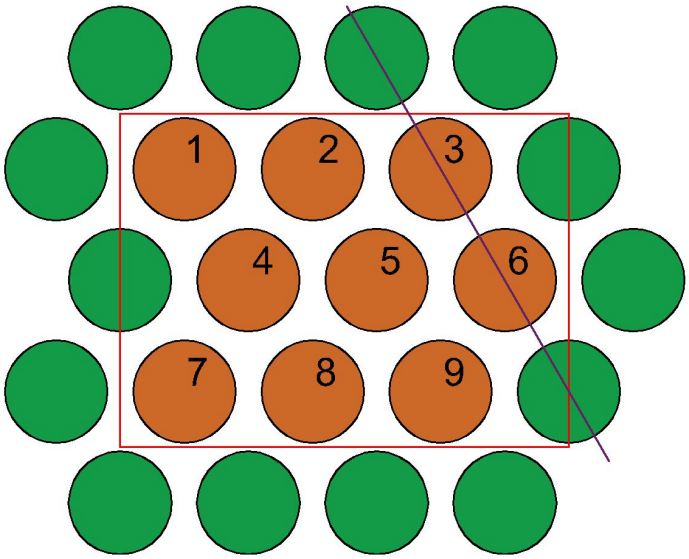


Figure 4.1: Topology example of a battery pack showing neighbouring cells.

by

$$\frac{d\mathbf{T}_h}{dt} = \frac{1}{C_{c,h}} \sum_{v \in \mathcal{V}} \mathbf{q} \left(\mathbf{T}_v - \mathbf{T}_h \right) + \mathbf{Q}_h, \tag{4.20}$$

where $\mathbf{T}_h$ is the temperature of the h-th cell, $C_{c,h}$ its thermal capacity and $\mathbf{Q}_h$ the heat it generates. The subscript v represents all the surrounding objects at temperature $\mathbf{T}_v$ that cell h is exchanging heat with. The heat exchange coefficient $\mathbf{q}$ varies based on which type of thermal exchange is happening. Here for clarity of notation has been left general. The set $\mathcal{V}$ contains all the cells in the battery pack, top and bottom plate and the ambient, in such a way that $v \in \mathcal{V} = \{1, 2, \ldots, N, b, t, amb\}$. To extend the model to the N cells of a battery pack, define the vectors

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 & \cdots & \mathbf{Q}_N \end{bmatrix}^\top, \tag{4.21}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 & \cdots & \mathbf{T}_N & \mathbf{T}_b & \mathbf{T}_t \end{bmatrix}^\top, \tag{4.22}$$

which can be used to define the continuous-time state-space model that comprises Equation (4.20)

$$\frac{d\mathbf{T}}{dt} = A_T \mathbf{T} + B_Q \mathbf{Q} + B_a T_f, \tag{4.23}$$

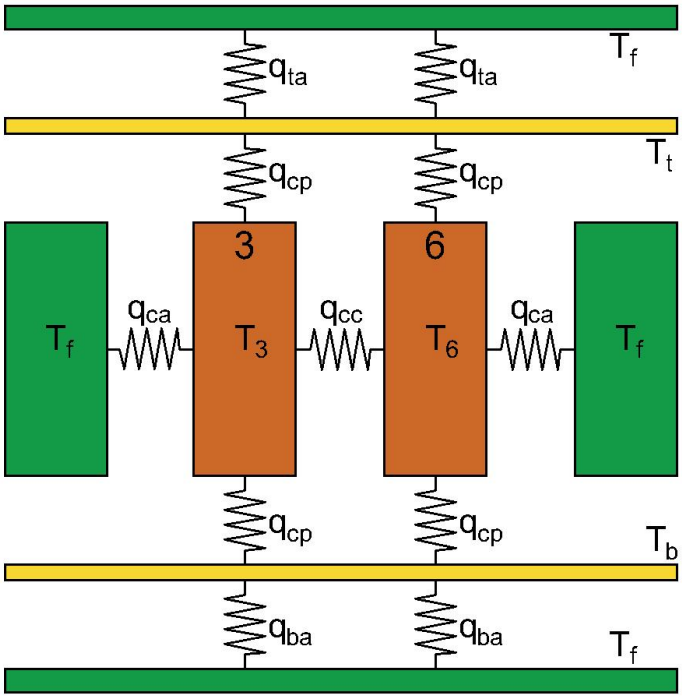


Figure 4.2: Section of the battery pack of Figure 4.1 representing the thermal connections between cells and ambient.

with with

$$A_T = \begin{bmatrix} -\frac{q_{cc}}{C_c}\mathcal{L} - \frac{q_{ca}}{C_c}\Phi_a - \frac{2q_{cp}}{C_c}\mathbb{I}_N & \frac{q_{cp}}{C_c}\mathbf{1}_N & \frac{q_{cp}}{C_c}\mathbf{1}_N \\ \frac{q_{cp}}{C_p}\mathbf{1}_N^\top & \frac{-Nq_{cp}-q_{ta}}{C_p} & 0 \\ \frac{q_{cp}}{C_p}\mathbf{1}_N^\top & 0 & \frac{-Nq_{cp}-q_{ba}}{q_{cp}} \end{bmatrix}, \tag{4.24}$$

$$B_Q = \begin{bmatrix} \frac{1}{C_c}\mathbb{I}_N & 0 & 0 \end{bmatrix}^\top, \tag{4.25}$$

$$B_a = \begin{bmatrix} \frac{q_{ca}}{C_c}\Phi_a\mathbf{1}_N & \frac{q_{ta}}{C_p} & \frac{q_{ba}}{C_p} \end{bmatrix}^\top, \tag{4.26}$$

where $\mathbb{I}_N$ is an identity matrix of dimension N and $\mathbf{1}_N$ a unity vector of N elements. The $\mathcal{L}$ matrix is the Laplacian as used in [Rath et al., 2020], where a weighted undirected graph is used to represent the different thermal conductivity, and is defined as

$$\mathcal{L} = \mathcal{D} - \mathcal{A}. \tag{4.27}$$

The matrix $\mathcal{D}$ is the degree matrix, of dimension $N \times N$, having on its diagonal at element $\mathcal{D}(h,h)$, the number of cells connected to cell h. The adjacency matrix $\mathcal{A}$ indicates whether there is thermal coupling between two different nodes in such a way that if two cells h and v are connected together, it results $\mathcal{A}(h,v) = \mathcal{A}(v,h) = 1$. C_c and C_p are respectively the cell thermal capacity

and the plates thermal capacity. Finally, define the ambient spot matrix Φ_a as

$$\Phi_a = (6\mathbb{I}_N - \mathcal{D}), \tag{4.28}$$

representing the interactions the the surrounding ambient. Still referring to [van de Ven et al., 2022], the model gets then discretized using forward-Euler

$$\mathbf{T}(k_t + 1) = (\mathbb{I}_{N+2} + t_s A_T)\,\mathbf{T}(k_t) + t_s B_Q \mathbf{Q}(k_t) + t_s B_a T_f(k_t), \tag{4.29}$$

with t_s being the sampling time. This is a fully discrete system and therefore contains a set of AEs.

Combining the electrical model equations (4.17) with the algebraic constraints (4.19) and thermal dynamics (4.29), the full model of the battery pack under consideration is defined. Note that the unknowns of this system are considered to be the current $\mathbf{I}$ and the temperature $\mathbf{T}$; once these are known over the full horizon, then the system is fully defined. The inputs, considered to be known over the full horizon, are the current flowing in the battery pack $\mathbf{I}_{tot}$ and the ambient temperature T_f.

4.2 Quasi-Newton solver

Consider the time-discrete model of the battery pack just presented. Define the unknown vector $\mathbf{X}$ as

$$\mathbf{X} = \begin{bmatrix}\mathbf{I}\ \mathbf{T}\end{bmatrix}^\top, \tag{4.30}$$

with

$$\mathbf{I} = \begin{bmatrix}\mathbf{I}_1\ \mathbf{I}_2\ \cdots\ \mathbf{I}_N\end{bmatrix}^\top, \tag{4.31}$$

$$\mathbf{T} = \begin{bmatrix}\mathbf{T}_1\ \mathbf{T}_2\ \cdots\ \mathbf{T}_N\ \mathbf{T}_b\ \mathbf{T}_t\end{bmatrix}^\top, \tag{4.32}$$

where the current flowing in each cell is denoted with $\mathbf{I}_h, \forall h \in 1, \ldots, N$, the temperature with $\mathbf{T}_h, \forall h \in 1, \ldots, N$ and, finally, top and bottom plate temperature respectively with $\mathbf{T}_t$ and $\mathbf{T}_b$. Note that each element is a vector containing a number of elements equal to the samples in the considered horizon. Remember that once these quantities are known, the system is fully defined and the other states can be calculated accordingly. The set of AEs obtained after discretizing the system and including algebraic boundaries can be solved using any root-finding algorithm. In particular, the equation to be solved is

$$\mathcal{F}(\mathbf{X}) = 0, \tag{4.33}$$

with the function $\mathcal{F}$ defined as follows

$$\mathcal{F}(\mathbf{X}) = \begin{bmatrix} \sum_{h=1}^{N} \mathbf{I}_h - \mathbf{I}_{tot} \\ \mathbf{V}_1 - \mathbf{V}_2 \\ \mathbf{V}_2 - \mathbf{V}_3 \\ \vdots \\ \mathbf{V}_{N-1} - \mathbf{V}_N \\ \mathbf{T}_1 - \hat{\mathbf{T}}_1 \\ \mathbf{T}_2 - \hat{\mathbf{T}}_2 \\ \vdots \\ \mathbf{T}_N - \hat{\mathbf{T}}_N \\ \mathbf{T}_b - \hat{\mathbf{T}}_b \\ \mathbf{T}_t - \hat{\mathbf{T}}_t \end{bmatrix}. \tag{4.34}$$

As shown in the electrical model section, $\mathbf{V}_h$ is a function of $\mathbf{I}_h$ and $\mathbf{T}_h$ and is calculated using TOOFAB.

A further clarification is instead needed for the temperatures. Here a slight modification had to be introduced in order to solve this problem. While the first $(1+N)$ rows of $\mathcal{F}(\mathbf{X})$ have a physical meaning, being the Kirchhoff's laws, the last $(N+2)$ rows (corresponding to the temperatures) are more artificial. The first term $\mathbf{T}_v, \forall v \in \{1, \ldots, N, b, t\}$ comes directly from TOOFAB and represents the temperature uncoupled from the other elements in the battery pack; $\hat{\mathbf{T}}_v, \forall v \in \{1, \ldots, N, b, t\}$ instead is computed using the heat generation of each cell $\mathbf{Q}_h, \forall h \in \{1, \ldots, N\}$ as input of the thermal model described in Equation (4.29).

A single fixed-point iteration is defined as

$$\mathbf{X}^{n+1} = \mathbf{X}^n - \mathcal{J}(\mathbf{X}^n)^{-1}\mathcal{F}(\mathbf{X}^n), \tag{4.35}$$

with $n \in 1, \ldots, N_N$ being the iteration number and N_N the maximum number of iterations allowed.

$\mathcal{J}(\mathbf{X}^n)$ is the Jacobian of the function $\mathcal{F}(\mathbf{X})$. In order to simplify the problem and have a faster simulation, the Jacobian is written explicitly and simplified, where possible. Note that the derivatives are computed at each time instant. For compactness of notation, a vectorial notation is used. It can be logically divided in 4 sub-matrices

$$\mathcal{J}(\mathbf{X}^n) = \begin{bmatrix} \mathcal{J}_1 & \mathcal{J}_2 \\ \mathcal{J}_3 & \mathcal{J}_4 \end{bmatrix}. \tag{4.36}$$

- $\mathcal{J}_1$ contains the partial derivatives of the Kirchhoff equilibrium (voltage and current) with respect to current. The dimension of $\mathcal{J}_1$ is $((N \times t_t) \times$

$(N \times t_t))$.

$$\mathcal{J}_1 = \begin{bmatrix} \mathbb{I}_{t_t} & \mathbb{I}_{t_t} & \mathbb{I}_{t_t} & \mathbb{I}_{t_t} & \cdots & \mathbb{I}_{t_t} & \mathbb{I}_{t_t} \\ \frac{\partial \mathbf{V}_1}{\partial \mathbf{I}_1} & -\frac{\partial \mathbf{V}_2}{\partial \mathbf{I}_2} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \mathbf{0}_{t_t} & \frac{\partial \mathbf{V}_2}{\partial \mathbf{I}_2} & -\frac{\partial \mathbf{V}_3}{\partial \mathbf{I}_3} & \mathbf{0}_{t_t} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \frac{\partial \mathbf{V}_3}{\partial \mathbf{I}_3} & -\frac{\partial \mathbf{V}_4}{\partial \mathbf{I}_4} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \frac{\partial \mathbf{V}_4}{\partial \mathbf{I}_4} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \cdots & \frac{\partial \mathbf{V}_{N-1}}{\partial \mathbf{I}_{N-1}} & -\frac{\partial \mathbf{V}_N}{\partial \mathbf{I}_N} \end{bmatrix}. \tag{4.37}$$

The quantity $\mathbf{0}_{t_t}$ indicates a matrix of dimension $t_t \times t_t$ of zeros. Notice how the first row is just composed of identity matrices, because $\mathbf{I}_{tot}$ is a constant. The reasonable assumption made here is that each voltage $\mathbf{V}_h, \forall h = 1, \ldots, N$ depends exclusively from the current $\mathbf{I}_h, \forall h = 1, \ldots, N$. This allows to eliminate a big part of this matrix.

- $\mathcal{J}_2$ contains the partial derivative of the Kirchhoff equilibrium (voltage and current) with respect to temperature.

$$\mathcal{J}_2 = \begin{bmatrix} \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \frac{\partial \mathbf{V}_1}{\partial \mathbf{T}_1} & -\frac{\partial \mathbf{V}_2}{\partial \mathbf{T}_2} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \mathbf{0}_{t_t} & \frac{\partial \mathbf{V}_2}{\partial \mathbf{T}_2} & -\frac{\partial \mathbf{V}_3}{\partial \mathbf{T}_3} & \mathbf{0}_{t_t} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \frac{\partial \mathbf{V}_3}{\partial \mathbf{T}_3} & -\frac{\partial \mathbf{V}_4}{\partial \mathbf{T}_4} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \frac{\partial \mathbf{V}_4}{\partial \mathbf{T}_4} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \cdots & \frac{\partial \mathbf{V}_{N-1}}{\partial \mathbf{T}_{N-1}} & -\frac{\partial \mathbf{V}_N}{\partial \mathbf{T}_N} & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \end{bmatrix}. \tag{4.38}$$

The dimension of $\mathcal{J}_2$ is $((N \times t_t) \times ((N+2) \times t_t))$. Again, notice how the first row is composed of zeroes, because the current does not depend on the temperature and the last two columns are zeros as well, because the voltages do not depend on $\mathbf{T}_b$ and $\mathbf{T}_t$.

- $\mathcal{J}_3$ contains the partial derivative of the temperature component with respect to current.

$$\mathcal{J}_3 = \begin{bmatrix} \frac{\partial}{\partial \mathbf{I}_1}(\mathbf{T}_1 - \hat{\mathbf{T}}_1) & \mathbf{0}_{t_t} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \mathbf{0}_{t_t} & \frac{\partial}{\partial \mathbf{I}_2}(\mathbf{T}_2 - \hat{\mathbf{T}}_2) & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \cdots & \mathbf{0}_{t_t} & \frac{\partial}{\partial \mathbf{I}_N}(\mathbf{T}_N - \hat{\mathbf{T}}_N) \\ \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \\ \mathbf{0}_{t_t} & \mathbf{0}_{t_t} & \cdots & \mathbf{0}_{t_t} & \mathbf{0}_{t_t} \end{bmatrix}. \tag{4.39}$$

The dimension of $\mathcal{J}_3$ is $(((N+2)\times t_t)\times(N\times t_t))$. The last two rows are null because of the assumption that the current contribution to the derivative of top and bottom plate temperatures is negligible.

- $\mathcal{J}_4$ contains the partial derivative of the temperature component with respect to temperature.

$$\mathcal{J}_4 = \text{diag}\begin{bmatrix} \frac{\partial}{\partial \mathbf{T}_1}(\mathbf{T}_1 - \hat{\mathbf{T}}_1) \\ \frac{\partial}{\partial \mathbf{T}_2}(\mathbf{T}_2 - \hat{\mathbf{T}}_2) \\ \vdots \\ \frac{\partial}{\partial \mathbf{T}_N}(\mathbf{T}_N - \hat{\mathbf{T}}_N) \\ \frac{\partial}{\partial \mathbf{T}_b}(\mathbf{T}_b - \hat{\mathbf{T}}_b) \\ \frac{\partial}{\partial \mathbf{T}_t}(\mathbf{T}_t - \hat{\mathbf{T}}_t). \end{bmatrix} \tag{4.40}$$

The dimension of $\mathcal{J}_4$ is $(((N+2)\times t_t)\times((N+2)\times t_t)))$.

Since the system is discrete, each derivative at every time instant is calculated using the finite difference between successive time steps. The fixed-point algorithm is reported in Algorithm 4.

Remark 8 *A small clarification is needed on how the derivatives are calculated in the algorithm. Consider the dynamics of the single cell h, given by*

$$[\mathbf{V}_h, \mathbf{Q}_h, \mathbf{T}_h] = DFN_h(\mathbf{I}_h\ \mathbf{T}_h). \tag{4.41}$$

Define also the same output quantities calculated using an input shifted by δ

$$[\delta(\mathbf{V}_h)_{\mathbf{I}}, \delta(\mathbf{Q}_h)_{\mathbf{I}}, \delta(\mathbf{T}_h)_{\mathbf{I}}] = DFN_h((\mathbf{I}_h + \delta_w)\ \mathbf{T}_h), \tag{4.42}$$

$$[\delta(\mathbf{V}_h)_{\mathbf{T}}, \delta(\mathbf{Q}_h)_{\mathbf{T}}, \delta(\mathbf{T}_h)_{\mathbf{T}}] = DFN_h(\mathbf{I}_h\ (\mathbf{T}_h + \delta_w)). \tag{4.43}$$

The notations $\delta(\cdot)_{\mathbf{I}}$ and $\delta(\cdot)_{\mathbf{T}}$ denote output quantities calculated using a shifted input. Then the derivatives are calculated, for example, as

$$\frac{\partial \mathbf{V}_h}{\partial \mathbf{I}_h} = \frac{\delta(\mathbf{V}_h)_{\mathbf{I}} - \mathbf{V}_h}{\delta_w}. \tag{4.44}$$

At this point, a tool capable of simulating a battery pack composed on N cells given the current $\mathbf{I}_{tot}$ and an initial guess $\mathbf{X}^0$ has been built. This solution method will be referred to in the rest of this work as *centralized* resolution and the vector $\mathbf{X}$ returned from the algorithm will be taken as the reference result.

4.2.1 Simulation examples

In this Section, some simulation examples are given. One clarification has to be made regarding the difference among cells: if they were to be equal one another, the solution of the problem would be trivial, the current would split

Algorithm 4 (Fixed-point root-finding algorithm)

Require: $\mathbf{I}_{tot}$, a finite difference δ_w, a maximum number of iterations N_N and a tolerance ϵ_n.

Set $n = 1$ and $\mathcal{F}(\mathbf{X}^0) \leftarrow \inf$.
while $||\mathcal{F}(\mathbf{X}^n) - \mathcal{F}(\mathbf{X}^{n-1})|| > \epsilon_n$ and $n < N_N$ **do**
  for all $h \in \{1, \dots, N\}$ **do**
    Compute $\mathbf{V}_h$, $\mathbf{Q}_h$ and $\mathbf{T}_h$ using DFN_h with input $\dot{\mathbf{x}}_h^{n-1} = \left[\mathbf{I}_h^{n-1}\ \mathbf{T}_h^{n-1}\right]^\top$.
    Compute $\delta(\mathbf{V}_h)_\mathbf{I}$, $\delta(\mathbf{Q}_h)_\mathbf{I}$ and $\delta(\mathbf{T}_h)_\mathbf{I}$ using DFN_h with input $\dot{\mathbf{x}}_h^{n-1} = \left[(\mathbf{I}_h^{n-1} + \delta_w)\ \mathbf{T}_h^{n-1}\right]^\top$.
    Compute $\delta(\mathbf{V}_h)_\mathbf{T}$, $\delta(\mathbf{Q}_h)_\mathbf{T}$ and $\delta(\mathbf{T}_h)_\mathbf{T}$ using DFN_h with input $\dot{\mathbf{x}}_h^{n-1} = \left[\mathbf{I}_h^{n-1}\ (\mathbf{T}_h^{n-1} + \delta_w)\right]^\top$.
  end for
  Compute $\tilde{\mathbf{T}}$ using (4.29) with input $\mathbf{Q}$.
  Compute $\delta(\tilde{\mathbf{T}})_\mathbf{I}$ using (4.29) with input $\delta(\mathbf{Q})_\mathbf{I}$.
  Compute $\delta(\tilde{\mathbf{T}})_\mathbf{T}$ using (4.29) with input $\delta(\mathbf{Q})_\mathbf{T}$.
  Compute $\mathcal{F}(\mathbf{X}^n)$ and $\mathcal{J}(\mathbf{X}^n)$.
  Solve $\mathbf{X}^n = \mathbf{X}^{n-1} - \mathcal{J}(\mathbf{X}^{n-1})^{-1}\mathcal{F}(\mathbf{X}^{n-1})$.
end while

evenly among them. In reality what happens is that manufacturing processes, different aging and balancing issues in the battery pack, create some slight difference between the cells, resulting in a difference performance and, most importantly, a different current distribution (in the case of a parallel connection). To reproduce this behaviour, a variance has been introduced on the cell capacity. In all the examples present in this work, this value has been taken randomly in an interval $\pm 10\%$ for each cell.

In Figure 4.4c an example of the simulation of a battery pack is given. In particular the topology of the battery pack has been chosen with $N = 15$, with 3 rows composed of 5 cells each. Note how terminal voltage (in Figure 4.4a) is equal for all the cells, as required by Kirchhoff's laws. In Figure 4.3 instead, it is possible to notice the different current distribution between the different cells; the same holds for SOC and temperature. Moreover, it is also possible to see how the difference between cells affects the internal states. In Figure 4.5, ϕ_e, c_e and j_n are represented for the different cells at two different time instants, $t = 300\ s$ and $t = 700\ s$. Note that these are spatial distributions along the length of the cell. Here each line corresponds to a different cell. Additionally, notice how j_n is defined only in the electrodes and not in the separator. The vertical lines here represent the spatial division between the elements of the cell, cathode, separator and anode.

As it is possible to see from Algorithm 4, an evaluation of the precision level reached by the algorithm is the norm of the difference of $\mathcal{F}$ at successive

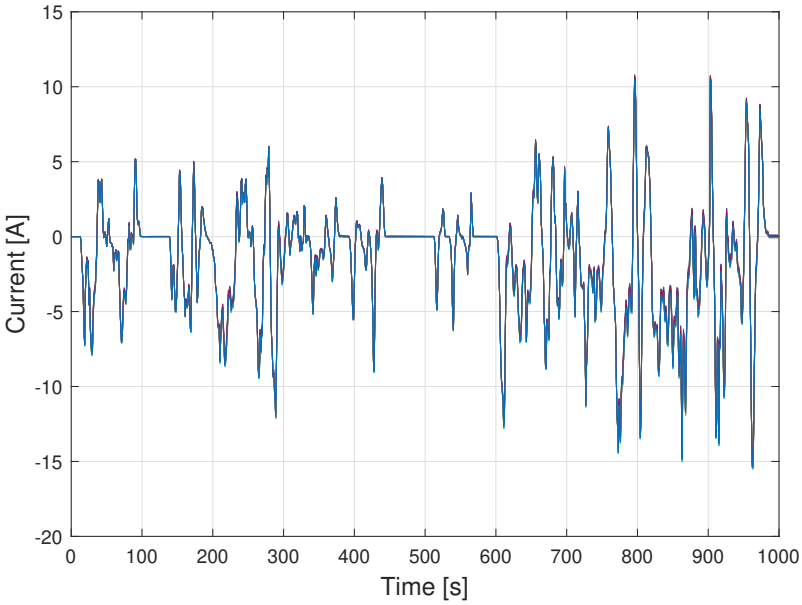
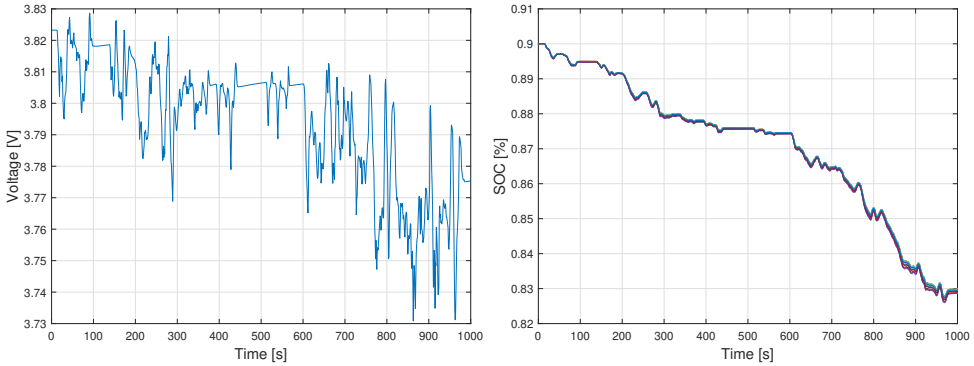
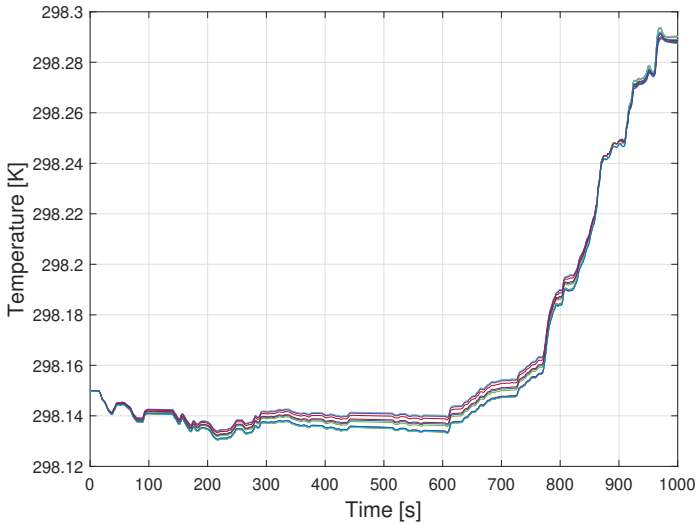


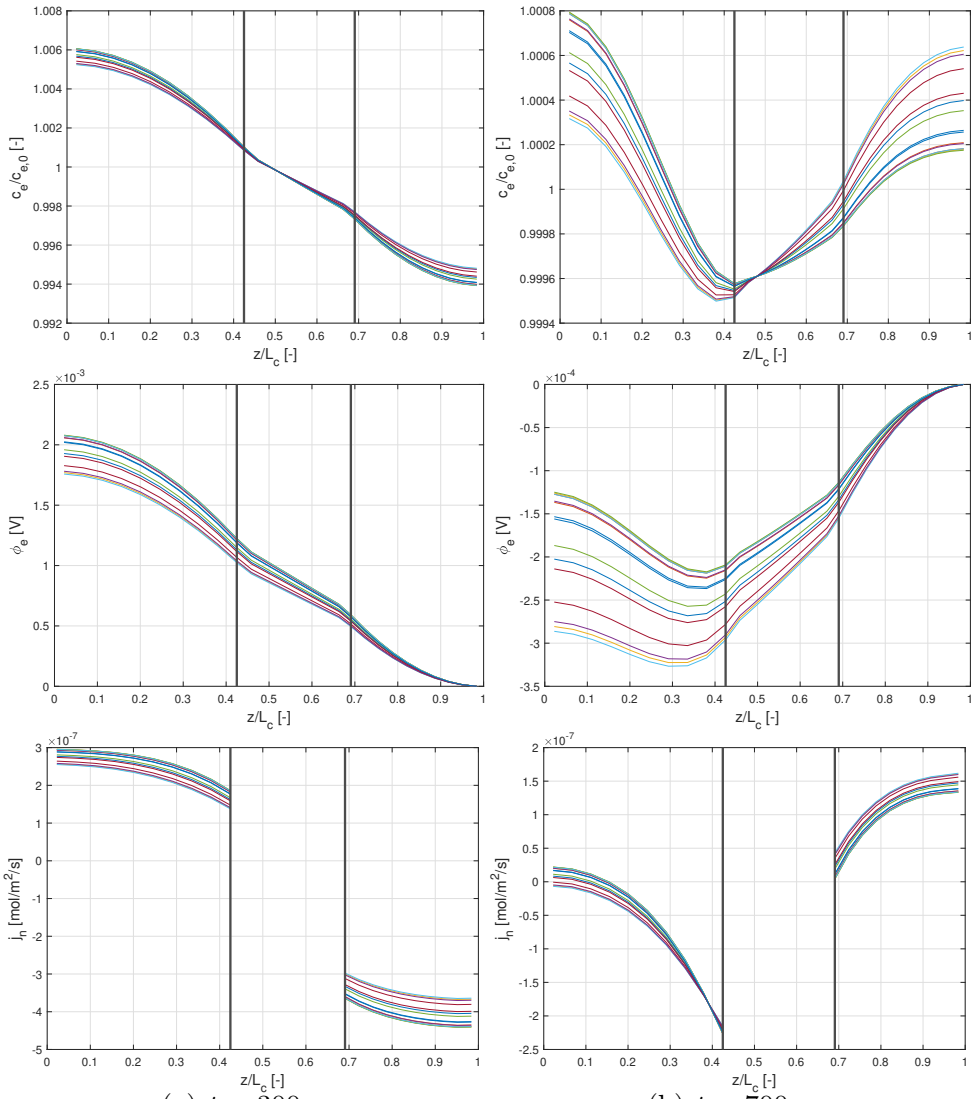
Figure 4.3: Cell current.



(a) Terminal voltage. (b) Cell SOC.



(c) Cell temperature.



(a) $t = 300\ s$ (b) $t = 700\ s$

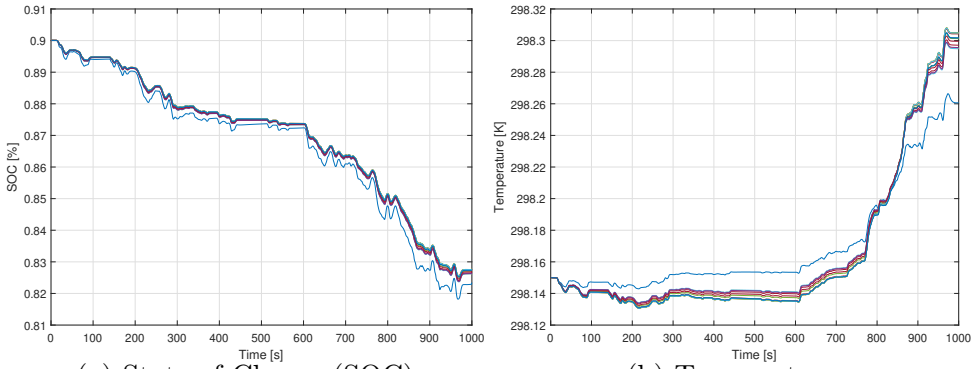
Figure 4.5: Internal states of the single cells of the battery pack at different time instants, $t = 300\ s$ and $t = 700\ s$.

iterations as

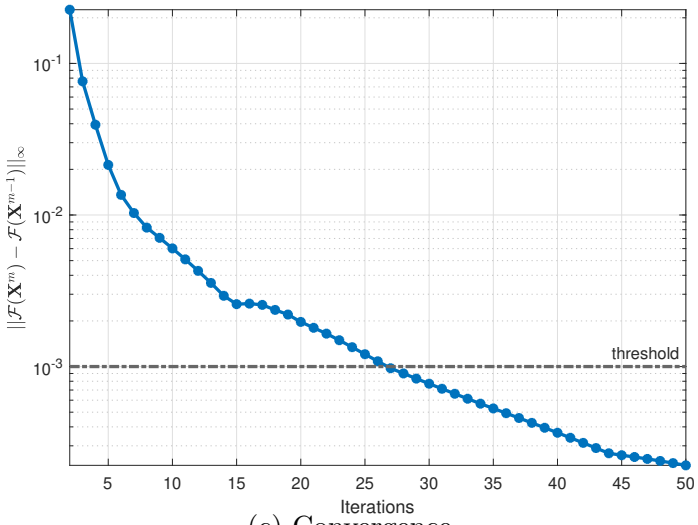
$$||\mathcal{F}(\mathbf{X}^n) - \mathcal{F}(\mathbf{X}^{n-1})||_\infty. \tag{4.45}$$

This quantity can be used to evaluate the level of precision reached by iterations and take decision on whether it is worth to keep refining the results with more iterations or a desirable level of precision has been accomplished.

Another feature of this solver is the capability of simulating the battery pack also in adverse situations, for example in the presence of faults or other type of disruptions. This also proves the robustness of the proposed quasi-Newton solver with respect to the Jacobian approximation. To showcase a plausible scenario, the capacity of a single cell has been altered to be 50% less of the nominal value to simulate a sudden loss of capacity or premature



(a) State of Charge (SOC) (b) Temperature



(c) Convergence

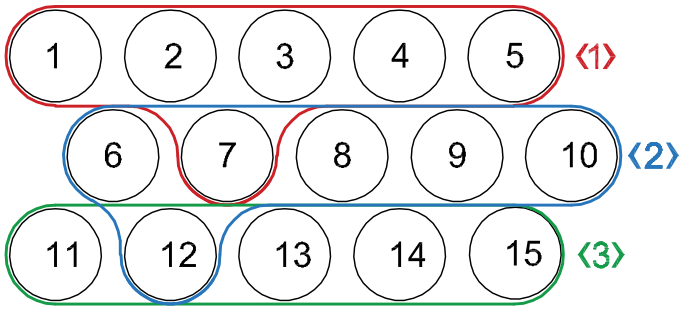
Figure 4.6: Examples of state-space time evolution of a battery pack with a capacity loss on cell 8.

aging. The previous example with $N = 15$ was still considered with the same conditions and the capacity decrease was injected on cell 8. In Figure 4.6 some notable quantities are represented (SOC and temperature) and it is possible to clearly notice the different behaviour of the faulty cell with respect to the other ones.

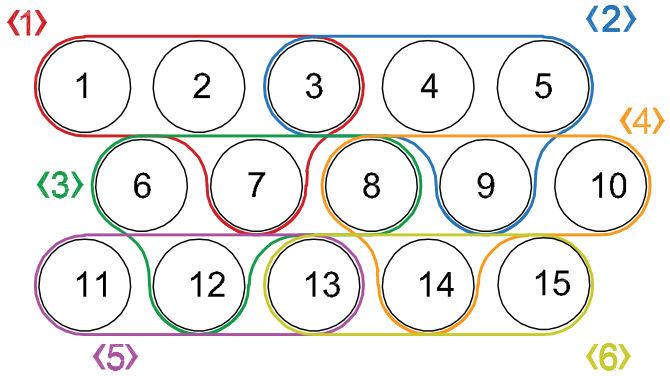
4.3 Waveform Relaxation and Anderson Acceleration

In this Section, the modifications needed to the WR method are detailed. Because a full thermal model is added, new couplings between subsystems are created. This has to be taken into account when exchanging information between subsystems. For the sake of clarity, some important preliminary notation is repeated here.

Given a battery pack composed of N parallel-connected cells, consider the



(a) Example of an overlapping decomposition for a battery pack with $N = 15$ and $a = 3$.



(b) Example of an overlapping decomposition for a battery pack with $N = 15$ and $a = 6$.

Figure 4.7: Examples of different overlapping decompositions for the same battery pack with $N = 15$.

set $\mathcal{I} := \{1, 2, \ldots, N\} \subset \mathbb{N}$, where each index corresponds to a cell. $\mathcal{I}$ gets divided in smaller subsets in such a way that $\cup_{j=1}^{a} \langle j \rangle = \mathcal{I}$, where $a \in \mathbb{N}$ is the number of subsystems in which the battery pack is divided and j is the subsystem index, with $\langle j \rangle \subset \mathcal{I}$, $j = 1, \ldots, a$. In Figure 4.7, an example of the decomposition used is presented. In particular, in Figure 4.7a, a system with $N = 15$ and $a = 3$ is shown; the subdomains in this example are composed of

$$\langle 1 \rangle = \{1, 2, 3, 4, 5, 7\},\ \langle 2 \rangle = \{6, 7, 8, 9, 10, 12\},\ \langle 3 \rangle = \{11, 12, 13, 14, 15\}.$$

Notice that cell 7 is shared between subsystems $\langle 1 \rangle$ and $\langle 2 \rangle$, while cell 12 is shared between $\langle 2 \rangle$ and $\langle 3 \rangle$. In Figure 4.7b instead, the same 15 cells are divided in 6 subsystems, thus resulting in

$$\langle 1 \rangle = \{1, 2, 3, 7\},\ \langle 2 \rangle = \{3, 4, 5, 9\},\ \langle 3 \rangle = \{6, 7, 8, 12\},$$
$$\langle 4 \rangle = \{8, 9, 10, 14\},\ \langle 5 \rangle = \{11, 12, 13\},\ \langle 6 \rangle = \{13, 14, 15\}.$$

In this case more cells are shared between different subsystems, however note how each cell is shared at most by two subsystems. The peculiar shape of the subsystems has been chosen with the objective of minimizing the overlap between subsystems. As shown in Section 3.4, having an overlap greater than 1 between subsystems is not necessary and introduces additional calculations that would slow down the resolution. Previously, a mono-dimensional battery pack was considered, because there was no thermal coupling, the spatial positioning of cells was not really a concern. In this part of the work, instead, having added a full thermal model, relative spatial positioning of cells is very important. The chosen shape allows to have minimal overlap: the battery pack is divided into rows of subsystems, with in addition a cell from the following row. If the row under consideration is odd, the included cell from the following row will be the one between the 2^{nd} and the 3^{rd}; if the row is even, the included cell will be in the row below between the 1^{st} and the 2^{nd} cells of the row under consideration. Also, differently from the previous part of the work, subsystems can have a different number of cells, as the last row obviously does not include any cell from the row below. The number of cells contained in each subsystem (i.e. the cardinality) will be indicated with s. In general, the only given values for a generic battery pack are the number of cells N and its topology: the number of rows will be denoted by e, and the number of cells in each row (that we will consider constant), will be indicated with f. For the examples of Figure 4.7, we have that $e = 3$ and $f = 5$. The obvious constraint on e and f is that it has to hold $e * f = N$.

Consider subsystem $\langle j \rangle$ and denote with $\langle j \rangle, \ell$ the ℓ-th index (or cell) of the j-th subdomain. The iterative method relies on the simulation of these smaller subsystems. One of the key points is that they have the same structure of the original problem and can be solved using the same algorithm (Algorithm 4). Moreover, they have to be independent from each other, so that they can be solved in parallel. In order to obtain this, it has to be ensured that $\mathbf{I}$ and $\mathbf{T}$ for which the problem is solved, are referred only to $\langle j \rangle$. The current equilibrium can be formulated as

$$\sum_{\ell=1}^{s} \mathbf{I}_{\langle j \rangle, \ell} + \mathbf{I}_{rest, \langle j \rangle} - \mathbf{I}_{tot} = 0, \tag{4.46}$$

where

$$\mathbf{I}_{rest, \langle j \rangle} = \sum_{h \in \mathcal{I} \setminus \langle j \rangle} \mathbf{I}_h. \tag{4.47}$$

This term comprises the currents in all the cell of the battery pack with the exception of the ones belonging to subsystem $\langle j \rangle$. The introduction of the term $\mathbf{I}_{rest, \langle j \rangle}(t)$ allows to avoid accounting for the same current multiple times despite the overlap between different subsystems. The same approach has to be applied for the thermal dynamics: consider Equation (4.23), define an expression that divides the temperatures belonging to the subsystem from the

ones belonging to the surrounding elements

$$\frac{d\mathbf{T}_{\langle j\rangle}}{dt} = A_{T,\langle j\rangle}\mathbf{T}_{\langle j\rangle} + A_{rest,\langle j\rangle}\mathbf{T}_{rest,\langle j\rangle} + B_{Q,\langle j\rangle}\mathbf{Q}_{\langle j\rangle} + B_{a,\langle j\rangle}T_f, \tag{4.48}$$

with $\mathbf{T}_{\langle j\rangle}$ containing all the temperatures of subsystem $\langle j\rangle$ plus $\mathbf{T}_{b,\langle j\rangle}$ and $\mathbf{T}_{t,\langle j\rangle}$. Similarly to the definition given before, the matrices are defined as

$$A_T = \begin{bmatrix} -\frac{Q_{cc}}{C_c}\mathcal{L}_{\langle j\rangle} - \frac{q_{ca}}{C_c}\Phi_{a,\langle j\rangle} - \frac{2q_{cp}}{C_c}\mathbb{I}_s & \frac{q_{cp}}{C_c}\mathbf{1}_s & \frac{q_{cp}}{C_c}\mathbf{1}_s \\ \frac{q_{cp}}{C_p}\mathbf{1}_s^\top & \frac{-sq_{cp}-q_{ta}}{C_p} & 0 \\ \frac{q_{cp}}{C_p}\mathbf{1}_s^\top & 0 & \frac{-sq_{cp}-q_{ba}}{C_p} \end{bmatrix}, \tag{4.49}$$

$$B_Q = \begin{bmatrix} \frac{1}{C_c}\mathbb{I}_s & 0 & 0 \end{bmatrix}^\top, \tag{4.50}$$

$$B_a = \begin{bmatrix} \frac{q_{ca}}{C_c}\Phi_{a,\langle j\rangle}\mathbf{1}_s & \frac{q_{ta}}{C_p} & \frac{q_{ba}}{C_p} \end{bmatrix}^\top. \tag{4.51}$$

Note that the formulation of the spot ambient matrix Φ_a is different from the case presented above and defined as

$$\Phi_{a,\langle j\rangle} = \Phi_a - R_{\langle j\rangle}, \tag{4.52}$$

where the matrix $R_{\langle j\rangle}$ has on its diagonal position the number of cells that generate heat but are in a different subsystem. Also the Laplacian $\mathcal{L}_{\langle j\rangle}$ has to be slightly modified

$$\mathcal{L}_{\langle j\rangle} = \mathcal{D}_{\langle j\rangle} - \mathcal{A}_{\langle j\rangle}, \tag{4.53}$$

where $\mathcal{A}_{\langle j\rangle}$ is defined as before, but only the cells of subdomain $\langle j\rangle$ are considered, while $\mathcal{D}_{\langle j\rangle}$ is the degree matrix where only the cells of the present subsystem are considered. Moreover, $\mathbf{T}_{rest,\langle j\rangle}$ is defined as

$$\mathbf{T}_{rest,\langle j\rangle} = \{\mathbf{T}_h : h \in \mathcal{I} \setminus \langle j\rangle\}, \tag{4.54}$$

while $A_{rest,\langle j\rangle}$ is given by

$$A_{rest,\langle j\rangle} = \frac{Q_{cc}}{C_c}\tilde{\mathcal{A}}_{\langle j\rangle}. \tag{4.55}$$

The adjacency matrix $\tilde{\mathcal{A}}_{\langle j\rangle}$ is of dimension $(s \times N)$, however, differently from $\mathcal{A}$, it is not symmetrical any more. $\tilde{\mathcal{A}}(a,b) = 1$ only if cell a of subsystem $\langle j\rangle$ is adjacent to cell b of any other subsystem.

In order to clarify these definitions, consider the following numerical example. Consider the division in subsystems depicted in Figure 4.7b and, in particular, focus on subsystem $\langle 1\rangle$. In Figure 4.8 a close-up of $\langle 1\rangle$ is provided, where all the relevant neighbouring elements are represented. The subdomain under consideration is composed of cells $\{1, 2, 3, 7\}$; the surrounding adjacent elements are in part cells that belong to other subsystems and produce heat

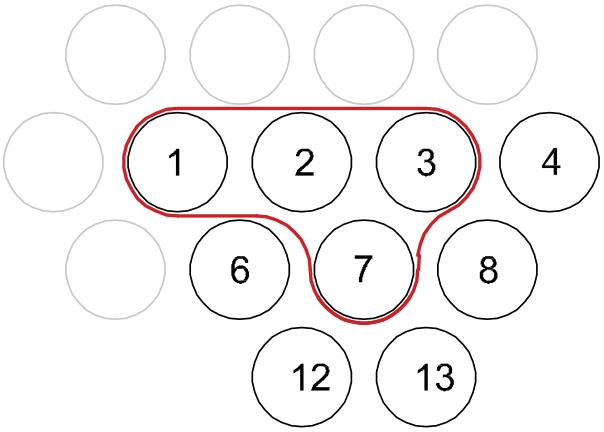


Figure 4.8: Example of a subsystem and its neighbouring cells.

and in part elements at ambient temperature (represented in gray). Start by defining the degree and the adjacency matrix

$$\mathcal{D}_{\langle 1\rangle} = \begin{matrix} 1 & 2 & 3 & 7 & h \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 7 \end{matrix}, \quad \mathcal{A}_{\langle 1\rangle} = \begin{matrix} 1 & 2 & 3 & 7 & h \end{matrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 7 \end{matrix}, \tag{4.56}$$

where the indexes h outside the matrix indicate the cell number. From the definition of $\mathcal{D}_{\langle 1\rangle}$ and $\mathcal{A}_{\langle 1\rangle}$, it is possible to calculate $\mathcal{L}_{\langle 1\rangle}$ as the difference of the two

$$\mathcal{L}_{\langle 1\rangle} = \begin{matrix} 1 & 2 & 3 & 7 & h \end{matrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 7 \end{matrix}. \tag{4.57}$$

Next, define the ambient spot matrix Φ_a and the complementary spot matrix R as

$$\Phi_a = \begin{matrix} 1 & 2 & 3 & 7 & h \end{matrix} \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 7 \end{matrix}, \quad R_{\langle 1\rangle} = \begin{matrix} 1 & 2 & 3 & 7 & h \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 7 \end{matrix}, \tag{4.58}$$

resulting in a corrected ambient spot matrix $\Phi_{a,\langle 1\rangle}$ equal to

$$\Phi_{a,\langle 1\rangle} = \begin{matrix} \begin{matrix} 1 & 2 & 3 & 7 \end{matrix} & h \\ \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 7 \end{matrix} \end{matrix} . \tag{4.59}$$

The last element to be defined is the adjacency matrix between subsystem $\langle 1\rangle$ and the other cell in different subsystems.

$$\tilde{\mathcal{A}}_{\langle j\rangle} = \begin{matrix} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \end{matrix} & h \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 7 \end{matrix} \end{matrix} . \tag{4.60}$$

At this point, Equation 4.34 needs to be rewritten in a *domain decomposition* formulation. For the single subsystem it holds

$$\mathcal{F}_{\langle j\rangle}(\mathbf{X}_{\langle j\rangle}) = \mathcal{F}_{\langle j\rangle}\left(\begin{bmatrix}\mathbf{I}_{\langle j\rangle} & \mathbf{T}_{\langle j\rangle}\end{bmatrix}\right) = \begin{bmatrix} \sum_{\ell=1}^{s} \mathbf{I}_{\langle j\rangle,\ell} + \mathbf{I}_{rest,\langle j\rangle} - \mathbf{I}_{tot} \\ \mathbf{V}_{\langle j\rangle,1} - \mathbf{V}_{\langle j\rangle,2} \\ \mathbf{V}_{\langle j\rangle,2} - \mathbf{V}_{\langle j\rangle,3} \\ \vdots \\ \mathbf{V}_{\langle j\rangle,N-1} - \mathbf{V}_{\langle j\rangle,N} \\ \mathbf{T}_{\langle j\rangle,1} - \hat{\mathbf{T}}_{\langle j\rangle,1} \\ \mathbf{T}_{\langle j\rangle,2} - \hat{\mathbf{T}}_{\langle j\rangle,2} \\ \vdots \\ \mathbf{T}_{\langle j\rangle,N} - \hat{\mathbf{T}}_{\langle j\rangle,N} \\ \mathbf{T}_{\langle j\rangle,b} - \hat{\mathbf{T}}_{\langle j\rangle,b} \\ \mathbf{T}_{\langle j\rangle,t} - \hat{\mathbf{T}}_{\langle j\rangle,t} \end{bmatrix} = 0, \tag{4.61}$$

It was mentioned previously that the method is iterative and this allows to have an estimation for the values $\mathbf{I}_{rest,\langle j\rangle}$ and $\mathbf{T}_{rest,\langle j\rangle}$ if these are taken from previous iterations.

4.3.1 Waveform Relaxation algorithm

Consider an iteration k, for which a value of $\mathbf{I}^{k-1}$ and $\mathbf{T}^{k-1}$ is provided; if $k = 1$, i.e. the first iteration, then an appropriate initial guess is given. The

single WR iteration is computed by solving in parallel

$$\begin{cases} \mathcal{F}^k_{\langle j \rangle} \left(\left[\tilde{\mathbf{I}}^k_{\langle j \rangle} \ \tilde{\mathbf{T}}^k_{\langle j \rangle} \right], \left[\mathbf{I}^{k-1} \ \mathbf{T}^{k-1} \right] \right) = 0, \\ \tilde{\mathbf{I}}^k_{\langle j \rangle} = D_{\langle j \rangle} \mathbf{I}^k_{\langle j \rangle} + (\mathbb{I}_{\langle j \rangle} - D_{\langle j \rangle}) \mathbf{I}^{k-1}_{\langle j \rangle}, \\ \tilde{\mathbf{T}}^k_{\langle j \rangle} = D_{\langle j \rangle} \mathbf{T}^k_{\langle j \rangle} + (\mathbb{I}_{\langle j \rangle} - D_{\langle j \rangle}) \mathbf{T}^{k-1}_{\langle j \rangle}, \end{cases} \tag{4.62}$$

for $k = 1, 2, \ldots, k_{max}$, where $\mathbb{I}_{\langle j \rangle}$ is an identity matrix of dimension $s \times s$ and $D_{\langle j \rangle}$ is a diagonal matrix having entries

$$D_{\langle j \rangle}(h, h) := \begin{cases} 0.5 & \text{if } (h \in \langle j \rangle) \wedge (h \in \langle g \rangle), \ (\forall g = 1, 2, \ldots a) \wedge (g \neq j), \\ 1 & \text{otherwise.} \end{cases} \tag{4.63}$$

By introducing this weighing coefficient, a stronger coupling between subsystems is created; if a cell is shared between multiple subsystems (the assumption is that each cell is contained in at most two subsystems), then its temperature and current values are averaged between the one obtained in the previous iteration and the one for which the subsystem is being solved. The choice and the importance of this relaxation parameter were discussed in the previous Section. In this particular case, a cell can be contained in two subsystems at most, hence the factor 0.5. For more details regarding this topic, please refer to [Gander, 1999, Gander and Ruehli, 2010, Gander and Ruehli, 2004, Gander, 2008]. Notice also how the function $\mathcal{F}^k_{\langle j \rangle}$ uses as input the vectors $\left[\mathbf{I}^{k-1} \ \mathbf{T}^{k-1} \right]$ coming from the previous iteration (differently from Equation 4.61), to compute $\mathbf{I}_{rest,\langle j \rangle}$ and $\mathbf{T}_{rest,\langle j \rangle}$. Moreover, it is important to point out that system 4.62 has the same formulation of 4.34 and can be therefore solved with Algorithm 4. Note that $\hat{\mathbf{T}}$ is not present in (4.62) because they are calculated in each subsystem and do not need to be exchanged between different subsystems, as they are only used to correct the value of $\mathbf{T}$.

At this point, it is possible to present the full WR algorithm. Notice how the correction step previously presented in 3.2 was not modified, therefore the details are not reported here. The reason why it is not necessary to include the temperature in the correction step is that the couplings created by the addition of temperatures are not strong as the ones created by the electrical parallel connection. The resolution of the subsystems is performed in Line 3 and can be done in parallel, exploiting the parallel capabilities of modern processors and the fact that subsystems are independent. The iterations are stopped either when a certain tolerance ϵ_{wr} is reached (calculated as the absolute value of the difference between $\mathcal{F}^k$ and $\mathcal{F}^{k-1}$) or a maximum number of iterations k_{max} is reached. Notice how also Algorithm 5 can be considered a fixed point iteration; in fact, consider the unknown vector $\mathbf{X} = [\mathbf{I} \ \mathbf{T}]$, then it is possible to see that the algorithm takes as input a value of $\mathbf{X}$ coming from previous iterations and outputs a better estimation of it. This property can be exploited in order to speed up the convergence of the method using AA. Take a single iteration of

Algorithm 5 (WR Jacobi-type algorithm)

Require: $\mathbf{I}_{tot}$, an initial guess $\mathbf{I}^0$ and $\mathbf{T}^0$, a tolerance ϵ_{wr} and a maximum number of iterations k_{max}.

Set $k \leftarrow 1$, $\mathbf{I}^1 \leftarrow$ inf and $\mathbf{T}^1 \leftarrow$ inf.
while $||\mathcal{F}^k - \mathcal{F}^{k-1}|| > \epsilon$ and $k < w_{max}$ **do**
  for all $j \in \{1, \dots, a\}$ **do**
    Compute $I^{k-1}_{rest,\langle j \rangle}$.
    Solve (4.62) to get $\mathbf{I}^k_{\langle j \rangle}$ and $\mathbf{T}^k_{\langle j \rangle}$.
  end for
  Compute $\mathbf{I}^{*,k}_{\langle j \rangle,\ell}$ for $j = 1, \dots, a$ and $\ell = 1, \dots, s$ using (3.25) and (3.26).
  Set $\mathbf{I}^k_{\langle j \rangle,\ell} \leftarrow \mathbf{I}^{*,k}_{\langle j \rangle,\ell}$ for $j = 1, \dots, a$ and $\ell = 1, \dots, s$.
  Set $k \leftarrow k + 1$.
end while

Algorithm 5, and define the operator S as

$$\mathbf{X}^k = S\left(\mathbf{X}^{k-1}\right). \tag{4.64}$$

Algorithm 6 contains the formulation of AA adapted for this particular case of a battery pack with temperature dynamics. This algorithm is an adaptation of

Algorithm 6 Anderson Acceleration algorithm

Require: Fixed-point operator S, initial guess $\mathbf{X}^0$, m, k_{max}, ϵ_{aa}.

Set $k \leftarrow 1$, $\mathbf{X}^1 = S(\mathbf{X}^0)$, $\gamma^0 = \mathbf{X}^1 - \mathbf{X}^0$ and $s_0 \leftarrow \mathbf{X}^1$.
while $k < k_{max}$ and $||\gamma^k|| > \epsilon_{aa}$ **do**
  Set $m_k \leftarrow \min(m, k)$.
  Compute $s_k = S(\mathbf{X}^k)$.
  Set $\gamma^k \leftarrow s_k - \mathbf{X}^k$.
  Set $\Delta\gamma^{k-1} \leftarrow \gamma^k - \gamma^{k-1}$ and $\tilde{\mathbf{\Gamma}}_k \leftarrow [\Delta\gamma^{k-m_k}, \dots, \Delta\gamma^{k-1}]$.
  Solve the LS problem $\eta^k = [\eta^k_0, \dots, \eta^k_{m_k-1}]^\top = \text{argmin}_\eta ||\gamma^k - \tilde{\mathbf{\Gamma}}_k \eta||_2$.
  Set $\Delta S_{k-1} \leftarrow s_k - s_{k-1}$ and $\mathcal{S}_k \leftarrow [\Delta S_{k-m_k}, \dots, \Delta S_{k-1}]$.
  Set $\mathbf{X}^{k+1} \leftarrow s_k - \mathcal{S}_k \eta^k$.
  Set $k \leftarrow k + 1$.
end while

the one presented in [Walker, 2011], where the Euclidean norm is used in the optimization problem, in order to have a linear system as first-order optimality condition [Giorgi and Guerraggio, 1994]. Moreover, the optimization problem presented in [Walker, 2011] is formulated as an unconstrained problem; to do so we define γ^k as

$$\gamma^k = S(\mathbf{X}^k) - \mathbf{X}^k, \tag{4.65}$$

i.e. the difference between the vector $\mathbf{X}^k$ and the application of the operator S to the vector itself. This residual is then inserted in a matrix $\tilde{\mathbf{\Gamma}}^k$, where all

the differences of the residuals are stored. Successively proceed by solving a least-square problem

$$\min_{\eta=[\eta_0^k,\dots,\eta_{m_k-1}^k]^\top} \|\gamma^k - \tilde{\mathbf{\Gamma}}^k \eta\|_2. \tag{4.66}$$

Once the weighing coefficients η have been found, it is possible to proceed with an update step

$$\begin{aligned} \mathbf{X}^{k+1} &= S(\mathbf{X}^k) - \sum_{b=0}^{m_k-1} \eta_b^k [S(\mathbf{X}^{k-m_k+b-1}) - S(\mathbf{X}^{k-m_k+b})] \\ &= S(\mathbf{X}^k) - \mathcal{S}^k \eta^k. \end{aligned} \tag{4.67}$$

Notice how m (and consequentially m_k) can be used to determine the depth of iterations we want to look back at. As it happened for the WR algorithm, a maximum number of iterations k_{max} and a tolerance ϵ_{aa} are used to respectively stop iterations if a maximum is reached or if a certain tolerance is obtained. More details about these methods, convergence proofs and the full explanation are not reported here and can be found in [Walker, 2011, Walker and Ni, 2011, Ciaramella and Fabrini, 2021].

4.4 Simulation results and discussion

In this Section, some significant results obtained with the proposed method are highlighted. All the simulations were carried out on a PC with an Intel i9-10920X CPU (12 cores) and 32GB or RAM. The maximum number of available cores in the processor is an important parameter to take into consideration. Because the subsystems are solved in parallel, having more subsystems solved simultaneously is undoubtedly beneficial to speed up computations. The first step is to set a reference time for *centralized* simulations as N increases: the topology of the battery pack was chosen with $e = 6$, with the value of f calculated as $f = N/e$. For a chosen value of N, the system is simulated and the time is recorded. Because a processor is sharing resources with other background tasks running on the computer, for the same N, the simulation was repeated multiple times in order to ensure fairness. Moving over to the WR method, accelerated with AA, some clarifications are in order regarding the topology of the battery pack. As was shown in Section 3.2, having the number of subsystems equal to the number of processors yields the best results. Denoting with n_{proc} the number of cores in the processor, set $e = n_{proc}$ and each row of the battery pack coincides with a subsystem. f is once again calculated once N and e are defined. As was done for the *centralized* method, each simulation for a fixed N and n_{proc}, was repeated several times. In Figure 4.9 the proposed method is compared with the *centralized* approach. The continuous blue line represents this latter one and the dotted lines represent scaling for different values of n_{proc}. Scaling of the *centralized* method is polynomial, while

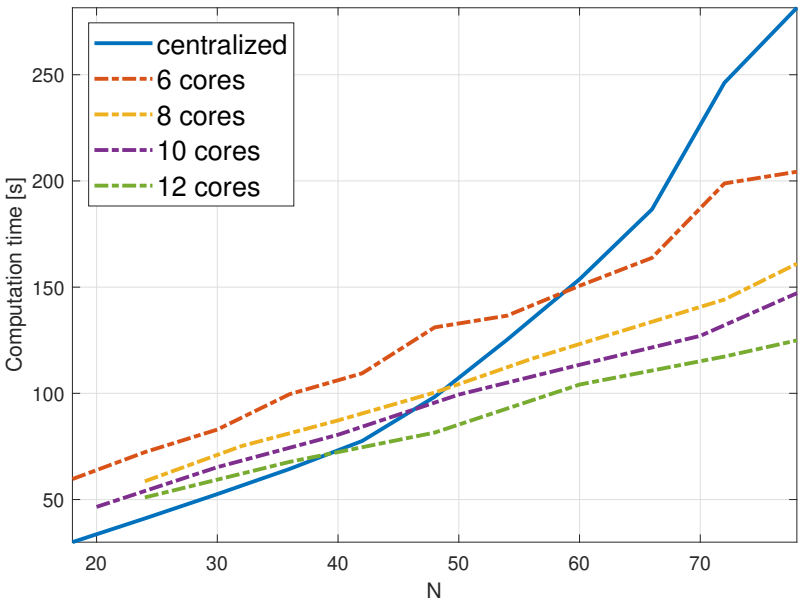


Figure 4.9: Comparison between computation times with the *centralized* method and the proposed WR/AA approach.

all the others have an almost linear behaviour. This means that, depending on the value of n_{proc}, the WR/AA method will become more time efficient at a certain point. In particular, for $n_{proc} = 6$ this happens at $N = 58$, for $n_{proc} = 8$ at $N = 48$, for $n_{proc} = 10$ at $N = 45$ and finally for $n_{proc} = 12$ at $N = 38$. These are optimal results that show the effectiveness of the proposed WR/AA method, that could be additionally improved by using more parallel cores.

4.5 Conclusions

In this Chapter it was shown how Waveform Relaxation and Anderson Acceleration can be used to speed up the computation time of large parallel-connected battery packs. With respect to the previous Chapter, where the method was presented, a full thermal model was implemented for the battery pack. This step required modifications to the method and in particular, the way subsystems are chosen and quantities are exchanged between them. In addition, in order to make the method as general as possible, the use of a Quasi-Newton solver is proposed to find the roots of a discretized DAE system representing the battery pack. This allows for any cell model to be used.

Further developments on this topic may include the extension to a battery pack configuration including both parallel and series-connected cells and consequently the generalization of the code to a battery pack with a general topology and connections. From there, it would be possible to develop a toolbox for fast simulation of large battery packs.

Part III

Set-based State Estimation and Fault Detection

Chapter 5

Set-based state estimation

The estimation of states in a dynamical system has been historically a challenging problem, tackled with a multitude of different approaches. When dealing with state estimation, the main obstacles to overcome are the limited number of measurements available and the inevitable inaccuracies of the model used. For the particular case under consideration, the measurements that can be captured from a battery pack or a battery cell are limited to current, voltage and temperature. None of the internal states of the battery, independently from the model used, can be measured directly. In fact, some very important states, such as SOC, can only be estimated using the available measurements.

Among the many different approaches in literature, the use of set-based methods is suggested in this Chapter. Different set representations have been proposed in literature to address the problem of set-based estimation: polytopes [Shamma and Tu, 1997], intervals [Jaulin et al., 2001], zonotopes [Alamo et al., 2008] and constrained zonotopes [Scott et al., 2016]. These set representations differ in accuracy and computational effort. In the context of Li-ion batteries, interval based approaches have been used for state of charge (SOC) estimation of a single cell [Rausch et al., 2014], in [Zhang et al., 2020] and in [Locatelli et al., 2021], for all the states of the cell. The work of [Zhang et al., 2021] uses an interval observer to estimate states of a battery pack, including thermal dynamics. An interval PDE observer is implemented in [Perez and Moura, 2015], for the state estimation of an electrochemical model.

The set-based approach proposed here is based on constrained zonotopes. They have the advantage of combining the efficiency and scalability of zonotopes with the flexibility of convex polytopes. Therefore, they represent an efficient methodology which is able to compute significantly tighter enclosures than e.g. intervals, at modest additional cost [Scott et al., 2016]. To further validate the results, a comparison is proposed between constrained zonotopes and interval sets; these latter are computed with a forward-backward algorithm according to [Jaulin et al., 2001]. Before detailing the proposed set-based method, some preliminaries on set operations are given.

5.1 Set operations

Starting from interval analysis, consider $\underline{x}, \overline{x} \in \mathbb{R}$, such that $\underline{x} \leq \overline{x}$. An interval $[x] \subset \mathbb{R}$ is a nonempty set of real numbers defined as

$$[x] \triangleq \{x \in \mathbb{R} : \underline{x} \leq x \leq \overline{x}\}. \tag{5.1}$$

Moreover, the midpoint of an interval $[x]$ is defined as

$$\text{mid}([x]) \triangleq (1/2)(\overline{x} + \underline{x}), \tag{5.2}$$

while the radius as

$$\text{rad}([x]) \triangleq (1/2)(\overline{x} - \underline{x}). \tag{5.3}$$

The set of all intervals over $\mathbb{R}$ is denoted by $\mathbb{IR}$. The set of all interval vectors in $\mathbb{R}^n$ is denoted by $\mathbb{IR}^n$. A box $[X] \in \mathbb{IR}^n$ is defined as

$$[X] = ([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n]). \tag{5.4}$$

The midpoint and radius of $[X]$ are defined respectively by

$$\text{mid}([X]) \triangleq (\text{mid}([\underline{x}_1, \overline{x}_1]), \ldots, \text{mid}([\underline{x}_n, \overline{x}_n]) \tag{5.5}$$

$$\text{rad}([X]) \triangleq (\text{rad}([\underline{x}_1, \overline{x}_1]), \ldots, \text{rad}([\underline{x}_n, \overline{x}_n]). \tag{5.6}$$

A real arithmetic operation $\odot$ is extended to intervals $[x_1], [x_2] \in \mathbb{IR}$ and defined as

$$[x_1] \odot [x_2] \triangleq \{x_1 \odot x_2 : x_1 \in [x_1], x_2 \in [x_2]\}. \tag{5.7}$$

The intersection of two intervals is defined as

$$[x_1] \cap [x_2] \triangleq [\max\{\underline{x}_1, \underline{x}_2\}, \min\{\overline{x}_1, \overline{x}_2\}]. \tag{5.8}$$

More details on interval mathematics, such as inclusion functions and basic operations, can be found in [Moore et al., 2009].

Moving toward more complex sets a *convex polytope* is defined according to Definition 1.

Definition 1 *P is a* convex polytopes *if it is bounded and*

$$\exists(\mathbf{H}, \mathbf{k}) \in \mathbb{R}^{n_h \times n} : P = \{\boldsymbol{z} \in \mathbb{R}^n : \mathbf{H}\boldsymbol{z} \leq \boldsymbol{k}\} \tag{5.9}$$

holds.

The halfspace-representation (H-rep) of P has been used here. However it is possible to represent a convex polytope as the convex hull of its vertices (V-rep).

Some basic operations that characterize the set-based applications are now

defined. Let Z, $W \subset \mathbb{R}^n$, $Y \subset \mathbb{R}^m$, and $\mathbf{R} \in \mathbb{R}^{m \times n}$, then

$$\mathbf{R}Z \triangleq \{\mathbf{R}\boldsymbol{z} : \boldsymbol{z} \in Z\}, \tag{5.10}$$

$$Z \oplus W \triangleq \{\boldsymbol{z} + \boldsymbol{w} : \boldsymbol{z} \in Z,\, \boldsymbol{w} \in W\}, \tag{5.11}$$

$$Z \cap_{\mathbf{R}} Y \triangleq \{\boldsymbol{z} \in Z : \mathbf{R}\boldsymbol{z} \in Y\}, \tag{5.12}$$

where (5.10) denotes the linear image of Z, (5.11) the Minkowski sum of sets, while (5.12) is a generalized intersection [Scott et al., 2016]. Using intervals the Minkowski sum can be computed exactly, but (5.10) and (5.12) are conservative due to the wrapping effect[1]. In contrast convex polytopes are closed under all the three operations, but they can be very costly and numerically in high dimensions

Remark 9 (5.10) *and* (5.11) *are computed efficiently in vertex representation (V-rep), while* (5.12) *is computed efficiently in half-space representation (H-rep). Despite this the conversion between representation is computationally expensive.*

Constrained zonotopes are an extension of zonotopes [Scott et al., 2016], which allow (5.11)-(5.12) to be computed exactly with a low computational burden.

Definition 2 *A set $Z \subset \mathbb{R}^n$ is a* constrained zonotope *if there exists*

$$(\mathbf{G}, \boldsymbol{c}, \mathbf{A}, \boldsymbol{b}) \in \mathbb{R}^{n \times n_g} \times \mathbb{R}^n \times \mathbb{R}^{n_c \times n_g} \times \mathbb{R}^{n_c} \tag{5.13}$$

such that

$$Z = \{\boldsymbol{c} + \mathbf{G}\boldsymbol{\xi} : \|\boldsymbol{\xi}\|_\infty \leq 1, \mathbf{A}\boldsymbol{\xi} = \boldsymbol{b}\}\,. \tag{5.14}$$

Equation (5.14) represents the so called *constrained generator representation* (CG-rep). $\boldsymbol{G}_z$ represents the *generator matrix* whose columns correspond to the *generators*, $\boldsymbol{c}_z$ is the *center*, and $\mathbf{A}_z\boldsymbol{\xi} = \boldsymbol{b}_z$ are the *constraints*. For compactness reasons we use the shortened notation $Z = \{\mathbf{G}_z, \boldsymbol{c}_z, \mathbf{A}_z, \boldsymbol{b}_z\}$, and $Z = \{\mathbf{G}, \boldsymbol{c}\}$ for zonotopes. Moreover, the operations (5.15)-(5.17) can be easily computed in CG-rep as

$$\mathbf{R}Z = \{\mathbf{R}\mathbf{G}_z, \mathbf{R}\boldsymbol{c}_z, \mathbf{A}_z, \boldsymbol{b}_z\}\,, \tag{5.15}$$

$$Z \oplus W = \left\{ \begin{bmatrix}\mathbf{G}_z & \mathbf{G}_w\end{bmatrix}, \boldsymbol{c}_z + \boldsymbol{c}_w, \begin{bmatrix}\mathbf{A}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_w\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_z \\ \boldsymbol{b}_w\end{bmatrix} \right\}, \tag{5.16}$$

$$Z \cap_{\mathbf{R}} Y = \left\{ \begin{bmatrix}\mathbf{G}_z & \mathbf{0}\end{bmatrix}, \boldsymbol{c}_z, \begin{bmatrix}\mathbf{A}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_y \\ \mathbf{R}\mathbf{G}_z & -\mathbf{G}_y\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_z \\ \boldsymbol{b}_y \\ \boldsymbol{c}_y - \mathbf{R}\boldsymbol{c}_z\end{bmatrix} \right\}. \tag{5.17}$$

[1]Domain representation growth due to over-estimation at each sample time

Consider a class of nonlinear discrete-time systems described by the following equations

$$\begin{aligned} \boldsymbol{x}_{k_t} &= \boldsymbol{f}(\boldsymbol{x}_{k_t-1}, \boldsymbol{u}_{k_t-1}, \boldsymbol{w}_{k_t-1}) \quad \text{for} \quad k_t \geq 1, \\ \boldsymbol{y}_{k_t} &= \boldsymbol{g}(\boldsymbol{x}_{k_t}, \boldsymbol{u}_{k_t}, \boldsymbol{v}_{k_t}) \qquad\quad \text{for} \quad k \geq 0, \end{aligned} \tag{5.18}$$

where $\boldsymbol{x}_{k_t} \in \mathbb{R}^{n_x}$ is the system state vector, $\boldsymbol{u}_{k_t} \in \mathbb{R}^{n_u}$ are the known inputs, $\boldsymbol{y}_{k_t} \in \mathbb{R}^{n_y}$ indicates the measured outputs, while $\boldsymbol{w}_{k_t} \in \mathbb{R}^{n_w}$ and $\boldsymbol{v}_{\boldsymbol{k_t}} \in \mathbb{R}^{n_v}$ denote process and output uncertainties respectively. Initial condition $\boldsymbol{x}_0 \in X_0$ and the disturbances are bounded by $\boldsymbol{w}_{k_t} \in W$ and $\boldsymbol{v}_{k_t} \in V$, where X_0, W and V are known compact sets.

The objective of set-based state estimation is to compute, at each $k_t \geq 0$, tight enclosures $\hat{X}_{k_t}$ of the set of states consistent with the model dynamics, the bounded uncertainties X_0, W, V and the collected measurements $\boldsymbol{y}_{k_t}$. This is done through a prediction-update algorithm, that computes the compact sets $\bar{X}_{k_t}$ and $\hat{X}_{k_t}$ as follows

$$\bar{X}_{k_t} \supseteq \{\mathbf{f}(\mathbf{x}, \mathbf{u}_{k_t-1}, \mathbf{w}_{k_t-1}) : \mathbf{x} \in \hat{X}_{k_t-1},\, \boldsymbol{w}_{k_t-1} \in W\}, \tag{5.19}$$

$$\hat{X}_{k_t} \supseteq \{\boldsymbol{x} \in \bar{X}_{k_t} : \mathbf{g}(\mathbf{x}, \mathbf{u}_{k_t}, \mathbf{v}_{k_t}) = \mathbf{y}_{k_t},\, \mathbf{v}_{k_t} \in V\}, \tag{5.20}$$

in which (5.19) and (5.20) are referred to as *prediction step* and *update step* respectively. Note that, in general, the set operations defined in (5.19)-(5.20) cannot be evaluated exactly. This is why tight guaranteed enclosures (meaning, in this context, tight outer approximations) are computed instead. Moreover, order reduction techniques, which existed for zonotopes and provide outer approximations, have been developed also for constrained zonotopes thus allowing to obtain a good trade-off between complexity and accuracy. The complexity order of zonotopes and constrained zonotopes is defined respectively as

$$n_o^{[CZ]} = \frac{n_g - n_c}{n}, \quad n_o^{[Z]} = \frac{n_g}{n}. \tag{5.21}$$

At this point, the tools needed to compute the prediction-update algorithm (5.19) and (5.20) are defined for nonlinear discrete-time systems using constrained zonotopes [Rego et al., 2021]. Theorem 1 defines a constrained zonotopic enclosure of the product between an interval matrix and a constrained zonotope.

Theorem 1 [Rego et al., 2020] Let $X = \{\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\} \subset \mathbb{R}^m$ be a constrained zonotope with n_g generators and n_c constraints, let $\mathbf{J} \in \mathbb{IR}^{n \times m}$ be an interval matrix, and consider the set $S = \mathbf{J}X \triangleq \{\hat{\mathbf{J}}\mathbf{x} : \hat{\mathbf{J}} \in \mathbf{J}, \mathbf{x} \in X\} \subset \mathbb{R}^n$. Let $\bar{\mathbf{G}} \in \mathbb{R}^{n \times \bar{n}_g}$ and $\bar{\mathbf{c}} \in \mathbb{R}^n$ satisfy $X \subseteq \{\bar{\mathbf{G}}, \bar{\mathbf{c}}\}$, and let $\mathbf{m}$ be an interval vector such that $\mathbf{m} \supseteq (\mathbf{J} - \text{mid}(\mathbf{J}))\bar{\mathbf{c}}$ and $\text{mid}(\mathbf{m}) = \mathbf{0}$. Finally, let $\mathbf{P} \in \mathbb{R}^{n \times n}$ be a diagonal matrix defined by $P_{ii} = \text{rad}(m_i) + \sum_{j=1}^{\bar{n}_g} \sum_{k=1}^{m} \text{rad}(J_{ik})|\bar{G}_{kj}|$ for all

$i = 1, 2, \ldots, n$. Then, S is contained in the *inclusion*

$$S \subseteq \triangleleft(\mathbf{J}, X) \triangleq \mathbf{J}X \oplus \mathbf{P}B_\infty^n.$$

Before defining the prediction-update algorithm a preliminary lemma is needed.

Lemma 1 Let $\boldsymbol{\alpha} : \mathbb{R}^n \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_\alpha}$ be of class $\mathcal{C}^1$ and let $\nabla_x \boldsymbol{\alpha}$ denote the gradient of $\boldsymbol{\alpha}$ with respect to its first argument. Let $X \subset \mathbb{R}^n$ and $W \subset \mathbb{R}^{n_w}$ be constrained zonotopes, and let $\mathbf{J} \in \mathbb{IR}^{n \times n}$ be an interval matrix satisfying

$$\nabla_x^T \boldsymbol{\alpha}(\square X, W) \triangleq \{\nabla_x^T \boldsymbol{\alpha}(\mathbf{x}, \mathbf{w}) : \mathbf{x} \in \square X,\ \mathbf{w} \in W\} \subseteq \mathbf{J}. \tag{5.22}$$

For every $\mathbf{x} \in X$, $\mathbf{w} \in W$, and $\boldsymbol{\gamma}_x \in \square X$, there exists $\hat{\mathbf{J}} \in \mathbf{J}$ such that $\boldsymbol{\alpha}(\mathbf{x}, \mathbf{w}) = \boldsymbol{\alpha}(\boldsymbol{\gamma}_x, \mathbf{w}) + \hat{\mathbf{J}}(\mathbf{x} - \boldsymbol{\gamma}_x)$.

Lemma 1 provides the mean value extension for propagating implicitly constrained zonotopes through nonlinear mappings. For the full proof, refer to [Rego et al., 2021]. Note that with $\nabla_x^T \boldsymbol{\alpha}(\square X, W)$ the true image set is denoted and not interval extensions or other outer enclosures. Finally, the next two proposition explain how to perform the mean value prediction and update steps respectively.

Proposition 1 Let $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \to \mathbb{R}^n$ be of class $\mathcal{C}^1$ and let $\nabla_x \mathbf{f}$ denote the gradient of $\mathbf{f}$ with respect to its first argument. Let $\mathbf{u} \in \mathbb{R}^{n_u}$, and let $X \subset \mathbb{R}^n$ and $W \subset \mathbb{R}^{n_w}$ be a constrained zonotope. Choose any $\boldsymbol{\gamma}_x \in \square X$. If Z_w is a constrained zonotope such that $\mathbf{f}(\boldsymbol{\gamma}_x, \mathbf{u}, W) \subseteq Z_w$ and $\mathbf{J} \in \mathbb{IR}^{n \times n}$ is an interval matrix satisfying $\nabla_x^T \mathbf{f}(\square X, \mathbf{u}, W) \subseteq \mathbf{J}$, then $\mathbf{f}(X, \mathbf{u}, W) \subseteq Z_w \oplus \triangleleft(\mathbf{J}, X - \boldsymbol{\gamma}_x)$.

Proposition 2 Let $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_y}$ be of class $\mathcal{C}^1$, let $\mathbf{u} \in \mathbb{R}^{n_u}$, let $X \subset \mathbb{R}^n$ and $V \subset \mathbb{R}^{n_v}$ be a constrained zonotope, and choose any $\mathbf{y} \in \mathbb{R}^{n_y}$ such that $\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{v})$ for some $(\mathbf{x}, \mathbf{v}) \in X \times V$. Choose any $\boldsymbol{\gamma}_x \in \square X$ and any $\tilde{\mathbf{J}} \in \mathbb{R}^{n_y \times n}$. If Z_v is a constrained zonotope such that $-\mathbf{g}(\boldsymbol{\gamma}_x, \mathbf{u}, V) \subseteq Z_v$, and $\mathbf{J} \in \mathbb{IR}^{n_y \times n}$ is an interval matrix satisfying $\nabla_x^T \mathbf{g}(\square X, \mathbf{u}, V) \subseteq \mathbf{J}$, then

$$\{\mathbf{x} \in X : \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{v}) = \mathbf{y},\ \mathbf{v} \in V\} \subseteq X \cap_{\mathbf{C}} Y,$$

where $\mathbf{C} = \tilde{\mathbf{J}}$, and $Y = (\mathbf{y} + \tilde{\mathbf{J}}\boldsymbol{\gamma}_x) \oplus Z_v \oplus \triangleleft(\tilde{\mathbf{J}} - \mathbf{J}, X - \boldsymbol{\gamma}_x)$.

Note that the constrained zonotopes Z_w and Z_v in Propositions 1 and 2 are obtained using the mean value extension. Moreover, the interval matrices $\mathbf{J}$, $\mathbf{J}_w$ (Proposition 1), and $\tilde{\mathbf{J}}$ (Proposition 2), are all computed using interval arithmetic. In Proposition 2 the free parameter $\tilde{\mathbf{J}}$ is chosen as the center of the interval matrix $\mathbf{J}$. $(\boldsymbol{\gamma}_x, \boldsymbol{\gamma}_w, \boldsymbol{\gamma}_v)$ in Lemma 1 and Propositions 1 and 2, were chosen as $(\boldsymbol{\gamma}_x, \boldsymbol{\gamma}_w, \boldsymbol{\gamma}_v) \in \square X \times \square W \times \square V$. For further details refer to [Rego et al., 2021].

Remark 10 *The constrained zonotope Z_w in Proposition 1 can be obtained using the mean value extension $\mathbf{f}(\boldsymbol{\gamma}_x, \mathbf{u}, W) \subseteq Z_w = \mathbf{f}(\boldsymbol{\gamma}_x, \mathbf{u}, \boldsymbol{\gamma}_w) \oplus \triangleleft(\mathbf{J}_w, W - \boldsymbol{\gamma}_w)$ for a chosen point $\boldsymbol{\gamma}_w \in \square W$, with $\mathbf{J}_w$ being an interval matrix satisfying $\mathbf{J}_w \supseteq \nabla_w^T \mathbf{f}(\boldsymbol{\gamma}_x, \mathbf{u}, \square W)$.*

Remark 11 *The constrained zonotope Z_v in Proposition 2 can be obtained as $Z_v = -\mathbf{g}(\boldsymbol{\gamma}_x, \mathbf{u}, \boldsymbol{\gamma}_v) \oplus \triangleleft(-\mathbf{J}_v, V - \boldsymbol{\gamma}_v) \supseteq -\mathbf{g}(\boldsymbol{\gamma}_x, \mathbf{u}, V)$ for some $\boldsymbol{\gamma}_v \in \square V$ and interval matrix $\mathbf{J}_v \supseteq \nabla_v^T \mathbf{g}(\boldsymbol{\gamma}_x, \mathbf{u}, \square V)$. The matrix $\tilde{\mathbf{J}}$ is a free parameter in Proposition 2. Choosing $\tilde{\mathbf{J}} = mid(\mathbf{J})$ gives $mid(\tilde{\mathbf{J}} - \mathbf{J}) = \mathbf{0}$, and hence $\triangleleft(\tilde{\mathbf{J}} - \mathbf{J}, X - \boldsymbol{\gamma}_x) = mid(\mathbf{J}) \tilde{-} \mathbf{J}(X - \boldsymbol{\gamma}_x) \oplus \mathbf{P}B_\infty^{n_y} = \mathbf{P}B_\infty^{n_y}$, with $\mathbf{P}$ defined as in Theorem 1.*

Remark 12 *The interval matrices $\mathbf{J}$, $\mathbf{J}_w$ (Proposition 1), and $\tilde{\mathbf{J}}$, $\mathbf{J}_v$ in Proposition 2, are all computed using interval arithmetic.*

In the following, set-based state estimation is applied to fault detection in Li-ion batteries.

5.2 Set-based fault detection

One of the critical aspects to consider when operating Li-ion batteries is safety. Several faults can occur during battery operation which could result in significant performance degradation and potentially serious consequences, such as fires or explosions [Tran and Fowler, 2020]. Battery faults can be essentially of three types: cell faults, sensor faults and actuator faults. Cell faults are the most critical ones and include Over-charging (OC), Over-discharging (OD), overheating, External short-circuit (ESC), Internal short-circuit (ISC) and thermal runaway [Bandhauer et al., 2011, Hu et al., 2020]. These faults can also combine and be mutually causal. Sensor faults instead cause issues in the feedback loops of the BMS, that typically relies on sensor measurements [Xiong et al., 2019]. Actuator faults include cooling system faults, bus faults and connector faults [Liu et al., 2014]. The device responsible for handling faults, as well as other operational aspects, is the BMS. The functioning principle is to take measures coming from the battery pack (voltage, current and temperature), analyze them and take decisions. The BMS can help in minimizing the effect of a fault, eliminating it and keeping the battery operation inside a safe range.

Many fault diagnostic algorithms have been developed for Li-ion batteries, which can be divided in two different groups, based on the methodology on which they rely: data-driven and model-based. Data-driven methods are able to detect the faults by analyzing data without relying on a model. In [Xia et al., 2017a] ISC are detected with a method based on the correlation coefficient of voltage curves. A similar approach is proposed in [Li and Wang, 2018], where an interclass correlation coefficient-based method is able to capture the off-trend voltage drops. [Wang et al., 2017] and [Liu et al., 2018] use Shannon

entropy theory to capture thermal abnormalities and predict a fault. In [Xiong et al., 2012], the authors propose a probabilistic rule-based method to find OD failures; the rules to detect faults are based on unusual drops in voltage and an increase in temperature. Big data statistical methods are employed in [Zhao et al., 2017] to find abnormal changes of cell terminal voltages in a battery pack with a neural network algorithm. Note that the tuning of data-driven methods is usually time-consuming and costly, and therefore performed offline. Model-based methods rely on mathematical models in order to capture the physical behaviour of the battery. These approaches are generally preferred to data-driven methods, because they provide a better insight and understanding of the nature of the fault. Moreover, they allow for an understandable design process and continuous validation at all stages of the method development.

Model-based fault detection strategies compare measures coming from the battery pack with the ones generated by the mathematical model. The potential difference is then analyzed to determine whether a fault has occurred. In the literature, several works have tackled this problem using different approaches. In [Dey et al., 2015], the authors analyze thermal faults in a single cell using the residual method with a nonlinear Luenberger-type observer to estimate core temperature. The nominal model is generated by an ECM. For the same conditions, a sliding mode observer is used in [Marcicki et al., 2010] and [Dey et al., 2016] to identify sensor faults. In [Singh et al., 2013] and [Couto and Kinnaert, 2018], respectively OC and OD, internal and sensor faults are identified using a linear Kalman filter as an observer; the first work uses an ECM to generate the nominal measure signals, while the second one uses an EM. [Alavi et al., 2013] uses a particle filter to identify faulty plating phenomena (i.e. sudden capacity fade). Note that in these two latter works, the authors were able to take into account gaussian measurement and model uncertainties. In [Dey et al., 2017] thermal faults are identified using respectively a nonlinear Lyapunov observer and a robust observer, that allow to account for Gaussian uncertainties on the model. A different approach was proposed by [Seo et al., 2017], where the measure signal is compared with the nominal model generated by an ECM and with the faulty model of an ISC (or, in principle, any other fault). If the signal does not coincide with the nominal one, a fault might have occurred. Another work proposed by [Li and Wang, 2018] uses the interclass correlation coefficient to diagnose ISC faults on an entire battery pack, taking into account model uncertainties. Except for [Couto and Kinnaert, 2018], the faults are considered to be abrupt, meaning that they happen instantly and are not incremental.

One of the issues of model-based fault-detection approaches is the fact that the mathematical models have imprecisions and are only an approximation of reality. Moreover, measurement uncertainties provide an additional layer of unpredictability. Most of the approaches proposed in the literature either do not take into account any uncertainty or they rely on the hypothesis that the distribution is known a priori. However, this is not realistic in general.

On the other side, set-based approaches do not require the knowledge of the distributions but rely on the assumption of unknown but bounded uncertainties only. In many cases, e.g. physical bounds, such assumption is reasonable and appears appropriate in the context of battery state estimation.

Set-based fault detection consists in evaluating the consistency of the obtained measurements $\mathbf{y}_k$ with the predicted output set generated by the nominal (healthy) dynamics, i.e. it consists in performing, at each k_t, the inclusion check

$$\mathbf{y}_{k_t} \in \mathbf{g}(\bar{X}_{k_t}, \mathbf{u}_{k_t}, V). \tag{5.23}$$

A fault is therefore detected when the measurement does not lie in the output set anymore, i.e. $\mathbf{y}_{k_t} \notin \mathbf{g}(\bar{X}_{k_t}, \mathbf{u}_{k_t}, V)$ or, equivalently, when the update step (5.20) results in $\hat{X}_{k_t}$ being empty. The accuracy of the set-based estimation is fundamental in order to achieve an early fault detection thus limiting the consequences related to the fault. The motivation for using constrained zonotopes is therefore apparent. Since they can provide tighter enclosures compared to zonotopes and intervals, at the price of a mild increase in complexity, they can lead to an earlier detection. In the following, the advantages of a set-based approach based on constrained zonotopes over an interval method based on a forward-backward algorithm are shown.

5.2.1 Single cell set-based fault detection

The contents of this Chapter are based on publication **P.2**

Consider the ECM presented in Section 2.4, an in particular, the DP model, consisting of two RC blocks. The relevant model dynamics are reported in the following.

$$\begin{cases} \dfrac{dV_1(t)}{dt} = -\dfrac{1}{R_1C_1}V_1(t) + \dfrac{1}{C_1}I_{cell}(t), \\ \dfrac{dV_2(t)}{dt} = -\dfrac{1}{R_2C_2}V_2(t) + \dfrac{1}{C_2}I_{cell}(t), \\ \dfrac{d\mathrm{SOC}(t)}{dt} = -\dfrac{I_{cell}(t)}{3600C_{cell}}, \\ \dfrac{dT_c(t)}{dt} = \dfrac{T_s(t) - T_c(t)}{R_cC_c} + \dfrac{Q(t)}{C_c}, \\ \dfrac{dT_s(t)}{dt} = \dfrac{T_f(t) - T_s(t)}{R_uC_c} + \dfrac{T_s(t) - T_c(t)}{R_cC_s}, \\ V_{cell}(t) = V_{EMF}(\mathrm{SOC}(t)) + V_1(t) + V_2(t) + R_{int}I_{cell}(t), \end{cases} \tag{5.24}$$

with the function $V_{EMF}(\text{SOC}(t))$ defined as a polynomial expression as

$$\begin{aligned} V_{EMF}(\text{SOC}(t)) = a + b \cdot \text{SOC}(t) + c \cdot \text{SOC}^2(t) + d \cdot \text{SOC}^3(t) + e \cdot \text{SOC}^4(t) + \\ + f \cdot \text{SOC}^5(t) \ + g \cdot \text{SOC}^6(t) + h \cdot \text{SOC}^7(t) + i \cdot \text{SOC}^8(t) + \\ + j \cdot \text{SOC}^9(t), \end{aligned} \tag{5.25}$$

where the coefficients used are shown in Table 5.1. These values have been obtained by interpolation from the work of [Lin et al., 2014b].

Coefficient	a	b	c	d	e
Value	2.611	17.04	-204.4	1369	-5423
Coefficient	f	g	h	i	j
Value	13210	-19970	18260	-9247	1990

Table 5.1: Coefficient values for $V_{EMF}(\text{SOC}(t))$

As mentioned before, different types of faults are possible. In this part of the work, only thermal faults are considered and, in particular, the ones described in [Dey et al., 2015]. Three different faults can be considered:

- *Fault 1*: R_u significantly deviates from its nominal value, representing a convective cooling fault;
- *Fault 2*: R_c significantly deviates from its nominal value, representing an internal thermal resistance fault;
- *Fault 3*: an additional heat generation term contributes to the rise of the core temperature T_c, causing thermal runaway.

The modified thermal model including the presence of these faults is equal to

$$\begin{cases} \dfrac{dT_c(t)}{dt} = \dfrac{T_s(t) - T_c(t)}{(R_c + \Delta R_c)C_c} + \dfrac{Q(t)}{C_c} + \Delta T_c, \\ \dfrac{dT_s(t)}{dt} = \dfrac{T_f(t) - T_s(t)}{(R_u + \Delta R_u)C_c} + \dfrac{T_s(t) - T_c(t)}{(R_c + \Delta R_c)C_s}, \end{cases} \tag{5.26}$$

where ΔR_c and ΔR_u represent respectively *Fault 1* and *Fault 2*, while ΔT_c represents thermal runaway of *Fault 3*. It is assumed that multiple faults cannot occur at the same time. Moreover, it will be assumed that faults happen suddenly, meaning that at a certain time instant, the faulty value will be instantly shifted by a certain amount. Note that faults are mutually causal, and can create a cascading effect. However, in the cases under consideration, the assumption will be that fault detection is fast enough to avoid it.

When dealing with mathematical models, the parameter identification may lead to an imprecision in the estimated parameter values. This fact can be taken into account in the model by considering parametric uncertainties (see

e.g. [Locatelli et al., 2021]). The base value for electrical and thermal parameters (reported in Table 5.2) was taken from [Lin et al., 2014b] and the uncertainty was set for each parameter to ±2.5%. As it happens in a real

Parameter	Value	Unit
C_1	2100	Ah
C_2	70000	Ah
C_{cell}	2.3	Ah
R_{int}	0.01	Ω
R_1	0.02	Ω
R_2	0.02	Ω
R_u	3.08	KW^{-1}
R_c	1.94	KW^{-1}
C_c	62.7	JK^{-1}
C_s	4.5	JK^{-1}

Table 5.2: Electro-thermal model parameters

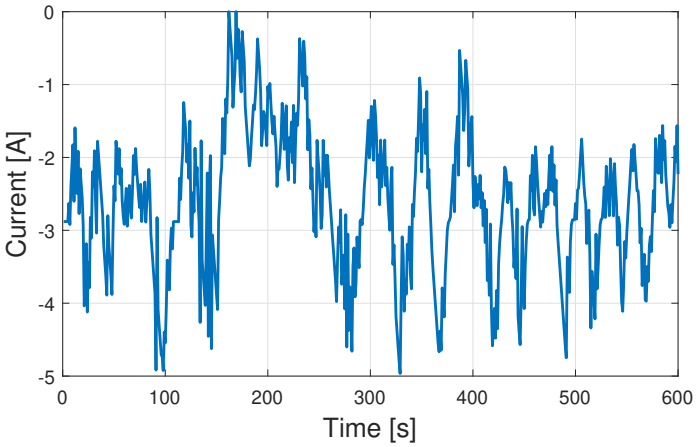


Figure 5.1: Input current profile $u(t)$ applied to the cell.

case scenario, the measurable quantities are terminal voltage V and surface temperature T_s. The measurement noise has been set respectively to ±1 mV and ±0.1 K. The initial state condition is uncertain and lies in the zonotope

$$\bar{X}_0 = \{\mathbf{G}_{x_0}, \mathbf{c}_{x_0}\}, \tag{5.27}$$

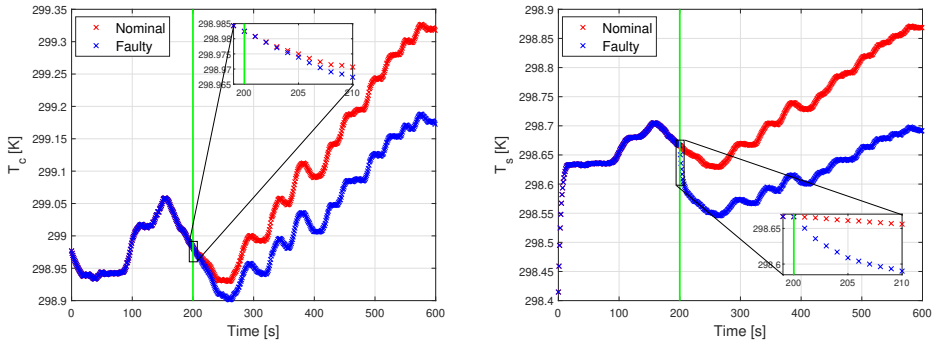
with

$$\mathbf{G}_{x_0} = \text{diag}([0.001\ 0.001\ 0.05\ 1\ 1]), \tag{5.28}$$

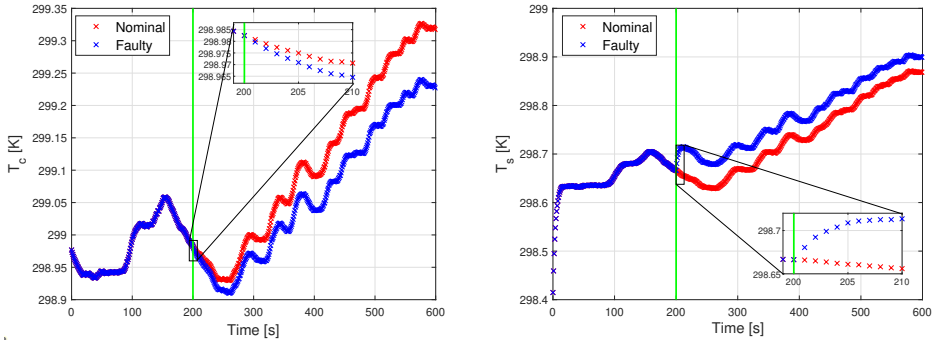
and

$$\mathbf{c}_{x_0} = [0.005\ 0.005\ 0.25\ 298.15\ 298.15]^\top. \tag{5.29}$$

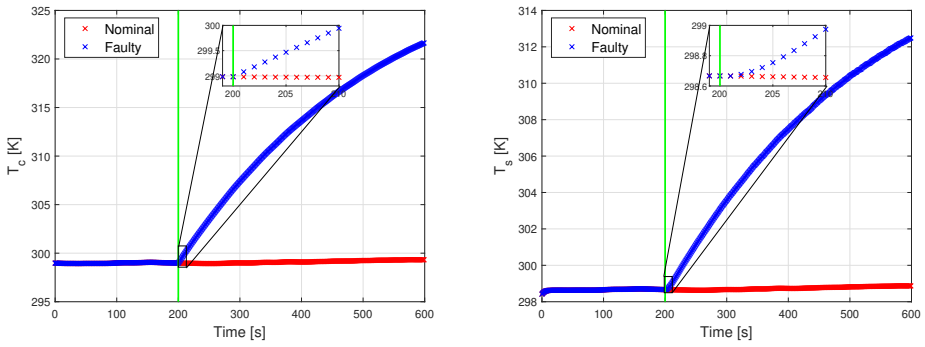
In particular, for the zonotope center the SOC has been set equal to 25% and both core and surface temperature to ambient temperature.



Fault 1 $\Delta R_u = -30\%$



Fault 2 $\Delta R_c = -30\%$



Fault 3 $\Delta T_c = 0.1\ K/s$

Figure 5.4: Comparison between nominal (in red) and faulty thermal states (in blue) for all the three types of faults presented. In green the time instant in which the fault is injected.

All the simulations described in the following rely on the input current profile reported in Figure 5.1. This profile can represent an electric vehicle driving cycle.

The cell parameters used for simulating the healthy cell were extracted randomly within the parameter bounds mentioned above. Similarly, the initial condition used in all the simulations was extracted randomly within $\bar{X}_0$. Measurements were obtained by simulating the cell nominal and faulty models in a time interval of $[0, 600]$ seconds. The fault was injected at $k = 200$ s.

Figure 5.4 shows the effect of the three faults $\Delta R_u = -30\%$, $\Delta R_c = -30\%$ and $\Delta u = 0.1$ K (the percentage variation is from the base value) in comparison with the nominal (healthy) dynamics. After the fault is injected, the state trajectory (in this case only T_c and T_s are depicted) deviates significantly from the nominal one after a few seconds. Note that only the faulty parameter has been changed, all the other parameters are the same for the nominal and faulty cases. While by looking at these figures fault detection would seem an easy task, since not all states are measured and the nominal model is uncertain (besides the measurement uncertainty, the initial condition and the parameter values are not known exactly) the trajectory deviation could be, for some time, explained in terms of model uncertainty, thus making fault detection difficult in practice. Within the context of set-based estimation, strategies able to provide a tighter enclosure of the sets defined in the prediction-update algorithm, such as constrained zonotopes, will be able to alleviate this issue, resulting in an earlier fault detection.

In Figure 5.5 it is shown the constrained zonotopes set-based estimation approach applied in open-loop (i.e. by performing prediction steps (5.19) only) and in closed-loop (i.e. by performing both prediction and update steps (5.20)) to a Li-ion cell model. The figure provides the projection of the sets in the state-space of V_1, V_2 and SOC at $k = \{2, 12, 22, 42, 95\}$ s; in blue are depicted the open-loop sets, while in yellow the closed-loop enclosures (note that, for clarity, not all the time instants have been plotted but only a sample of them, in order to avoid having overlapping sets). As expected, these latter include a much smaller volume, as the system output measurement allows to restrict the feasible state values. Also, notice that, differently from, e.g. intervals, constrained zonotopes are able to capture the dependency between state variables, thus reducing conservatism.

Finally, it is possible to analyze the effectiveness of the proposed method in detecting faults and compare the performance of intervals and constrained zonotopes. In Figure 5.7, the output space (that includes V and T_s) is represented for the case of Fault 1 (similar results have been obtained in the other cases); the red line represents the measure coming from the cell, while in green and yellow respectively we have interval and constrained zonotope sets. These are obtained by a mean value extension of the output function, using the predicted states $\bar{X}_k$, i.e. the set in equation (5.23). Interval sets instead are obtained solving the constraint satisfaction problem and using the forward-backward algorithm. When the measure coming from the system does not lie inside these sets, the fault has been detected. The first observation is that intervals are very conservative in the prediction step: their size (and there-

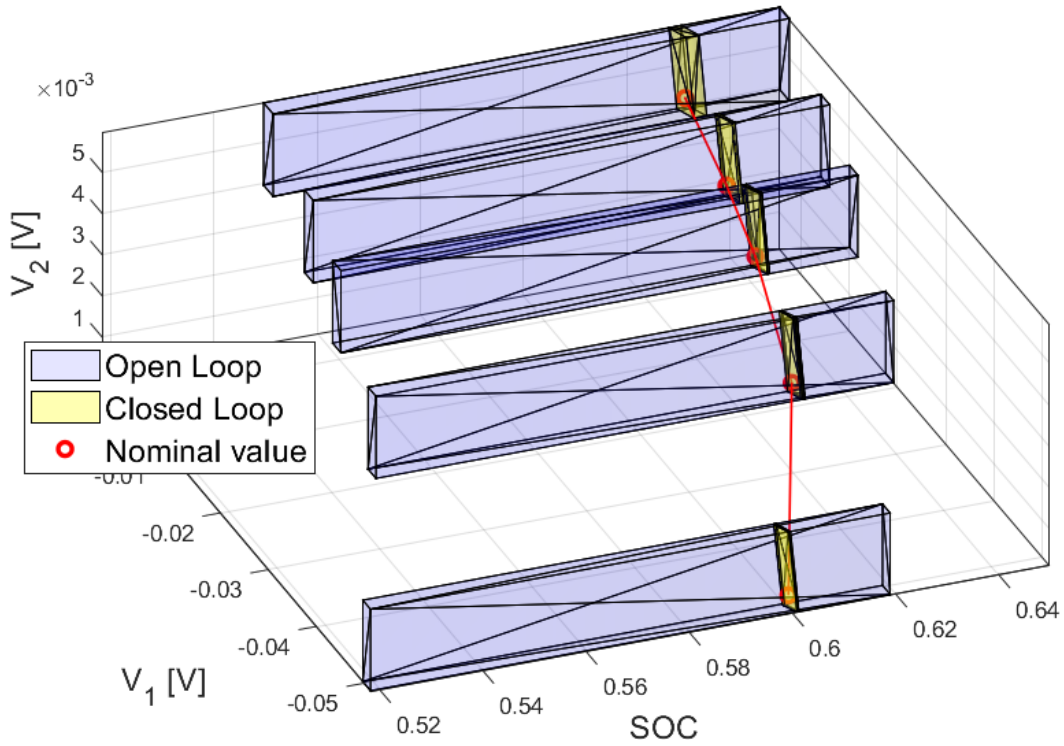


Figure 5.5: Comparison between open-loop and closed-loop state trajectories using constrained zonotopes.

fore, the feasible output values) increases during the simulation, making fault detection practically impossible.

With regards to the time needed to the detect the fault, an analysis was performed varying the magnitude of the fault. The instant in which the fault is detected is provided in Table 5.3 (the resistive fault is in percentage of the base value). Here, it is possible to see that the bigger the fault, the faster the detection time is. For a comparison of computation times of constrained zonotopes and interval sets (and other set representations), refer to [Scott et al., 2016].

Fault	**Magnitude**	**Detection time**
Fault 1	$\Delta R_u = -30\%$	4 s
Fault 2	$\Delta R_c = -30\%$	6 s
Fault 1	$\Delta R_u = -45\%$	3 s
Fault 2	$\Delta R_c = -45\%$	5 s
Fault 3	$\Delta T_c = 0.01\ K/s$	16 s
Fault 3	$\Delta T_c = 0.1\ K/s$	6 s

Table 5.3: Time needed for the developed method to detect the fault.

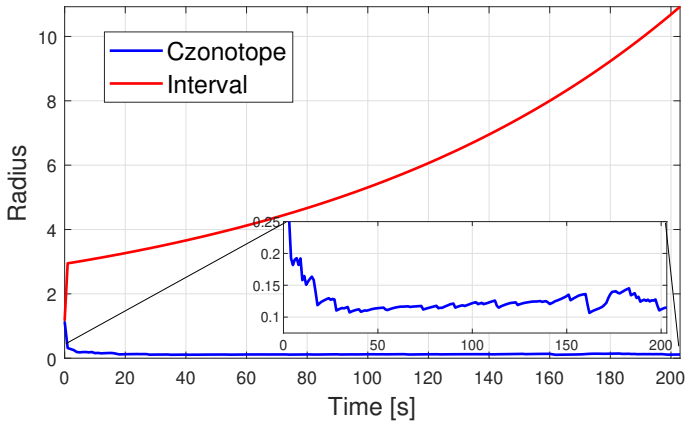


Figure 5.6: Comparison between radii of interval and constrained zonotopes on the output (Fault 1).

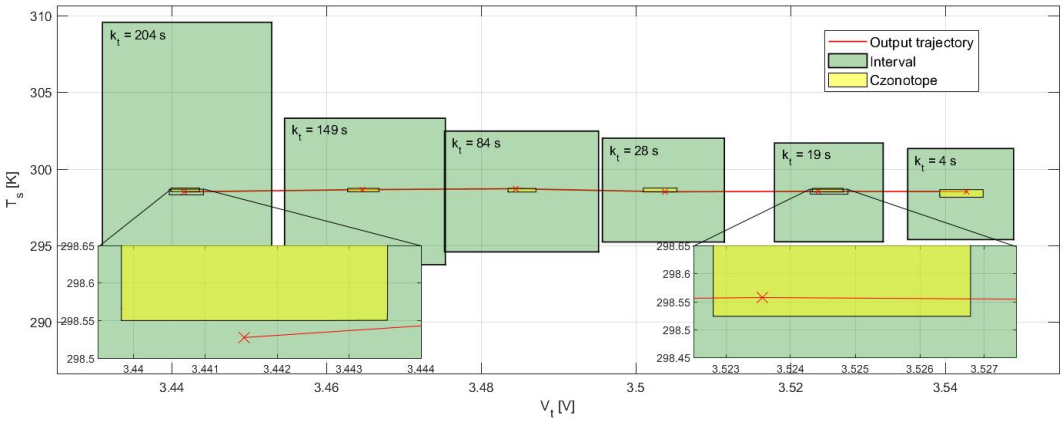


Figure 5.7: Comparison between interval and constrained zonotopes on output trajectories (Fault 1).

In Figure 5.6 the radius[2] of the output sets defined in (5.23), obtained using intervals and constrained zonotopes, are depicted for every time instant within the time window $[0, 200]s$. As it can be noticed, the intervals explode over time. When analyzing the reasons of such explosion, we notice that a significant growth of the bounds along the component of T_c occurs in the prediction phase. This is due to the parameter uncertainty present in the system. While the update phase should reduce such uncertainty, intervals are refined only on the state components that explicitly appear in the output equation (see [Locatelli et al., 2021]). Because this is not the case for T_c, the result is that it never gets refined and the closed-loop and open- loop estimation are the same for that component. A bad estimation of T_c heavily affects also the estimation of the other sets. This fact makes it impossible to

[2]Computed as half the length of the longest edge of the interval hull, as in [Scott et al., 2016].

detect the fault with the interval method, because if the set is big, the measure will always be consistent with the prediction (i.e. lies in the set).

Fault detection methods heavily depend on many factors for their effectiveness; first of all, the instant in which the fault happens, is very relevant. If for example a fault occurs when no current is flowing in the battery pack, there is nothing that could detect the fault. Moreover, other factors to take into account are the input current and uncertainties; the first one because there are some signals that excite the system better than others in order to detect faults. The second one is an issue if there are big uncertainties on the measurement, it is obvious that if the measures are consistent with different operating modes (nominal and faulty), no fault can be detected.

In this part of the work the advantages of constrained zonotopes over intervals in the detection of thermal faults in Li-ion cells were demonstrated. As highlighted by the results, constrained zonotopes are able to provide a set-based estimation which results in tight enclosures where, differently from intervals, it is possible to capture the coupling between the different state components. This feature, even in the presence of parameter uncertainties, allows to detect faults in a very effective way.

5.2.2 Battery pack set-based fault detection

The contents of this Chapter are based on publication **P.3**

The work presented in Section 5.2.1 can be further expanded by taking into consideration a simple battery pack. In particular, a series-connected battery pack is considered here, with coupled thermal dynamics. In addition to the Li-ion cell model presented before, the parameters of the ECM are considered to be dependent from SOC and temperature. Consider the series-connected cells presented in Figure 5.8. The electrical model considered for the single cell is the DP model. To account for the different cells in the battery pack, the following notation is introduced here: $\{h\}$ at the subscript will indicate the number of the cell in the battery pack, with $h \in \{1, \dots, N\}$, N being the total number of cells. Note that, since only series-connected cells are considered, the current flowing in the pack, $\mathbf{I}_{tot}(t)$, will be the same for all the cells ($\{h\}$ at the subscript is therefore omitted). $\mathbf{I}_{tot}(t)$ is the only input of the considered system, assumed positive when charging.

For the h-th cell, the dynamical model of the electrical part consists of the

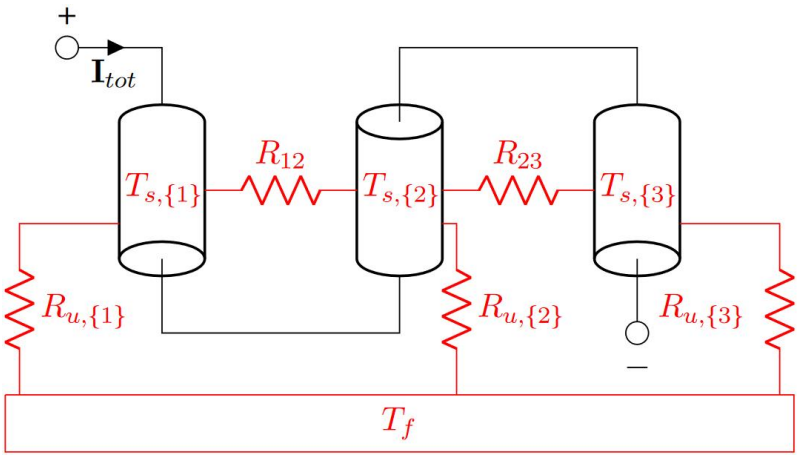


Figure 5.8: Series-connected cells with thermal resistances.

following set of equations

$$\frac{dV_{1,\{h\}}(t)}{dt} = -\frac{V_{1,\{h\}}(t)}{R_{1,\{h\}}(\mathrm{SOC}_{\{h\}}(t), T_{c,\{h\}}(t))C_{1,\{h\}}(\mathrm{SOC}_{\{h\}}(t), T_{c,\{h\}}(t))} + \\ + \frac{\mathbf{I}_{tot}(t)}{C_{1,\{h\}}(\mathrm{SOC}_{\{h\}}(t), T_{c,\{h\}}(t))}, \tag{5.30}$$

$$\frac{dV_{2,\{h\}}(t)}{dt} = -\frac{V_{2,\{h\}}(t)}{R_{2,\{h\}}(\mathrm{SOC}_{\{h\}}(t), T_{c,\{h\}}(t))C_{2,\{h\}}(\mathrm{SOC}_{\{h\}}(t), T_{c,\{h\}}(t))} + \\ + \frac{\mathbf{I}_{tot}(t)}{C_{2,\{h\}}(\mathrm{SOC}_{\{h\}}(t), T_{c,\{h\}}(t))}, \tag{5.31}$$

$$\frac{d\mathrm{SOC}_{\{h\}}(t)}{dt} = \frac{\mathbf{I}_{tot}(t)}{3600 C_{cell,\{h\}}}. \tag{5.32}$$

The cell terminal voltage is given by

$$V_{\{h\}}(t) = V_{\mathrm{EMF},\{h\}}(\mathrm{SOC}, t) + V_{1,\{h\}}(t) + V_{2,\{h\}}(t) + R_{int,\{i\}}(T_{c,\{h\}}(t))\mathbf{I}_{tot}(t). \tag{5.33}$$

The formulation of $V_{\mathrm{EMF},\{h\}}(\mathrm{SOC}, t)$ is the same of Equation (5.25), with the coefficients reported in Table 5.1. Finally, the voltage across the terminals of the battery pack, here consisting of series connected cells only, is given by

$$V_t(t) = \sum_{h=1}^{N} V_{\{h\}}(t). \tag{5.34}$$

For the electrical part of the system it is assumed that only $V_t(t)$ can be measured. The cell thermal model is a two state system, that includes, for

each cell, core and surface temperature

$$\frac{dT_{c,\{h\}}(t)}{dt} = \frac{|\mathbf{I}_{tot}(t)||V_{1,\{h\}}(t) + V_{2,\{h\}}(t) + R_{int,\{i\}}(T_{c,\{h\}}(t))\mathbf{I}_{tot}(t)|}{C_{c,\{h\}}} + \\ + \frac{T_{s,\{h\}}(t) - T_{c,\{h\}}(t)}{R_{c,\{h\}}C_{c,\{h\}}}, \tag{5.35}$$

$$\frac{dT_{s,\{h\}}(t)}{dt} = -\frac{T_{s,\{h\}}(t) - T_f(t)}{R_{u,\{h\}}C_{s,\{h\}}} - \frac{T_{s,\{h\}}(t) - T_{c,\{h\}}(t)}{R_{c,\{h\}}C_{s,\{h\}}} \\ + \sum_{h\neq j} \frac{T_{s,\{j\}}(t) - T_{s,\{h\}}(t)}{R_{hj}C_{s,\{h\}}}, \tag{5.36}$$

where $R_{u,\{h\}}$, R_{hj} and $C_{s,\{i\}}$ indicate respectively the thermal resistance between the surface of the cell and the coolant, the thermal resistance between adjacent cells and the thermal capacity of the surface. The coolant dynamics are given by the following expression

$$\frac{dT_f(t)}{dt} = \sum_{h=1}^{N} \left(\frac{T_{s,\{h\}}(t) - T_f(t)}{R_{u,\{h\}}C_s} - \frac{\zeta(t)}{C_s} \right), \tag{5.37}$$

with $\zeta(t)$ indicating the thermal power released by the coolant to the external environment an C_u the thermal capacity of the element at temperature T_f. In order to better capture the dynamical behaviour of the battery pack, resistive and capacitive elements of each cell are considered to be dependent from SOC and temperature [Lin et al., 2014b]. The parameters are expressed with Arrhenius-like functions. $R_{int,\{h\}}$ depends exclusively from core temperature

$$R_{int,\{h\}}(\cdot) = a_{R_0} + \exp\left(\frac{b_{R_0}}{T_{c,\{h\}}(t) - c_{R_0}} \right). \tag{5.38}$$

For the other resistive elements, the relationship is given by

$$R_{1,\{h\}}(\cdot) = \left(a_{R_1} + b_{R_1}\mathrm{SOC}_{\{h\}}(t) + c_{R_1}\mathrm{SOC}^2_{\{h\}}(t) \right) \exp\left(\frac{d_{R_1}}{T_{c,\{h\}}(t) - e_{R_1}} \right), \tag{5.39}$$

$$R_{2,\{h\}}(\cdot) = \left(a_{R_2} + b_{R_2}\mathrm{SOC}_{\{h\}}(t) + c_{R_2}\mathrm{SOC}^2_{\{h\}}(t) \right) \exp\left(\frac{d_{R_2}}{T_{c,\{h\}}(t) - e_{R_2}} \right). \tag{5.40}$$

The following expression holds for the capacitors:

$$C_{1,\{h\}}(\cdot) = a_{C_1} + d_{C_1}T_{c,\{h\}}(t) + \left(b_{C_1} + e_{C_1}T_{c,\{h\}}(t)\right)\mathrm{SOC}_{\{h\}}(t) + \\ + \left(c_{C_1} + f_{C_1}T_{c,\{h\}}(t)\right)\mathrm{SOC}^2_{\{h\}}(t), \tag{5.41}$$

$$C_{2,\{h\}}(\cdot) = a_{C_2} + d_{C_2}T_{c,\{h\}}(t) + \left(b_{C_2} + e_{C_2}T_{c,\{h\}}(t)\right)\mathrm{SOC}_{\{h\}}(t) + \\ + \left(c_{C_2} + f_{C_2}T_{c,\{h\}}(t)\right)\mathrm{SOC}^2_{\{h\}}(t). \tag{5.42}$$

The identification, parametrization and validation of these parameters were done experimentally and the full results are presented in [Lin et al., 2014b].

In order to reduce the conservativeness arising from the use of the mean value theorem in the set-based estimation procedure, in the following, after having discretized the model using the Forward Euler's method with sampling time k_t, the uncertain electrical and thermal parameters in the system equations (5.30)-(5.32) and (5.35)-(5.37) have been replaced with their reciprocals which are denoted by $Y_{\{\cdot\}}$. For what concerns the electrical parameters, in order to preserve the dependency of $Y_{\{\cdot\}}$ from temperature and SOC, a fourth degree polynomial expression is adopted for each cell h with the form

$$\begin{aligned} Y_{\{\cdot\},h} =& p_{00} + p_{10}\cdot\mathrm{SOC}_{\{h\}}(t) + p_{01}\cdot T_{c,\{h\}}(t) + p_{20}\cdot\mathrm{SOC}^2_{\{h\}}(t) + \\ &+ p_{11}\cdot\mathrm{SOC}_{\{h\}}(t)\cdot T_{c,\{h\}}(t) + p_{02}\cdot T^2_{c,\{h\}}(t) + p_{30}\cdot\mathrm{SOC}^3_{\{h\}}(t) + \\ &+ p_{21}\cdot\mathrm{SOC}^2_{\{h\}}(t)\cdot T_{c,\{h\}}(t) + p_{12}\cdot\mathrm{SOC}_{\{h\}}(t)\cdot T^2_{c,\{h\}}(t) + \\ &+ p_{03}\cdot T^3_{c,\{h\}}(t) + p_{40}\cdot\mathrm{SOC}^4_{\{h\}}(t) + p_{31}\cdot\mathrm{SOC}^3_{\{h\}}(t)\cdot T_{c,\{h\}}(t) + \\ &+ p_{22}\cdot\mathrm{SOC}^2_{\{h\}}(t)\cdot T^2_{c,\{h\}}(t) + p_{13}\cdot\mathrm{SOC}_{\{h\}}(t)\cdot T^3_{c,\{h\}}(t), \end{aligned} \tag{5.43}$$

whose coefficients p were obtained from a fitting procedure of the curves $1/C_{1,\{h\}}$, $1/C_{2,\{h\}}$, $1/(R_{1,\{h\}}C_{1,\{h\}})$ and $1/(R_{2,\{h\}}C_{2,\{h\}})$. These quantities will be referred to as $Y_{C_1,\{h\}}$, $Y_{C_2,\{h\}}$, $Y_{R_1C_1,\{h\}}$ and $Y_{R_2C_2,\{h\}}$ respectively. The resulting discretized system, assuming $N = 3$ series-connected cells, is summarized in (5.44) as

$$\begin{aligned} x_{1,\{h\}}(k_t+1) &= x_{1,\{h\}}(k_t) - t_s\left[(Y_{R_1C_1,\{h\}} + w_{R_1C_1,\{h\}})\cdot x_{1,\{h\}}(k_t) + (Y_{C_1,\{h\}} + \right. \\ &\quad \left. + w_{C_1,\{h\}})u(k_t)\right], \\ x_{2,\{h\}}(k_t+1) &= x_{2,\{h\}}(k_t) - t_s\left[(Y_{R_2C_2,\{h\}} + w_{R_2C_2,\{h\}})\cdot x_{2,\{h\}}(k_t) + (Y_{C_2,\{h\}} + \right. \\ &\quad \left. + w_{C_2,\{h\}})u(k_t)\right], \\ x_{3,\{h\}}(k_t+1) &= x_{3,\{h\}}(k_t) + t_s(Y_{C_{cell},\{h\}} + w_{C_{cell}})u(k_t), \\ x_{4,\{h\}}(k_t+1) &= x_{4,\{h\}}(k_t) + t_s\left[(Y_{C_c} + w_{C_c})|u(k_t)|(x_{1,\{h\}}(k_t) + x_{2,\{h\}}(k_t) + \right. \\ &\quad + (R_{int,\{h\}} + w_{R_{int},\{h\}})u(k_t)| + (Y_{R_c} + w_{R_c})\cdot(Y_{C_c} + w_{C_c})\cdot \\ &\quad \left. \cdot(x_{5,\{h\}}(k_t) - x_{4,\{h\}}(k_t))\right], \end{aligned}$$

$$
\begin{aligned}
x_{5,\{h\}}(k_t+1) &= x_{5,\{h\}}(k_t) + t_s\left[(Y_{R_u}+w_{R_u})(Y_{C_s}+w_{C_s})(x_6(k_t)-x_{5,\{h\}}(k_t))+\right.\\
&+ (Y_{R_c}+w_{R_c})(Y_{C_s}+w_{C_s})\cdot(x_{4,\{h\}}(k_t)-x_{5,\{h\}}(k_t))+\\
&\left.+\sum_{h\neq j}(Y_{R_{hj}}+w_{R_{hj}})(Y_{C_s}+w_{C_s})\cdot(x_{5,\{j\}}(k_t)-x_{5,\{h\}}(k_t))\right],\\
x_6(k_t+1) &= x_6(k_t) + t_s\left[(Y_{R_u}+w_{R_u})(Y_{C_s}+w_{C_s})\sum_{h=1}^{3}(x_{5,\{h\}}(k_t)-x_6(k_t))\right],\\
y_1(k_t) &= \sum_{h=1}^{3} V_{EMF,\{h\}}(x_{3,\{h\}}(k_t))x_{1,\{h\}}(k_t) + x_{2,\{h\}}(k_t) + (R_{int,\{h\}}+\\
&+ w_{R_{int,\{h\}}})u(k_t) + v_1(k_t),\\
y_2(k_t) &= x_{5,\{2\}}(k_t) + v_2(k_t),
\end{aligned}
\tag{5.44}
$$

with

$$
\mathbf{x}_{\{h\}} = \begin{bmatrix} V_{1,\{h\}} & V_{2,\{h\}} & \mathrm{SOC}_{\{h\}} & T_{c,\{h\}} & T_{s,\{h\}}. \end{bmatrix} \tag{5.45}
$$

and $\mathbf{x} = \begin{bmatrix}\mathbf{x}_{\{1\}} & \mathbf{x}_{\{2\}} & \mathbf{x}_{\{3\}} & x_6\end{bmatrix}$, indicate the state variables for each cell; x_6 indicates the coolant temperature T_f. Note that ζ was set to zero, as it would happen in an active cooling system, where a fan kicks in only if T_f reaches a certain threshold. In this case it is assumed that the threshold is never reached. u indicates the input current I. In order to account for the fitting error, measurement noise, as well as the uncertainties arising from the identification procedure of both thermal and electrical parameter values, the model includes the presence of additive parameter uncertainties $w_{\{\}}$. Finally, the output of the system, indicated as $\mathbf{y} = [y_1 \quad y_2]$, includes respectively $V_t(k_t)$ (see Equation (5.34)) and $T_{s,\{2\}}$, i.e. only the surface temperature of the second cell is being measured. Note that the outputs are assumed affected by measurement noise v. In the following, both parameter and measurement noise will be considered unknown but bounded. The faults considered are the same of Section 5.2.1, to which only the subscript $\{h\}$ is added to indicate the cell on which the fault has occurred.

In the following, the performance of constrained zonotopes in detecting the presence of faults is evaluated. Tables 5.4 and 5.5 list the nominal values and the uncertainties of the thermal and electrical parameters. The initial state set of each single cell $\bar{X}_{0,\{h\}}$ is a zonotope described in G-rep by:

$$
X_{0,\{h\}} = \left\{ \begin{bmatrix} 0.005 \\ 0.005 \\ 0.5 \\ 303.15 \\ 303.15 \end{bmatrix}, \ \mathrm{diag}\begin{bmatrix} 0.0005 \\ 0.0005 \\ 0.0125 \\ 1 \\ 1 \end{bmatrix} \right\}, \tag{5.46}
$$

while for T_f (i.e. x_6) the initial temperature was set to $293.15K \pm 0.25K$. As it can be noticed, the parameter values and the initial conditions have been assumed the same for each cell. However, the fault detection strategy could be

generalized to the more realistic case where inevitable differences are present. The output measurements are affected by noise $v \in V$ with

$$V = \left\{ \begin{bmatrix} 0 & 0 \end{bmatrix}, \ \text{diag} \begin{bmatrix} 0.001 & 0.1 \end{bmatrix} \right\}. \tag{5.47}$$

Parameter	Center	Unit	Uncertainty	Value
C_{cell}	2.3	Ah	$w_{C_{cell}}$	± 0.0115
Y_{R_c}	0.5155	WK^{-1}	w_{R_c}	± 0.0026
$Y_{R_{u,\{2\}}}$	0.0544	WK^{-1}	$w_{R_{u,\{2\}}}$	± 0.0016
$Y_{R_{u,\{1\},\{3\}}}$	0.0544	WK^{-1}	$w_{R_{u,\{1\},\{3\}}}$	$\pm 2.72 \cdot 10^{-4}$
$Y_{R_{hj}}$	0.1	WK^{-1}	$w_{R_{hj}}$	± 0.05
Y_{C_c}	0.0159	KJ^{-1}	w_{C_c}	$\pm 7.97 \cdot 10^{-5}$
Y_{C_s}	0.2222	KJ^{-1}	w_{C_s}	± 0.0011

Table 5.4: ECM model parameters centers and uncertainties.

Uncertainty	Unit	Value
$w_{R_{int}}$	Ω^{-1}	$\pm 5 \cdot 10^{-5}$
$w_{R_1C_1}$	$(As\,\Omega)^{-1}$	$\pm 1.15 \cdot 10^{-4}$
$w_{R_2C_2}$	$(As\,\Omega)^{-1}$	$\pm 3.5 \cdot 10^{-6}$
w_{C_1}	$(As)^{-1}$	$\pm 2.381 \cdot 10^{-6}$
w_{C_2}	$(As)^{-1}$	$\pm 7.143 \cdot 10^{-8}$

Table 5.5: ECM parameter uncertainties

Measurement data have been collected by simulating (5.44) with $\mathbf{x}_0$ randomly chosen from X_0. Parameter and measurement disturbances are generated according to uniform random distributions. The known input profile is a constant discharging current $\mathbf{I}_{tot}(k_t) = -2.3$ A, $\forall\, k_t \geq 0$.

For each of the faults described previously, two different magnitudes have been considered. Fault 1 was injected using as values $\Delta T_{c\{h\}} = 0.1K/s$ and $\Delta T_{c\{h\}} = 0.2K/s$, faults 2 and 3 respectively with $\Delta R_{u,\{h\}} = 100\%$, $\Delta R_{u,\{h\}} = 250\%$, $\Delta R_{c,\{h\}} = 70\%$ and $\Delta R_{c,\{i\}} = 250\%$. For all the cases listed above it is assumed that the fault occurs $10s$ after the start of the simulation. Three different set representations are used and their performance compared:

- constrained zonotopes with a maximum number of generators and constraints of $n_{g,\max} = 312$ and $n_{c,\max} = 40$. This method will be indicated with CZ(40,17) (from Equation (5.21) the order is 17);
- zonotopes obtained using the constrained zonotopic method in [Rego et al., 2021], combined with a reduction to zero of the constraints. In order to keep the same complexity of CZ(40,17), the maximum number of generators is imposed equal to $n_{g,\max} = 272$. This approach is indicated with CZ(0,17);

- zonotopes, indicated with Z(17), where $n_{g,\max} = 272$ is imposed to obtain the same complexity order of the other methods. It is obtained by applying the state estimation strategy in [Alamo et al., 2008].

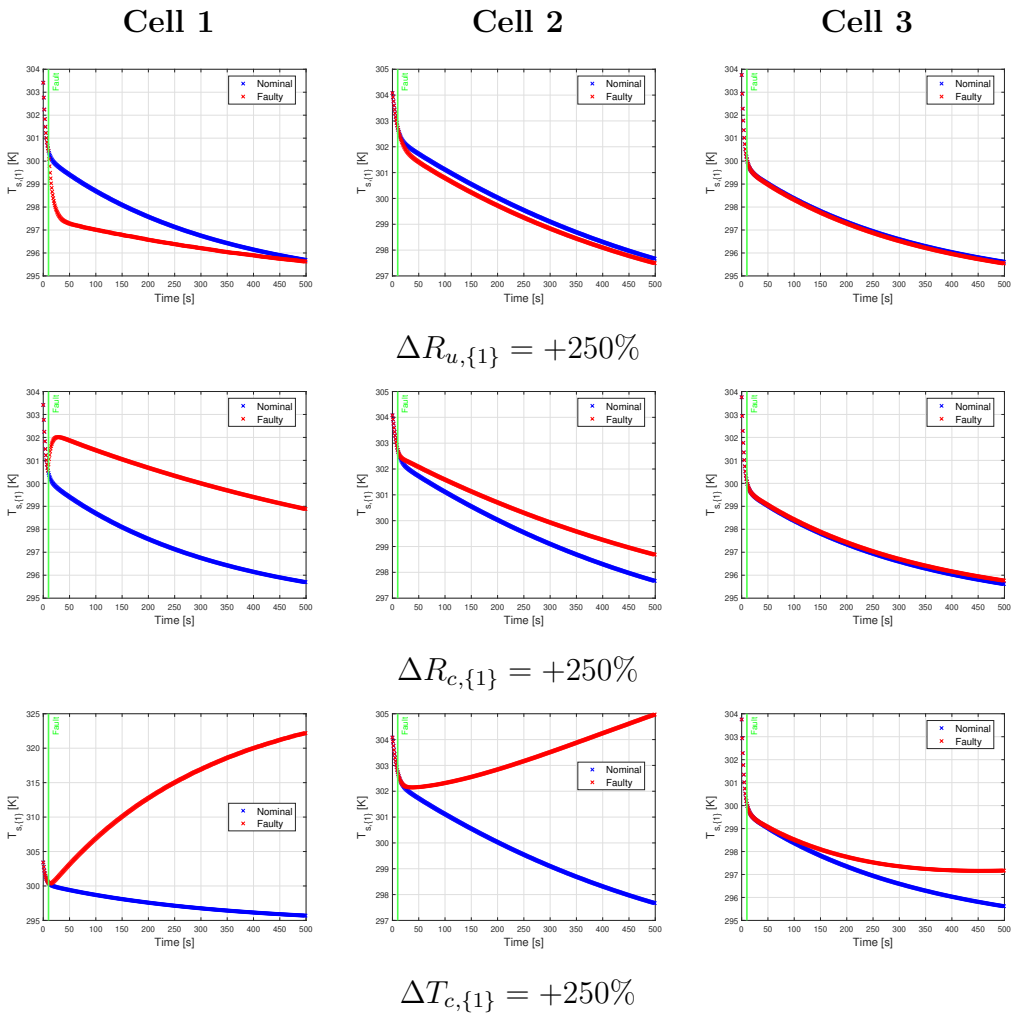


Figure 5.11: Comparison between nominal (blue) and faulty (red) thermal states for each cell and for all three types of faults presented. The green line represents the fault injection time (k_t=10).

The following examples assume that the fault is always on cell 1, while the sensor is placed centrally on cell 2. Figure 5.11 shows the state trajectories of $T_{s,\{h\}}(t)$ for all the cells in presence of the different types of faults. In blue the nominal trajectory without any fault, while the red line represents the state trajectory in presence of a fault. It should be pointed out that, even though the cells are thermally coupled to each other, if a fault occurs in cell 1, it may be difficult to identify it if the sensor is placed on cell 3. Figure 5.14 shows the projected sets $\hat{X}_{k_t}$, $k_t = 18$, on the T_c and T_s components, obtained using the three different set-based strategies. It is worth analyzing this figure in

combination with Table 5.3, where the time at which the fault is detected is listed.

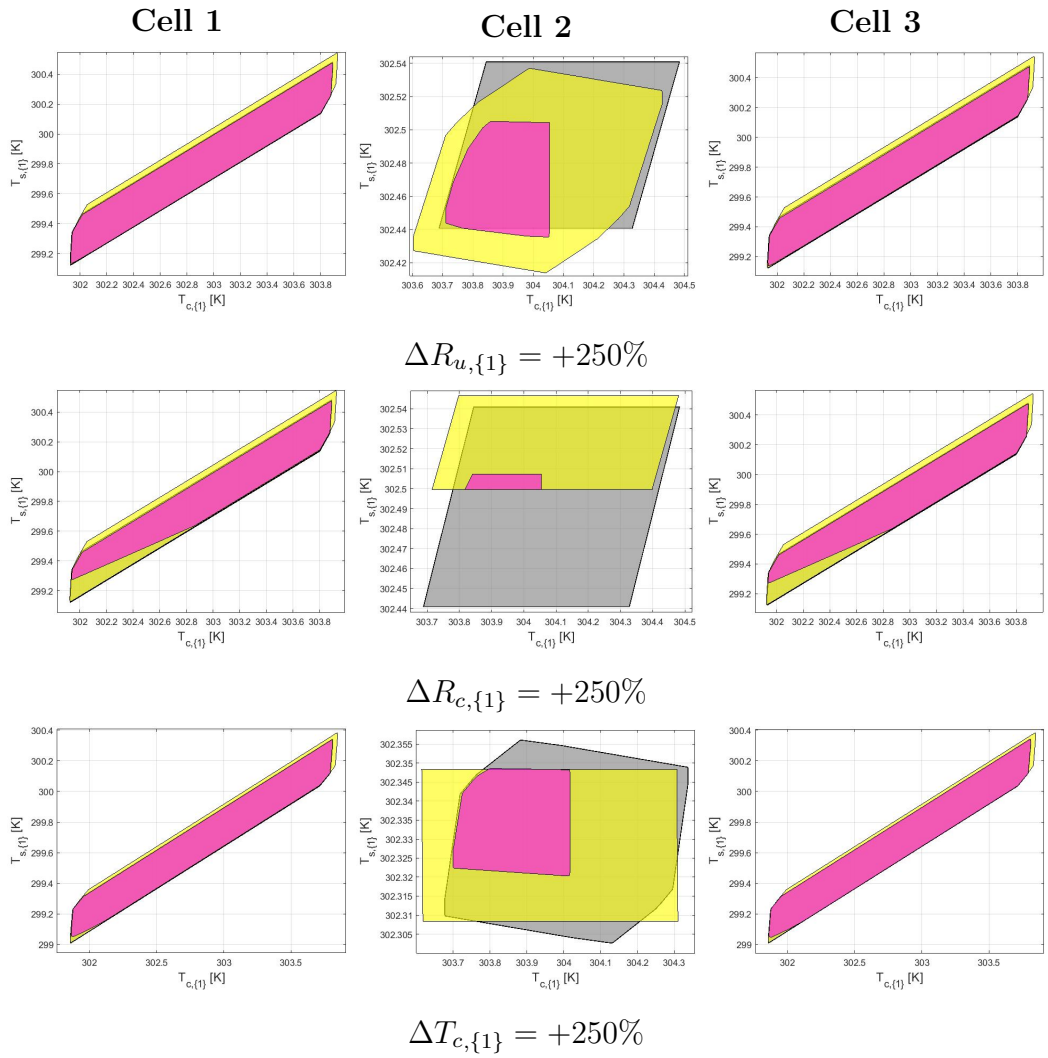


Figure 5.14: Comparison between CZ(40,17) (magenta), CZ(0,17) (yellow) and Z(17) (grey) on thermal states. The first two faults refers to $k_t = 15$, while the last one to $k_t = 18$. The fault is injected in cell 1 and the sensor is placed on cell 2.

CZ(40,17), in magenta, proves to be the most effective strategy in comparison with CZ(0,17) (in yellow) and Z(17) (in grey). Specifically, CZ(0,17) presents cases in which the fault is never found, and Z(17) is always an ineffective strategy. CZ(40,17) provides tighter enclosures that have the advantage of being able to identify sooner when sets of states consistent with the measurements are no longer consistent with the model and disturbances. Another way of comparing the performance of the proposed approach with the other set-based methods is to compare the radii of the enclosures. Figures 5.15 and 5.16 provide the radii (computed as the sum of the absolute value of all the

edges of the interval hull (1-norm), see [Scott et al., 2016] for more details) of the sets $\hat{X}_{k_t}$, $k_t \geq 0$. This figures also show, this time with respect to all state components, the advantage of using constrained zonotopes, even with zero constraints, over zonotopes in state estimation and, consequently, in fault detection.

The results comparing the different proposed reduction techniques are applied and can be seen in Table 5.6. It is possible to appreciate the advantage constrained zonotopes have over zonotopes and constrained zonotopes with zero constraints. In fact, zonotopes are never able to detect the fault, while CZ(0,17) only in a limited number of cases.

Fault	**CZ(40,17)**	**CZ(0,17)**	**Z(17)**
$\Delta T_{c,\{1\}} = +0.1\ K$	26 s	73 s	Not found
$\Delta T_{c,\{1\}} = +0.2\ K$	19 s	22 s	Not found
$\Delta R_{u,\{1\}} = +250\%$	7 s	11 s	Not found
$\Delta R_{u,\{1\}} = +100\%$	11 s	Not found	Not found
$\Delta R_{c,\{1\}} = +250\%$	6 s	Not found	Not found
$\Delta R_{c,\{1\}} = +70\%$	14 s	Not found	Not found

Table 5.6: Time needed to detect the fault.

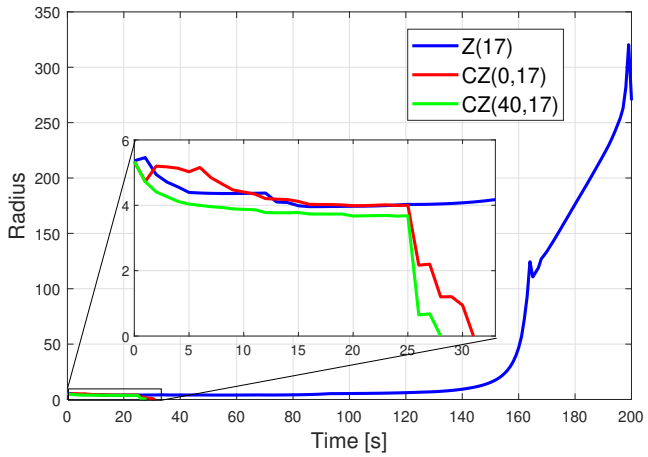
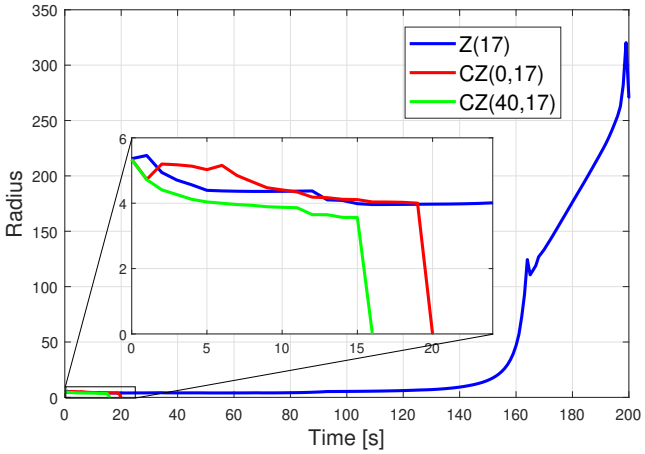
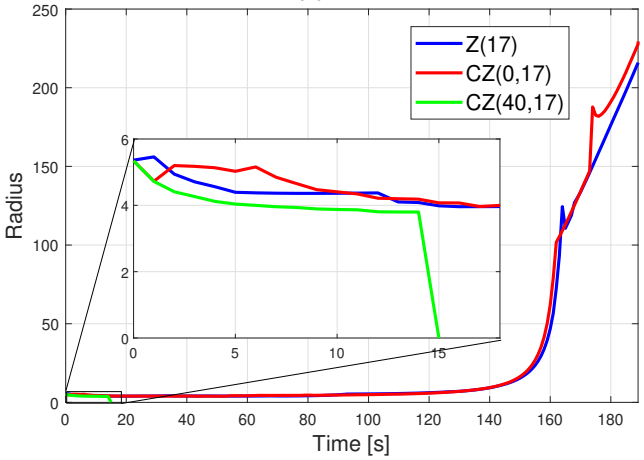


Figure 5.15: Comparison between the radii of CZ(40,17), CZ(0,17) and Z(17) of the states for fault $\Delta T_{c,\{1\}} = +0.2\ K$.



(a) $\Delta R_{u,\{1\}} = +250\%$.



(b) $\Delta R_{c,\{1\}} = +250\%$.

Figure 5.16: Comparison between the radii CZ(40,17), CZ(0,17) and Z(17) of the states for different types of faults.

5.3 Joint set-based state and parameter estimation

The contents of this Chapter are based on publication **P.4**

Mathematical models typically include a large number of unknown parameters, therefore parameter estimation is a critical step in gaining a better understanding of the process and in refining the state estimation phase. Several different methods have been proposed in the literature: a LS algorithm is used in [Lai et al., 2021] for the parameters of an ECM, while in [Pozzi et al., 2018] an optimal design of experiment is employed to improve the parameters identifiability; a recursive LS is used instead in [Xia et al., 2016]. The work in [Jackey et al., 2013] proposed a layered optimization technique based on Matlab/Simulink. All of these works provide a single value as estimation, not a guaranteed interval of possible values. This problem can be solved using, for example, set-based methods. For instance, in [Zhou et al., 2020], interval-based methods are proposed for the identification of ECM parameters.

Some state estimation approaches are able to improve the accuracy by refining the model parametric uncertainties. This method is referred to as joint state and parameter estimation. Within this context, some studies have applied to ECM models the use of extended Kalman filtering through sensitivity analysis [Rauh and Aschemann, 2012], and sigma point Kalman filter [Plett, 2006]. Moreover, particle filtering has been used in [Samadi et al., 2013] for online estimation, while [Zhang et al., 2018] has proposed a decoupled least-square estimation technique.

In this Section, a set-based joint state and parameter estimation approach is presented for discrete-time nonlinear systems based on constrained zonotopes, applied to an electro-thermal model of a Li-ion battery. The method is an extension of the work proposed in [Rego et al., 2022] to the case of nonlinear outputs. The results obtained are compared with the interval-based estimator proposed in [Locatelli et al., 2021] which has been extended to include parameter estimation in a unified framework. Unlike interval-based techniques, constrained zonotopes are able to capture mutual dependencies between states and parameters. This results in tighter bounds of the system state and parameter trajectories with respect to intervals, that is an advantage for Li-ion cell fault detection and model-based control.

Consider the nonlinear discrete-time dynamical system

$$\mathbf{x}_{k_t} = \mathbf{f}(\mathbf{x}_{k_t-1}, \mathbf{u}_{k_t-1}, \mathbf{p}, \mathbf{w}_{k_t-1}) \qquad k_t \geq 1, \tag{5.48a}$$

$$\mathbf{y}_{k_t} = \mathbf{g}(\mathbf{x}_{k_t}, \mathbf{u}_{k_t}, \mathbf{p}, \mathbf{v}_{k_t}) \qquad k_t \geq 0, \tag{5.48b}$$

where $\mathbf{x}_{k_t} \in \mathbb{R}^n$ is the system state, $\mathbf{u}_{k_t} \in \mathbb{R}^{n_u}$ the known input signal, $\mathbf{w}_{k_t} \in \mathbb{R}^{n_w}$ the process uncertainties, $\mathbf{p} \in \mathbb{R}^{n_p}$ the uncertain parameters, $\mathbf{y}_{k_t} \in \mathbb{R}^{n_y}$

the measured output and $\mathbf{v}_{k_t} \in \mathbb{R}^{n_v}$ the measurement uncertainties (assumed to be additive). The nonlinear functions $\mathbf{f}$ and $\mathbf{g}$ are assumed to be of class $\mathcal{C}^1$.

The initial state, the model parameters and the disturbances are assumed to be bounded, i.e. $\mathbf{x}_0 \in X_0$, $\mathbf{p} \in P$, $\mathbf{w}_{k_t} \in W$ and $\mathbf{v}_{k_t} \in V$, with X_0, P, W, V known compact sets. Joint state-parameter set-based estimation aims to compute at each $k_t \geq 0$, the set of all state and parameters $(\mathbf{x}_{k_t}, \mathbf{p})$, denoted Z_{k_t}, consistent with the initial uncertainties (on state and parameters), the known input, the model uncertain dynamics and the obtained measurements. In particular, in the following a slight modification of the previously presented prediction-update algorithm is detailed. It makes use of the relationship between the output $\mathbf{y_{k_t}}$ and the state $\mathbf{x}_{k_t-1}$. This can be easily obtained by replacing $\mathbf{x}_{k_t}$ in (5.48b) with its solution (5.48a) thus leading to

$$\mathbf{y}_{k_t} = \tilde{\mathbf{g}}(\mathbf{x}_{k_t-1}, \mathbf{u}_{k_t-1}, \mathbf{u}_{k_t}, \mathbf{p}, \mathbf{w}_{k_t-1}, \mathbf{v}_{k_t}), \quad k_t \geq 1. \tag{5.49}$$

Then, the prediction-update algorithm, can be formulated as follows. At time $k_t = 0$, measurement $\mathbf{y}_0$ is used to refine the initial set $X_0 \times P$

$$\bar{Z}_0 = X_0 \times P, \tag{5.50}$$

$$\hat{Z}_0 \supseteq \{(\mathbf{x}_0, \mathbf{p}) \in \bar{Z}_0 : \mathbf{g}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{p}, \mathbf{v}_0) = \mathbf{y}_0,\ \mathbf{v}_0 \in V\}. \tag{5.51}$$

From time $k_t = 1$ onward, the following steps are applied

$$\begin{aligned}\tilde{Z}_{k_t-1} \supseteq \{(\mathbf{x}_{k_t-1}, \mathbf{p}) \in \hat{Z}_{k_t-1} : \tilde{\mathbf{g}}(\mathbf{x}_{k_t-1}, \mathbf{u}_{k_t-1}, \mathbf{u}_{k_t}, \mathbf{p}, \mathbf{w}_{k_t-1}, \mathbf{v}_{k_t}) = \\ = \mathbf{y}_{k_t}, \mathbf{v}_{k_t} \in V, \mathbf{w}_{k_t-1} \in W\},\end{aligned} \tag{5.52}$$

$$\bar{Z}_{k_t} \supseteq \{(\mathbf{f}(\mathbf{x}_{k_t-1}, \mathbf{u}_{k_t-1}, \mathbf{p}, \mathbf{w}_{k_t-1}), \mathbf{p}) : (\mathbf{x}_{k_t-1}, \mathbf{p}) \in \tilde{Z}_{k_t-1},\ \mathbf{w}_{k_t-1} \in W\}, \tag{5.53}$$

$$\hat{Z}_{k_t} \supseteq \{(\mathbf{x}_{k_t}, \mathbf{p}) \in \bar{Z}_{k_t} : \mathbf{g}(\mathbf{x}_{k_t}, \mathbf{u}_{k_t}, \mathbf{p}, \mathbf{v}_{k_t}) = \mathbf{y}_{k_t},\ \mathbf{v}_{k_t} \in V\} \tag{5.54}$$

Equation (5.52) consists in an extra update steps to further refine the state-parameter set at time $k_t - 1$ using the measurement at time k_t. The obtained set is then propagated forward through the prediction step (5.53). Then, the set gets refined with the available measurement using the standard update step in (5.54). Note that, the exact computation of the operations described above is usually impossible in practice [Platzer and Clarke, 2007] and tight outer approximations are sought instead.

Remark 13 *The use of the extra update step* (5.52) *allows to partially alleviate the conservativeness introduced by the outer approximations. A further improvement could be achieved by accounting for the relationship between the measurement at time k_t and the prediction-update set at time instants prior to $k_t - 1$. However, this would result in an additional increase of computational complexity. Refer to [Rego et al., 2020] for a complexity analysis of constrained zonotopes based estimation.*

Joint state-parameter estimation with constrained zonotopes was addressed in [Rego et al., 2022] for the case of nonlinear systems with linear output equations. The approach can be easily extended to the case of nonlinear outputs by adapting Proposition 3 in [Rego et al., 2020] to the joint state-parameter case. In the following, the computation of the prediction step (5.53) is recalled and then how to address the update steps (5.52), (5.54) is presented.

Prediction step - Corollary 1 provides a method to compute an enclosure $\bar{Z}_{k_t}$ for the prediction step (5.53) assuming $\tilde{Z}_{k_t-1}$ and W are constrained zonotopes. $\square Z$ denotes the interval hull of Z while $\nabla_z^\top \boldsymbol{\kappa}$ indicates the Jacobian of $\boldsymbol{\kappa}$ with respect to $\mathbf{z}$.

Corollary 1 (Joint prediction [Rego et al., 2022]) *Let* $\mathbf{f}$ *in* (5.48a) *be of class* $\mathcal{C}^1$*. For* $k_t \geq 1$*, let:*

- $\mathbf{u}_{k-1} \in \mathbb{R}^{n_u}$,
- $\mathbf{w}_{k-1} \in W = \{\mathbf{G}_w, \mathbf{c}_w, \mathbf{A}_w, \mathbf{b}_w\}$,
- $\mathbf{z}_{k-1} = (\mathbf{x}_{k-1}, \mathbf{p}) \in \tilde{Z}_{k-1} = \{\tilde{\mathbf{G}}_{k-1}, \tilde{\mathbf{c}}_{k-1}, \tilde{\mathbf{A}}_{k-1}, \tilde{\mathbf{b}}_{k-1}\}$.

Choose any $\boldsymbol{\gamma}_z = (\boldsymbol{\gamma}_x, \boldsymbol{\gamma}_p) \in \square\tilde{Z}_{k_t-1}$*. Let* Z_w *be a constrained zonotope such that* $\mathbf{f}(\boldsymbol{\gamma}_x, \mathbf{u}_{k_t-1}, \boldsymbol{\gamma}_p, \mathbf{w}_{k_t-1}) \in Z_w$ *for all* $\mathbf{w}_{k_t-1} \in W$ *and let* $\mathbf{J}_z \in \mathbb{IR}^{n \times n+n_p}$ *satisfy* $\nabla_z^\top \mathbf{f}(\mathbf{x}_{k_t-1}, \mathbf{u}_{k_t-1}, \mathbf{p}, \mathbf{w}_{k_t-1}) \in \mathbf{J}_z$ *for all* $(\mathbf{x}_{k_t-1}, \mathbf{p}) \in \square\tilde{Z}_{k_t-1}$, $\mathbf{w}_{k_t-1} \in W$*. If* $\{\underline{\tilde{\mathbf{G}}}, \underline{\tilde{\mathbf{c}}}\}$ *is a zonotope with* $\underline{n}_g$ *generators satisfying* $\tilde{Z}_{k_t-1} - \boldsymbol{\gamma}_z \subseteq \{\underline{\tilde{\mathbf{G}}}, \underline{\tilde{\mathbf{c}}}\}$*, then* $(\mathbf{x}_{k_t}, \mathbf{p}) \in \bar{Z}_{k_t}$*, with*

$$\bar{Z}_{k_t} = \begin{bmatrix}\mathbf{H}\\ \mathbf{E}\end{bmatrix} \tilde{Z}_{k_t-1} \oplus \begin{bmatrix}\mathbf{H}\\ \mathbf{0}\end{bmatrix} (-\boldsymbol{\gamma}_z) \oplus \begin{bmatrix}\tilde{\mathbf{P}}\\ \mathbf{0}\end{bmatrix} B_\infty^n \oplus \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix} Z_w, \tag{5.55}$$

where $\mathbf{E} \triangleq [\mathbf{0}_{n_p \times n} \ \ \mathbb{I}_{n_p}]$, $\mathbf{H} \triangleq mid(\mathbf{J}_z)$, $\tilde{\mathbf{P}} \in \mathbb{R}^{n \times n}$ *is diagonal with* $\tilde{P}_{ii} = rad(m_i) + \sum_{j=1}^{\underline{n}_g} \sum_{\ell=1}^{n+n_p} rad(J_{z,i\ell}) |\underline{\tilde{G}}_{\ell j}|$*, and* $\mathbf{m} \triangleq (\mathbf{J}_z - mid(\mathbf{J}_z))\underline{\tilde{\mathbf{c}}} \in \mathbb{IR}^n$.

Remark 14 *Corollary 1 makes implicit use of the constrained zonotope inclusion described in Theorem 1. Note that* $\mathbf{J}_z$ *can be obtained by evaluating* $\nabla_z^\top \mathbf{f}$ *over the interval hull of the involved sets using interval arithmetic. Moreover, the constrained zonotope* Z_w *can be derived as* $Z_w = \mathbf{f}(\boldsymbol{\gamma}_x, \mathbf{u}_{k-1}, \boldsymbol{\gamma}_p, \boldsymbol{\gamma}_w) \oplus \triangleleft(\mathbf{J}_w, W - \boldsymbol{\gamma}_w) \supseteq \mathbf{f}(\boldsymbol{\gamma}_x, \mathbf{u}, \boldsymbol{\gamma}_p, W)$*, for a chosen* $\boldsymbol{\gamma}_w \in \square W$*, and interval matrix* $\mathbf{J}_w \supseteq \nabla_w^\top \mathbf{f}(\boldsymbol{\gamma}_x, \mathbf{u}, \boldsymbol{\gamma}_p, \square W)$.

With respect to other set-representations, such as intervals, the constrained zonotope in equation (5.55) allows to account for the coupling between state and parameter values. This feature, present also in the update steps, allows for a more accurate estimation.

Update step - The following Proposition provides a constrained zonotope enclosure $\hat{Z}_k$ for the update step (5.54) assuming $\bar{Z}_k$, V are constrained zonotopes. A similar result can be obtained for (5.52) by replacing function $\mathbf{g}$ with $\tilde{\mathbf{g}}$.

Proposition 3 (Joint update) *Let* $\mathbf{g}$ *in* (5.48b) *be of class* $\mathcal{C}^1$. *For* $k_t \geq 0$, *let:*

- $\mathbf{u}_{k_t} \in \mathbb{R}^{n_u}$,
- $\mathbf{z}_{k_t} = (\mathbf{x}_{k_t}, \mathbf{p}) \in \bar{Z}_{k_t} = \left\{\bar{\mathbf{G}}_{k_t}, \bar{\mathbf{c}}_{k_t}, \bar{\mathbf{A}}_{k_t}, \bar{\mathbf{b}}_{k_t}\right\}$,
- $\mathbf{v} \in V = \{\mathbf{G}_v, \mathbf{c}_v, \mathbf{A}_v, \mathbf{b}_v\}$.

Assume measurement $\mathbf{y}_{k_t} \in \mathbb{R}^{n_y}$ *is consistent with the predicted set and the output dynamics, i.e.* $\exists(\mathbf{x}_{k_t}, \mathbf{p}, \mathbf{v}_{k_t}) \in \bar{Z}_{k_t} \times V$ *such that* $\mathbf{y}_{k_t} = \mathbf{g}(\mathbf{x}_{k_t}, \mathbf{u}_{k_t}, \mathbf{p}, \mathbf{v}_{k_t})$. *Choose any* $\boldsymbol{\gamma}_z \in \square\bar{Z}_{k_t}$ *and any* $\tilde{\mathbf{J}} \in \mathbb{R}^{n_y \times n+n_p}$. *If* $Z_v = \{\mathbf{G}_v, \mathbf{c}_v, \mathbf{A}_v, \mathbf{b}_v\}$ *is a constrained zonotope such that* $-\mathbf{g}(\boldsymbol{\gamma}_x, \mathbf{u}, \boldsymbol{\gamma}_p, \mathbf{v}_{k_t}) \in Z_v$, *for all* $\mathbf{v}_{k_t} \in V$ *and* $\mathbf{J} \in \mathbb{R}^{n_y \times n+n_p}$ *is an interval matrix satisfying* $\nabla_z^\top \mathbf{g}(\mathbf{x}_{k_t}, \mathbf{u}_{k_t}, \mathbf{p}, \mathbf{v}_{k_t}) \in \mathbf{J}$, *for all* $(\mathbf{x}_{k_t}, \mathbf{p}) \in \square\bar{Z}_k$, $\mathbf{v}_k \in V$, *then* $\left\{(\mathbf{x}_{k_t}, \mathbf{p}) \in \bar{Z}_{k_t} : \mathbf{g}(\mathbf{x}_{k_t}, \mathbf{u}_{k_t}, \mathbf{p}, \mathbf{v}_{k_t}) = \mathbf{y}_{k_t}, \mathbf{v}_{k_t} \in V\right\} \subseteq \hat{Z}_{k_t} = \bar{Z}_{k_t} \cap_{\mathbf{C}} Y_{k_t}$, *where* $\mathbf{C} = \tilde{\mathbf{J}}$ *and*

$$Y_{k_t} = (\mathbf{y}_{k_t} + \tilde{\mathbf{J}}\boldsymbol{\gamma}_\mathbf{z}) \oplus Z_v \oplus \triangleleft(\mathbf{J} - \tilde{\mathbf{J}}, \bar{Z}_{k_t} - \boldsymbol{\gamma}_z),$$

Proof. *This result can be easily obtained by replacing* $\mathbf{x}_{k_t}$ *with* $(\mathbf{x}_{k_t}, \mathbf{p})$ *in the proof of Proposition 3 in [Rego et al., 2020].* ■

Remark 15 *The constrained zonotope* Z_v, *in Theorem 3, can be obtained as* $Z_v = -\mathbf{g}(\boldsymbol{\gamma}_x, \mathbf{u}, \boldsymbol{\gamma}_p, \boldsymbol{\gamma}_v) \oplus \triangleleft(-\mathbf{J}_v, V - \boldsymbol{\gamma}_v) \supseteq -\mathbf{g}(\boldsymbol{\gamma}_x, \mathbf{u}, \boldsymbol{\gamma}_p, V)$, *for a chosen* $(\boldsymbol{\gamma}_x, bm\gamma_p) \in \square X \times \square P$, *and interval matrix* $\mathbf{J}_v \supseteq \nabla^\top \mathbf{g}(\boldsymbol{\gamma}_x, \mathbf{u}, \boldsymbol{\gamma}_p, \square V)$.

Remark 16 *If the constrained zonotopes* $(\bar{Z}_k, V)$ *have* $(\bar{n}_g, n_{g_v})$ *generators, and* $(\bar{n}_c, n_{c_v})$ *constraints, respectively, then the enclosure obtained by Theorem 3 has* $\bar{n}_g + 2n_y + n_{g_v}$ *generators and* $n_c + n_{c_y} + n_{c_v}$ *constraints.*

Note that in the set-based joint and state estimation proposed here, the approximation given by $\boldsymbol{\mathcal{C}2}$) for the point $\boldsymbol{\gamma}_z$ presented in [Rego et al., 2020] is applied.

Remark 17 *Note that the prediction and the update steps lead to an increase of complexity of the resulting sets in terms of generators/constraints. Because of this, constrained zonotopes order reductions [Scott et al., 2016] are used at each iteration of the estimation technique in order to keep the complexity limited.*

The Li-ion cell model used in this part of the work is the Thévenin ECM,

presented in Section 2.4.1, whose equations are reported here for clarity

$$\begin{cases} \dfrac{dV_{th}(t)}{dt} = \dfrac{-1}{R_{th}C_{th}}V_{th}(t) + \dfrac{1}{C_{th}}I_{cell}(t), \\ \dfrac{d\mathrm{SOC}(t)}{dt} = -\dfrac{I_{cell}(t)}{3600C_{cell}}, \\ \dfrac{dT_c(t)}{dt} = \dfrac{T_s(t) - T_c(t)}{R_cC_c} + \dfrac{|I_{cell}(t)|\,|\left(V_{cell}(t) - V_{EMF}(\mathrm{SOC}(t))\right)|}{C_c}, \\ \dfrac{dT_s(t)}{dt} = \dfrac{T_f(t) - T_s(t)}{R_uC_c} + \dfrac{T_s(t) - T_c(t)}{R_cC_s}, \\ V_{cell}(t) = V_{EMF}(\mathrm{SOC}(t)) + V_{th}(t) + R_{int}I_{cell}(t). \end{cases} \tag{5.56}$$

The EMF potential is expressed here as a sixth degree polynomial expression of SOC

$$\begin{aligned} V_{EMF}(t) =& 3.592 + 2.780\,\mathrm{SOC}(t) - 9.495\,\mathrm{SOC}^2(t) + 21.17\,\mathrm{SOC}^3(t) + \\ & - 28.76\,\mathrm{SOC}^4(t) + 21.89\,\mathrm{SOC}^5(t) - 6.790\,\mathrm{SOC}^6(t), \end{aligned} \tag{5.57}$$

whose coefficients are obtained by interpolating the EMF curve reported in [Lin et al., 2014b].

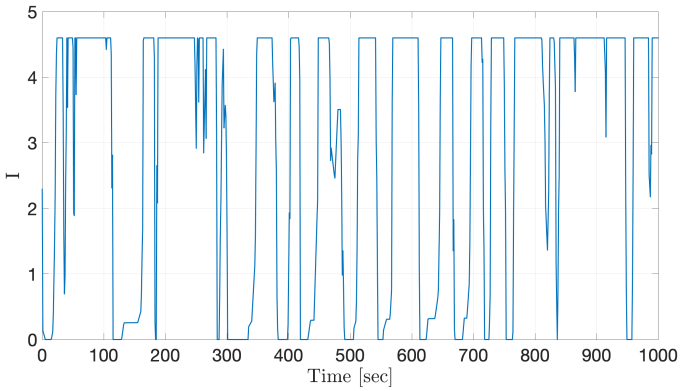


Figure 5.17: Input current profile I applied to the cell.

In order to use the presented set-based estimation approach, the electro-thermal model (5.56) is first discretized using the Forward Euler's method. Denoting with $\mathbf{x}$ the system states, $\mathbf{u}$ the inputs, $\mathbf{p}$ the parameters, and $\mathbf{v}$ the measurement noise, the discretized version of (5.56) can be compactly written as

$$\begin{cases} \mathbf{x}_{k_t} = \mathbf{x}_{k_t-1} + t_s\mathbf{f}(\mathbf{x}_{k_t-1}, \mathbf{u}_{k_t-1}, \mathbf{p}), & k_t \geq 1 \\ \mathbf{y}_{k_t} = \mathbf{g}(\mathbf{x}_{k_t}, \mathbf{u}_{k_t}, \mathbf{p}, \mathbf{v}_{\mathbf{k_t}}), & k_t \geq 0 \end{cases} \tag{5.58}$$

where $\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}) : \mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^n$ and $\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{p}, \mathbf{v}) : \mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_y}$ indicate respectively the continuous state and output dynamics, while t_s is the sampling time and $\mathbf{x}_0$ is the initial condition. The objective is to

estimate the set of states consistent with the model dynamics and the obtained measurements under the assumption of unknown but bounded uncertainties in terms of initial condition, measurement noise and parameters. Differently from standard set-based state estimation, also the uncertain parameter set is refined using the obtained measurements. The discretized system is equal to

$$\begin{cases} x_1(k_t) &= x_1(k_t-1) + t_s\dfrac{u(k_t-1)}{C_{cell}}, \\ x_2(k_t) &= x_2(k_t-1) + t_s(-x_2(k_t-1)p_{R_1C_1} + u(k_t-1)p_{C_1}), \\ x_3(k_t) &= x_3(k_t-1) + t_s((x_4(k_t-1) - x_3(k_t-1))p_{R_cC_c} + \\ &\quad +|u(k_t-1)||x_2(k_t-1) + R_{int}u(k_t-1)|\, p_{C_c}), \\ x_4(k_t) &= x_4(k_t-1) + t_s((T_f - x_4(k_t-1))p_{R_uC_c} + \\ &\quad +(x_4(k_t-1) - x_3(k_t-1))p_{R_cC_s}), \\ y_1(k_t) &= V_{EMF}(x_1(k_t)) + R_{int}u(k_t) + x_2(k_t) + v_1(k_t), \\ y_2(k_t) &= x_4(k_t) + v_2(k_t), \end{cases} \tag{5.59}$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^\top = \begin{bmatrix} \text{SOC} & V_1 & T_c & T_s \end{bmatrix}^\top \tag{5.60}$$

denotes the state vector, while u is the input current I. In order to perform the extra update step (5.52), the output quantities are expressed in accordance with (5.49) as

$$\begin{cases} y_1(k_t) &= V_{EMF}\left(x_1(k_t-1) + t_s\dfrac{u(k_t-1)}{C_{cell}}\right) + R_{int}u(k_t) + x_2(k_t-1) + \\ &\quad +t_s(-x_2(k_t-1)p_{R_1C_1} + u(k_t-1)p_{C_1}) + v_1(k_t), \\ y_2(k_t) &= x_4(k_t-1) + t_s((T_f - x_4(k_t-1))p_{R_uC_c} + \\ &\quad +(x_4(k_t-1) - x_3(k_t-1))p_{R_cC_s}) + v_2(k_t). \end{cases} \tag{5.61}$$

The measurable outputs to be affected by an unknown, but bounded, measurement noise $\mathbf{v}_k \in V = \{\text{diag}\,(0.001, 0.1)\,, \mathbf{0}2 \times 1\}$. The parametric uncertainty vector is equal to $\mathbf{p} = \begin{bmatrix} R_{int} & p_{R_1C_1} & p_{C_1} & p_{R_cC_c} & p_{C_c} & p_{R_uC_c} & p_{R_cC_s} \end{bmatrix}^\top \in P$, where the initial parametric set $P = \{\mathbf{G}_p, \mathbf{c}_p\}$, with

$$\begin{aligned} \mathbf{c}_p =& [0.0100\ \ 0.0238\ \ 0.0004762\ \ 0.0082\ \ 0.0160\ \ 0.0722\ \ 0.1145]^\top, \\ \mathbf{G}_p =& \text{diag}(5{\cdot}10^{-4}, 0.0012, 2.3810{\cdot}10^{-5}, 04.111{\cdot}10^{-4}, \\ & 7.9745{\cdot}10^{-4}, 0.00361, 0.00573). \end{aligned}$$

Note that, with respect to the general formulation of the problem given previously, it is assumed that in the dynamics there are parameter uncertainties only (no other process uncertainty is here considered). The battery capacity $C_{cell} = 2.3$Ah is assumed to be known. The initial state set X_0 is also a zonotope described by $\{\text{diag}(0.01, 0.001, 0.5, 0.5), [0.25\ \ 0\ \ 293.15\ \ 293.15]^\top\}$.

Measurement data have been collected by simulating (5.59) with

$$\mathbf{x}_0 = \begin{bmatrix} 0.25 & 0 & 293.15 & 293.15 \end{bmatrix} \in X_0,$$
$$\mathbf{p} = [0.0100 \;\; 0.0238 \;\; 0.0004762 \;\; 0.0082 \;\; 0.0160 \;\; 0.0722 \;\; 0.1145]^\top \in P.$$

Measurement disturbances are generated according to a uniform random distribution. The known input profile is depicted in Figure 5.17.

Figure 5.18 compares the enclosures of the two electrical state components (SOC and V_1) obtained using constrained zonotopes (green) and intervals (pink) every 100s starting at $k_t = 0$. Note that while the interval method in [Locatelli et al., 2021] allows multiple iterations of the algorithm at each time step, only one iteration is performed here. This is due to the fact that multiple iterations did not result in additional refinement of the interval enclosures.

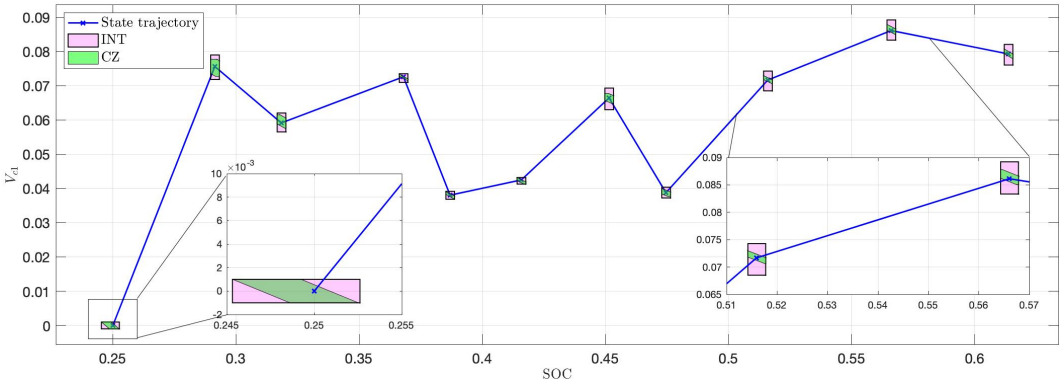


Figure 5.18: Enclosures of SOC and V_1 components obtained using interval (pink) and constrained zonotope (green) joint state estimation.

Figure 5.19 shows the comparison of the enclosures of the two thermal states (T_c and T_s) obtained with constrained zonotopes and intervals at $k_t \in \{0, 100, 200, 400, 900, 1000\}$. Although constrained zonotopes have a small edge in electrical states, this looks to be far more relevant in thermal states. The ability of to capture the dependence between states and parameters allows for this outcome.

In order to further demonstrate the effectiveness of constrained zonotopes compared to intervals, the radii[3] of the respective enclosures are shown in Figure 5.20. In particular, the first figure represents the radii of the enclosures of the state variables while; the second Figure shows the radii with respect to the parametric component. It is clear how the results highlight the advantages of the proposed method over intervals. The latter provide worse estimates of both the states and the parameters. In particular, intervals behave poorly in the estimation of the parameters, with the trend of the radii remaining almost constant along the parametric components. On the other side, constrained

[3]The radii are computed as the sum of the absolute value of all the edges of the interval hull (1-norm).

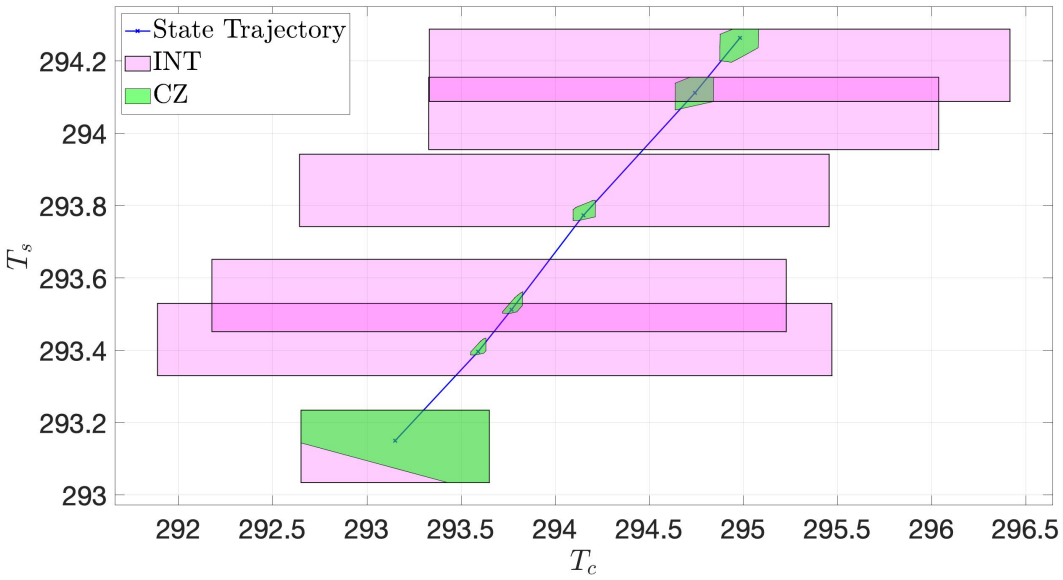


Figure 5.19: Enclosures of the T_c and T_s components obtained using interval (pink) and constrained zonotopes (green) joint state estimation.

zonotopes provide a continuous reduction of the parametric uncertainty which results being beneficial in the overall state estimation.

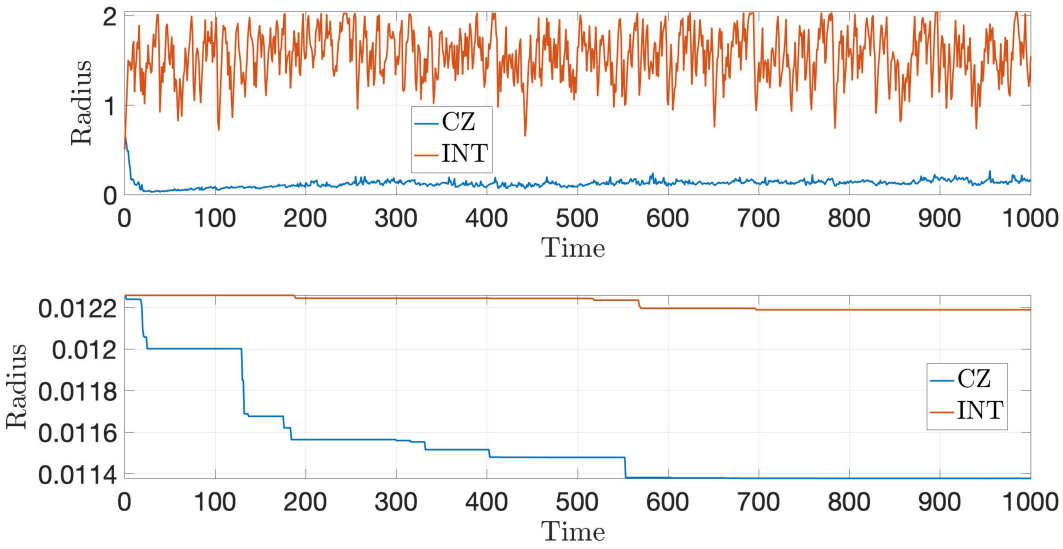


Figure 5.20: Radii of the enclosures of the state components (top) and of the parameter components (bottom) obtained using intervals (red) and CZs (blue).

Chapter 6

Conclusions

In this Chapter the use of set-based methods applied to fault detection was studied. After having defined the basis for set-based estimation, a fault detection algorithm was applied to a lithium-ion battery. The performance of constrained zonotopes and intervals was compared, with a considerable advantage obtained by the first set representation. Subsequently, the same fault detection strategy was applied to a battery pack with cells connected in series. Thermal exchange between the cells was included in the model. This time, the results obtained with constrained zonotopes were compared with zonotopes. To conclude, the state estimation algorithm was extended to include also parameter estimation. This allows to further refine the results and obtain smaller enclosures than before. Results were once again compared between constrained zonotopes and intervals. The most evident advantage was noted in the estimate of thermal states, as constrained zonotopes are able to capture mutual dependencies between the states.

Bibliography

[Alamo et al., 2008] Alamo, T., Bravo, J. M., Redondo, M. J., and Camacho, E. F. (2008). A set-membership state estimation algorithm based on dc programming. *Automatica*, 44(1):216–224.

[Alavi et al., 2013] Alavi, S. M., Samadi, M. F., and Saif, M. (2013). Plating mechanism detection in lithium-ion batteries, by using a particle-filtering based estimation technique. In *2013 American Control Conference*, pages 4356–4361. IEEE.

[Anderson, 1965] Anderson, D. G. (1965). Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560.

[Antón et al., 2013] Antón, J. Á., Nieto, P. G., de Cos Juez, F., Lasheras, F. S., Vega, M. G., and Gutiérrez, M. R. (2013). Battery state-of-charge estimator using the svm technique. *Applied Mathematical Modelling*, 37(9):6244–6253.

[Anton et al., 2013] Anton, J. C. A., Nieto, P. J. G., Viejo, C. B., and Vilán, J. A. V. (2013). Support vector machines used to estimate the battery state of charge. *IEEE Transactions on power electronics*, 28(12):5919–5926.

[Bandhauer et al., 2011] Bandhauer, T. M., Garimella, S., and Fuller, T. F. (2011). A critical review of thermal issues in lithium-ion batteries. *Journal of the Electrochemical Society*, 158(3).

[Baumhöfer et al., 2014] Baumhöfer, T., Brühl, M., Rothgang, S., and Sauer, D. U. (2014). Production caused variation in capacity aging trend and correlation to initial cell performance. *Journal of Power Sources*, 247:332–338.

[Bizeray et al., 2013] Bizeray, A., Duncan, S., and Howey, D. (2013). Advanced battery management systems using fast electrochemical modelling. *IET Conference Proceedings*, pages 2.2–2.2(1).

[Bruen and Marco, 2016] Bruen, T. and Marco, J. (2016). Modelling and experimental evaluation of parallel connected lithium ion cells for an electric vehicle battery system. *Journal of Power Sources*, 310:91–101.

[Cai and White, 2012] Cai, L. and White, R. E. (2012). Lithium ion cell modeling using orthogonal collocation on finite elements. *Journal of Power Sources*, 217:248–255.

[Cao et al., 2019] Cao, W., Wang, S.-L., Fernandez, C., Zou, C.-Y., Yu, C.-M., and Li, X.-X. (2019). A novel adaptive state of charge estimation method of full life cycling lithium-ion batteries based on the multiple parameter optimization. *Energy Science & Engineering*, 7(5):1544–1556.

[Chang, 1989] Chang, F.-Y. (1989). The generalized method of characteristics for waveform relaxation analysis of lossy coupled transmission lines. *IEEE Transactions on Microwave Theory and Techniques*, 37(12):2028–2038.

[Chang, 1990] Chang, F.-Y. (1990). Waveform relaxation analysis of rlcg transmission lines. *IEEE transactions on circuits and systems*, 37(11):1394–1415.

[Chaouqui et al., 2018] Chaouqui, F., Ciaramella, G., Gander, M. J., and Vanzan, T. (2018). On the scalability of classical one-level domain-decomposition methods. *Vietnam Journal of Mathematics*, 46(4):1053–1088.

[Chemali et al., 2017] Chemali, E., Kollmeyer, P. J., Preindl, M., Ahmed, R., and Emadi, A. (2017). Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries. *IEEE Transactions on Industrial Electronics*, 65(8):6730–6739.

[Chun et al., 2020] Chun, H., Kim, J., Yu, J., and Han, S. (2020). Real-time parameter estimation of an electrochemical lithium-ion battery model using a long short-term memory network. *IEEE Access*, 8:81789–81799.

[Ciaramella and Fabrini, 2021] Ciaramella, G. and Fabrini, G. (2021). Multilevel techniques for the solution of hjb minimum-time control problems. *Journal of Systems Science and Complexity*, 34(6):2069–2091.

[Ciaramella and Gander, 2017] Ciaramella, G. and Gander, M. J. (2017). Analysis of the parallel schwarz method for growing chains of fixed-sized subdomains: Part i. *SIAM Journal on Numerical Analysis*, 55(3):1330–1356.

[Ciaramella et al., 2019] Ciaramella, G., Gander, M. J., Halpern, L., and Salomon, J. (2019). Methods of reflections: relations with schwarz methods and classical stationary iterations, scalability and preconditioning. *The SMAI journal of computational mathematics*, 5:161–193.

[Couto and Kinnaert, 2018] Couto, L. D. and Kinnaert, M. (2018). Internal and sensor fault detection and isolation for li-ion batteries. *IFAC-PapersOnLine*, 51(24):1431–1438.

[Crow and Ilić, 1994] Crow, M. L. and Ilić, M. (1994). The waveform relaxation method for systems of differential/algebraic equations. *Mathematical and computer modelling*, 19(12):67–84.

[Delacourt et al., 2011] Delacourt, C., Ati, M., and Tarascon, J. (2011). Measurement of lithium diffusion coefficient in li y feso4f. *Journal of the Electrochemical Society*, 158(6):A741.

[Dey et al., 2015] Dey, S., Biron, Z. A., Tatipamula, S., Das, N., Mohon, S., Ayalew, B., and Pisu, P. (2015). On-board thermal fault diagnosis of lithium-ion batteries for hybrid electric vehicle application. *IFAC-PapersOnLine*, 48(15):389–394.

[Dey et al., 2016] Dey, S., Mohon, S., Pisu, P., and Ayalew, B. (2016). Sensor fault detection, isolation, and estimation in lithium-ion batteries. *IEEE Transactions on Control Systems Technology*, 24(6):2141–2149.

[Dey et al., 2017] Dey, S., Perez, H. E., and Moura, S. J. (2017). Model-based battery thermal fault diagnostics: Algorithms, analysis, and experiments. *IEEE Transactions on Control Systems Technology*, 27(2):576–587.

[Doyle et al., 1993] Doyle, M., Fuller, T. F., and Newman, J. (1993). Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *Journal of the Electrochemical society*, 140(6):1526.

[Dubal et al., 2016] Dubal, D. P., Wu, Y., and Holze, R. (2016). Supercapacitors: from the leyden jar to electric busses. *ChemTexts*, 2(3):1–19.

[Ecker et al., 2015a] Ecker, M., Käbitz, S., Laresgoiti, I., and Sauer, D. U. (2015a). Parameterization of a physico-chemical model of a lithium-ion battery: Ii. model validation. *Journal of The Electrochemical Society*, 162(9):A1849.

[Ecker et al., 2015b] Ecker, M., Käbitz, S., Laresgoiti, I., and Sauer, D. U. (2015b). Parameterization of a physico-chemical model of a lithium-ion battery: Ii. model validation. *Journal of The Electrochemical Society*, 162(9):A1849.

[Ecker et al., 2015c] Ecker, M., Tran, T. K. D., Dechent, P., Käbitz, S., Warnecke, A., and Sauer, D. U. (2015c). Parameterization of a physico-chemical model of a lithium-ion battery: I. determination of parameters. *Journal of The Electrochemical Society*, 162(9):A1836.

[Eichi and Chow, 2012] Eichi, H. R. and Chow, M.-Y. (2012). Modeling and analysis of battery hysteresis effects. In *2012 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 4479–4486. IEEE.

[Feinauer et al., 2019] Feinauer, J., Hein, S., Rave, S., Schmidt, S., Westhoff, D., Zausch, J., Iliev, O., Latz, A., Ohlberger, M., and Schmidt, V. (2019). Multibat: Unified workflow for fast electrochemical 3d simulations of lithium-ion cells combining virtual stochastic microstructures, electrochemical degradation models and model order reduction. *Journal of Computational Science*, 31:172–184.

[Forman et al., 2010] Forman, J. C., Bashash, S., Stein, J. L., and Fathy, H. K. (2010). Reduction of an electrochemistry-based li-ion battery model via quasi-linearization and pade approximation. *Journal of the Electrochemical Society*, 158(2):A93.

[Gander, 1999] Gander, M. J. (1999). A waveform relaxation algorithm with overlapping splitting for reaction diffusion equations. *Numerical linear algebra with applications*, 6(2):125–145.

[Gander, 2008] Gander, M. J. (2008). Schwarz methods over the course of time. *Electronic transactions on numerical analysis*, 31:228–255.

[Gander et al., 1999] Gander, M. J., Halpern, L., and Nataf, F. (1999). Optimal convergence for overlapping and non-overlapping schwarz waveform relaxation. In *11th international conference on domain decomposition methods*, pages 27–36.

[Gander and Ruehli, 2004] Gander, M. J. and Ruehli, A. E. (2004). Optimized waveform relaxation methods for rc type circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(4):755–768.

[Gander and Ruehli, 2010] Gander, M. J. and Ruehli, A. E. (2010). Optimized waveform relaxation solution of electromagnetic and circuit problems. In *19th Topical Meeting on Electrical Performance of Electronic Packaging and Systems*, pages 65–68. IEEE.

[Gander et al., 2014] Gander, W., Gander, M. J., and Kwok, F. (2014). *Scientific computing-An introduction using Maple and MATLAB*, volume 11. Springer Science & Business.

[Georén and Lindbergh, 2004] Georén, P. and Lindbergh, G. (2004). Characterisation and modelling of the transport properties in lithium battery gel electrolytes: Part i. the binary electrolyte pc/liclo4. *Electrochimica acta*, 49(21):3497–3505.

[Giorgi and Guerraggio, 1994] Giorgi, G. and Guerraggio, A. (1994). First order generalized optimality conditions for programming problems with a set constraint. In *Generalized Convexity*, pages 171–185. Springer.

[Gopalakrishnan and Offer, 2021] Gopalakrishnan, K. and Offer, G. J. (2021). A composite single particle lithium-ion battery model through system identification. *IEEE Transactions on Control Systems Technology*, 30(1):1–13.

[Han et al., 2021] Han, S., Tang, Y., and Rahimian, S. K. (2021). A numerically efficient method of solving the full-order pseudo-2-dimensional (p2d) li-ion cell model. *Journal of Power Sources*, 490:229571.

[Hannan et al., 2018] Hannan, M. A., Lipu, M. S. H., Hussain, A., Saad, M. H., and Ayob, A. (2018). Neural network approach for estimating state of charge of lithium-ion battery using backtracking search algorithm. *Ieee Access*, 6:10069–10079.

[He et al., 2011] He, H., Xiong, R., and Fan, J. (2011). Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. *energies*, 4(4):582–598.

[Hemavathi, 2020] Hemavathi, S. (2020). Overview of cell balancing methods for li-ion battery technology. *Energy storage*, 203:1–12.

[How et al., 2019] How, D. N., Hannan, M., Lipu, M. H., and Ker, P. J. (2019). State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review. *Ieee Access*, 7:136116–136136.

[Hu et al., 2015] Hu, X., Perez, H. E., and Moura, S. J. (2015). Battery charge control with an electro-thermal-aging coupling. In *Dynamic Systems and Control Conference*, volume 57243, page V001T13A002. American Society of Mechanical Engineers.

[Hu et al., 2020] Hu, X., Zhang, K., Liu, K., Lin, X., Dey, S., and Onori, S. (2020). Advanced fault diagnosis for lithium-ion battery systems: a review of fault mechanisms, fault features, and diagnosis procedures. *IEEE Industrial Electronics Magazine*, 14(3):65–91.

[Jackey et al., 2013] Jackey, R., Saginaw, M., Sanghvi, P., Gazzarri, J., Huria, T., and Ceraolo, M. (2013). Battery model parameter estimation using a layered technique: an example using a lithium iron phosphate cell. *SAE Technical Paper*, 2:1–14.

[Jain et al., 2020] Jain, P., Raghav, S., Dhillon, A., and Kumar, D. (2020). History and development of zinc batteries. *Zinc Batteries: Basics, Developments, and Applications*, pages 167–186.

[Jaulin et al., 2001] Jaulin, L., Braems, I., Kieffer, M., and Walter, E. (2001). Nonlinear state estimation using forward-backward propagation of intervals in an algorithm. In *Scientific computing, validated numerics, interval methods*, pages 191–201. Springer.

[Jiang et al., 2021] Jiang, C., Wang, S., Wu, B., Fernandez, C., Xiong, X., and Coffie-Ken, J. (2021). A state-of-charge estimation method of the power lithium-ion battery in complex conditions based on adaptive square root extended kalman filter. *Energy*, 219:119603.

[Kemper et al., 2015] Kemper, P., Li, S. E., and Kum, D. (2015). Simplification of pseudo two dimensional battery model using dynamic profile of lithium concentration. *Journal of Power Sources*, 286:510–525.

[Khalik et al., 2021a] Khalik, Z., Donkers, M., and Bergveld, H. J. (2021a). Model simplifications and their impact on computational complexity for an electrochemistry-based battery modeling toolbox. *Journal of Power Sources*, 488:229427.

[Khalik et al., 2021b] Khalik, Z., Donkers, M., Sturm, J., and Bergveld, H. J. (2021b). Parameter estimation of the doyle–fuller–newman model for lithium-ion batteries by parameter normalization, grouping, and sensitivity analysis. *Journal of Power Sources*, 499:229901.

[Kim et al., 2012] Kim, J., Seo, G.-S., Chun, C., Cho, B.-H., and Lee, S. (2012). Ocv hysteresis effect-based soc estimation in extended kalman filter algorithm for a lifepo 4/c cell. In *2012 IEEE International Electric Vehicle Conference*, pages 1–5. IEEE.

[Kim et al., 2019] Kim, M., Chun, H., Kim, J., Kim, K., Yu, J., Kim, T., and Han, S. (2019). Data-efficient parameter identification of electrochemical lithium-ion battery model using deep bayesian harmony search. *Applied Energy*, 254:113644.

[Korotkin et al., 2021] Korotkin, I., Sahu, S., O'Kane, S. E., Richardson, G., and Foster, J. M. (2021). Dandeliion v1: An extremely fast solver for the newman model of lithium-ion battery (dis) charge. *Journal of The Electrochemical Society*, 168(6):060544.

[Kurzweil, 2010] Kurzweil, P. (2010). Gaston planté and his invention of the lead–acid battery—the genesis of the first practical rechargeable battery. *Journal of Power Sources*, 195(14):4424–4434.

[Lai et al., 2021] Lai, Q., Ahn, H. J., Kim, Y., Kim, Y. N., and Lin, X. (2021). New data optimization framework for parameter estimation under uncertainties with application to lithium-ion battery. *Applied Energy*, 295:117034.

[Lashway and Mohammed, 2016] Lashway, C. R. and Mohammed, O. A. (2016). Adaptive battery management and parameter estimation through physics-based modeling and experimental verification. *IEEE Transactions on Transportation Electrification*, 2(4):454–464.

[Latz and Zausch, 2011] Latz, A. and Zausch, J. (2011). Thermodynamic consistent transport theory of li-ion batteries. *Journal of Power Sources*, 196(6):3296–3302.

[Lelarasmee, 1982] Lelarasmee, E. (1982). *The waveform relaxation method for time domain analysis of large scale integrated circuits: Theory and applications*. Electronics Research Laboratory, College of Engineering, University of ....

[Lelarasmee et al., 1982] Lelarasmee, E., Ruehli, A. E., and Sangiovanni-Vincentelli, A. L. (1982). The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE transactions on computer-aided design of integrated circuits and systems*, 1(3):131–145.

[Less et al., 2012] Less, G. B., Seo, J. H., Han, S., Sastry, A. M., Zausch, J., Latz, A., Schmidt, S., Wieser, C., Kehrwald, D., and Fell, S. (2012). Micro-scale modeling of li-ion batteries: parameterization and validation. *Journal of The Electrochemical Society*, 159(6):A697.

[Li and Wang, 2018] Li, X. and Wang, Z. (2018). A novel fault diagnosis method for lithium-ion battery packs of electric vehicles. *Measurement*, 116:402–411.

[Lidbeck and Syed, 2017] Lidbeck, A. and Syed, K. (2017). Experimental characterization of li-ion battery cells for thermal management in heavy duty hybrid applications. Master's thesis, Chalmers University.

[Lin et al., 2014a] Lin, X., Perez, H. E., Mohan, S., Siegel, J. B., Stefanopoulou, A. G., Ding, Y., and Castanier, M. P. (2014a). A lumped-parameter electro-thermal model for cylindrical batteries. *Journal of Power Sources*, 257:1–11.

[Lin et al., 2014b] Lin, X., Perez, H. E., Mohan, S., Siegel, J. B., Stefanopoulou, A. G., Ding, Y., and Castanier, M. P. (2014b). A lumped-parameter electro-thermal model for cylindrical batteries. *Journal of Power Sources*, 257:1–11.

[Liu et al., 2018] Liu, P., Sun, Z., Wang, Z., and Zhang, J. (2018). Entropy-based voltage fault diagnosis of battery systems for electric vehicles. *Energies*, 11(1):136.

[Liu et al., 2014] Liu, Z., Ahmed, Q., Rizzoni, G., and He, H. (2014). Fault detection and isolation for lithium-ion battery system using structural analysis and sequential residual generation. In *Dynamic Systems and Control Conference*, volume 46193. American Society of Mechanical Engineers.

[Locatelli et al., 2021] Locatelli, D., Rego, B., Raffo, G., and Raimondo, D. (2021). Interval state estimation based on constraint propagation for a

lithium-ion cell using an equivalent circuit model. *IFAC-PapersOnLine*, 54(3):602–608.

[Maciejewski et al., 2017] Maciejewski, M., Garcia, I. C., Schöps, S., Auchmann, B., Bortot, L., Prioli, M., and Verweij, A. (2017). Application of the waveform relaxation technique to the co-simulation of power converter controller and electrical circuit models. In *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 837–842. IEEE.

[Maiser, 2014] Maiser, E. (2014). Battery packaging-technology review. In *AIP Conference Proceedings*, volume 1597, pages 204–218. American Institute of Physics.

[Marcicki et al., 2010] Marcicki, J., Onori, S., and Rizzoni, G. (2010). Nonlinear fault detection and isolation for a lithium-ion battery management system. In *Dynamic Systems and Control Conference*, volume 44175, pages 607–614.

[Mizushima et al., 1980] Mizushima, K., Jones, P., Wiseman, P., and Goodenough, J. B. (1980). Lixcoo2 (0¡ x¡-1): A new cathode material for batteries of high energy density. *Materials Research Bulletin*, 15(6):783–789.

[Moayedi et al., 2013] Moayedi, S., Cingöz, F., and Davoudi, A. (2013). Accelerated simulation of high-fidelity models of supercapacitors using waveform relaxation techniques. *IEEE Transactions on Power Electronics*, 28(11):4903–4909.

[Moore et al., 2009] Moore, R. E., Kearfott, R. B., and Cloud, M. J. (2009). *Introduction to Interval Analysis*. SIAM, Philadelphia, PA, USA.

[Moura, 2015] Moura, S. J. (2015). Estimation and control of battery electrochemistry models: A tutorial. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 3906–3912. IEEE.

[Moura et al., 2016] Moura, S. J., Argomedo, F. B., Klein, R., Mirtabatabaei, A., and Krstic, M. (2016). Battery state estimation for a single particle model with electrolyte dynamics. *IEEE Transactions on Control Systems Technology*, 25(2):453–468.

[Ng et al., 2009] Ng, K. S., Moo, C.-S., Chen, Y.-P., and Hsieh, Y.-C. (2009). Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied energy*, 86(9):1506–1511.

[Northrop et al., 2011] Northrop, P. W., Ramadesigan, V., De, S., and Subramanian, V. R. (2011). Coordinate transformation, orthogonal collocation, model reformulation and simulation of electrochemical-thermal behavior of lithium-ion battery stacks. *Journal of The Electrochemical Society*, 158(12):A1461.

[Ohlberger et al., 2016] Ohlberger, M., Rave, S., and Schindler, F. (2016). Model reduction for multiscale lithium-ion battery simulation. In *Numerical Mathematics and Advanced Applications ENUMATH 2015*, pages 317–331. Springer.

[Pancaldi, 1990] Pancaldi, G. (1990). Volta and his battery. *Historical studies in the physical and biological sciences*, 21(1):123–160.

[Perez et al., 2017] Perez, H., Dey, S., Hu, X., and Moura, S. (2017). Optimal charging of li-ion batteries via a single particle model with electrolyte and thermal dynamics. *Journal of The Electrochemical Society*, 164(7):A1679.

[Perez and Moura, 2015] Perez, H. E. and Moura, S. J. (2015). Sensitivity-based interval pde observer for battery soc estimation. In *2015 American Control Conference (ACC)*, pages 323–328. IEEE.

[Platzer and Clarke, 2007] Platzer, A. and Clarke, E. M. (2007). The image computation problem in hybrid systems model checking. In *International Workshop on Hybrid Systems: Computation and Control*, pages 473–486. Springer.

[Plett, 2006] Plett, G. L. (2006). Sigma-point kalman filtering for battery management systems of lipb-based hev battery packs: Part 2: Simultaneous state and parameter estimation. *Journal of power sources*, 161(2):1369–1384.

[Pozzi et al., 2018] Pozzi, A., Ciaramella, G., Gopalakrishnan, K., Volkwein, S., and Raimondo, D. M. (2018). Optimal design of experiment for parameter estimation of a single particle model for lithiumion batteries. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6482–6487. IEEE.

[Pozzi et al., 2020] Pozzi, A., Zambelli, M., Ferrara, A., and Raimondo, D. M. (2020). Balancing-aware charging strategy for series-connected lithium-ion cells: A nonlinear model predictive control approach. *IEEE Transactions on Control Systems Technology*, 28(5):1862–1877.

[Prada et al., 2012] Prada, E., Di Domenico, D., Creff, Y., Bernard, J., Sauvant-Moynot, V., and Huet, F. (2012). Simplified electrochemical and thermal model of lifepo4-graphite li-ion batteries for fast charge applications. *Journal of The Electrochemical Society*, 159(9):A1508.

[Rahimi-Eichi et al., 2013] Rahimi-Eichi, H., Ojha, U., Baronti, F., and Chow, M.-Y. (2013). Battery management system: An overview of its application in the smart grid and electric vehicles. *IEEE industrial electronics magazine*, 7(2):4–16.

[Rath et al., 2020] Rath, S., Hoedemaekers, E., and Wilkins, S. (2020). Core temperature estimation for a cylindrical cell battery module. In *2020 Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, pages 1–10. IEEE.

[Rauh and Aschemann, 2012] Rauh, A. and Aschemann, H. (2012). Sensitivity-based state and parameter estimation for lithium-ion battery systems. In *Proc. of the IX International Conference SICPRO*, pages 469–485.

[Rausch et al., 2014] Rausch, M., Klein, R., Streif, S., Pankiewitz, C., and Findeisen, R. (2014). Set-based state of charge estimation for lithium-ion batteries. In *2014 American Control Conference*, pages 1566–1571. IEEE.

[Rego et al., 2022] Rego, B. S., Locatelli, D., Raimondo, D. M., and Raffo, G. V. (2022). Joint state and parameter estimation based on constrained zonotopes. *Automatica*, 142:110425.

[Rego et al., 2020] Rego, B. S., Raffo, G. V., Scott, J. K., and Raimondo, D. M. (2020). Guaranteed methods based on constrained zonotopes for set-valued state estimation of nonlinear discrete-time systems. *Automatica*, 111:108614.

[Rego et al., 2021] Rego, B. S., Scott, J. K., Raimondo, D. M., and Raffo, G. V. (2021). Set-valued state estimation of nonlinear discrete-time systems with nonlinear invariants based on constrained zonotopes. *Automatica*, 129:109638.

[Rumpf et al., 2018] Rumpf, K., Rheinfeld, A., Schindler, M., Keil, J., Schua, T., and Jossen, A. (2018). Influence of cell-to-cell variations on the inhomogeneity of lithium-ion battery modules. *Journal of The Electrochemical Society*, 165(11):A2587.

[Safari and Delacourt, 2010] Safari, M. and Delacourt, C. (2010). Mathematical modeling of lithium iron phosphate electrode: galvanostatic charge/discharge and path dependence. *Journal of The Electrochemical Society*, 158(2):A63.

[Samadi et al., 2013] Samadi, M., Alavi, S., and Saif, M. (2013). Online state and parameter estimation of the li-ion battery in a bayesian framework. In *2013 American Control Conference*, pages 4693–4698.

[Sand and Burrage, 1998] Sand, J. and Burrage, K. (1998). A jacobi waveform relaxation method for odes. *SIAM Journal on Scientific Computing*, 20(2):534–552.

[Santhanagopalan et al., 2006] Santhanagopalan, S., Guo, Q., Ramadass, P., and White, R. E. (2006). Review of models for predicting the cycling performance of lithium ion batteries. *Journal of power sources*, 156(2):620–628.

[Scott et al., 2016] Scott, J. K., Raimondo, D. M., Marseglia, G. R., and Braatz, R. D. (2016). Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica*, 69:126–136.

[Scrosati, 2011] Scrosati, B. (2011). History of lithium batteries. *Journal of solid state electrochemistry*, 15(7):1623–1630.

[Scrosati, 2002] Scrosati, W. A. v. S. B. (2002). *Advances in lithium-ion batteries*. Springer Science & Business Media.

[Seo et al., 2017] Seo, M., Goh, T., Park, M., Koo, G., and Kim, S. W. (2017). Detection of internal short circuit in lithium ion battery using model-based switching model method. *Energies*, 10(1):76.

[Shamma and Tu, 1997] Shamma, J. S. and Tu, K.-Y. (1997). Approximate set-valued observers for nonlinear systems. *IEEE transactions on automatic control*, 42(5):648–658.

[Shen et al., 2017] Shen, P., Ouyang, M., Lu, L., Li, J., and Feng, X. (2017). The co-estimation of state of charge, state of health, and state of function for lithium-ion batteries in electric vehicles. *IEEE Transactions on vehicular technology*, 67(1):92–103.

[Sheng and Xiao, 2015] Sheng, H. and Xiao, J. (2015). Electric vehicle state of charge estimation: Nonlinear correlation and fuzzy support vector machine. *Journal of Power sources*, 281:131–137.

[Singh et al., 2013] Singh, A., Izadian, A., and Anwar, S. (2013). Fault diagnosis of li-ion batteries using multiple-model adaptive estimation. In *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*, pages 3524–3529. IEEE.

[Subramanian et al., 2005] Subramanian, V. R., Diwakar, V. D., and Tapriyal, D. (2005). Efficient macro-micro scale coupled modeling of batteries. *Journal of The Electrochemical Society*, 152(10):A2002.

[Sulzer et al., 2021] Sulzer, V., Marquis, S. G., Timms, R., Robinson, M., and Chapman, S. J. (2021). Python battery mathematical modelling (pybamm). *Journal of Open Research Software*, 9(1).

[Tarascon and Armand, 2001] Tarascon, J.-M. and Armand, M. (2001). Issues and challenges facing rechargeable lithium batteries. *nature*, 414(6861):359–367.

[Torchio et al., 2016] Torchio, M., Magni, L., Gopaluni, R. B., Braatz, R. D., and Raimondo, D. M. (2016). Lionsimba: a matlab framework based on a finite volume model suitable for li-ion battery design, simulation, and control. *Journal of The Electrochemical Society*, 163(7):A1192.

[Tran and Fowler, 2020] Tran, M.-K. and Fowler, M. (2020). A review of lithium-ion battery fault diagnostic algorithms: Current progress and future challenges. *Algorithms*, 13(3):62.

[Truchot et al., 2014] Truchot, C., Dubarry, M., and Liaw, B. Y. (2014). State-of-charge estimation and uncertainty for lithium-ion battery strings. *Applied Energy*, 119:218–227.

[Valøen and Reimers, 2005] Valøen, L. O. and Reimers, J. N. (2005). Transport properties of lipf6-based li-ion battery electrolytes. *Journal of The Electrochemical Society*, 152(5):A882.

[van de Ven et al., 2022] van de Ven, B. A., Sneijders, R., Hoekstra, F. S., Bergveld, H. J., and Donkers, M. T. (2022). Combined cell-level estimation of state-of-charge and temperature in battery packs. In *2022 American Control Conference, ACC 2022*.

[Walker, 2011] Walker, H. F. (2011). Anderson acceleration: Algorithms and implementations. *WPI Math. Sciences Dept. Report MS-6-15-50*.

[Walker and Ni, 2011] Walker, H. F. and Ni, P. (2011). Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735.

[Wang et al., 2011] Wang, J., Liu, P., Hicks-Garner, J., Sherman, E., Soukiazian, S., Verbrugge, M., Tataria, H., Musser, J., and Finamore, P. (2011). Cycle-life model for graphite-lifepo4 cells. *Journal of power sources*, 196(8):3942–3948.

[Wang et al., 2019] Wang, S.-L., Fernandez, C., Cao, W., Zou, C.-Y., Yu, C.-M., and Li, X.-X. (2019). An adaptive working state iterative calculation method of the power battery by using the improved kalman filtering algorithm and considering the relaxation effect. *Journal of Power Sources*, 428:67–75.

[Wang et al., 2018] Wang, S.-L., Yu, C.-M., Fernandez, C., Chen, M.-J., Li, G.-L., and Liu, X.-H. (2018). Adaptive state-of-charge estimation method for an aeronautical lithium-ion battery pack based on a reduced particle-unscented kalman filter. *Journal of Power Electronics*, 18(4):1127–1139.

[Wang et al., 2017] Wang, Z., Hong, J., Liu, P., and Zhang, L. (2017). Voltage fault diagnosis and prognosis of battery systems based on entropy and z-score for electric vehicles. *Applied energy*, 196:289–302.

[White et al., 1985] White, J. A., Sangiovanni-Vincentelli, A., Odeh, F., and Ruehli, A. (1985). *Waveform relaxation: Theory and practice*. Electronics Research Laboratory, College of Engineering, UCB Berkeley, CA.

[Wu et al., 2022] Wu, X., Ma, J., Wang, J., Zhang, X., Zhou, G., and Liang, Z. (2022). Progress, key issues, and future prospects for li-ion battery recycling. *Global Challenges*, page 2200067.

[Xia et al., 2017a] Xia, B., Shang, Y., Nguyen, T., and Mi, C. (2017a). A correlation based fault detection method for short circuits in battery packs. *Journal of power Sources*, 337:1–10.

[Xia et al., 2016] Xia, B., Zhao, X., De Callafon, R., Garnier, H., Nguyen, T., and Mi, C. (2016). Accurate lithium-ion battery parameter estimation with continuous-time system identification methods. *Applied energy*, 179:426–436.

[Xia et al., 2017b] Xia, L., Najafi, E., Li, Z., Bergveld, H., and Donkers, M. (2017b). A computationally efficient implementation of a full and reduced-order electrochemistry-based model for li-ion batteries. *Applied Energy*, 208:1285–1296.

[Xing et al., 2014] Xing, Y., He, W., Pecht, M., and Tsui, K. L. (2014). State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures. *Applied Energy*, 113:106–115.

[Xiong et al., 2012] Xiong, J., Banvait, H., Li, L., Chen, Y., Xie, J., Liu, Y., Wu, M., and Chen, J. (2012). Failure detection for over-discharged li-ion batteries. In *2012 IEEE International Electric Vehicle Conference*, pages 1–5. IEEE.

[Xiong et al., 2019] Xiong, R., Yu, Q., Shen, W., Lin, C., and Sun, F. (2019). A sensor fault diagnosis method for a lithium-ion battery pack in electric vehicles. *IEEE Transactions on Power Electronics*, 34(10):9709–9718.

[Yang et al., 2019] Yang, F., Song, X., Xu, F., and Tsui, K.-L. (2019). State-of-charge estimation of lithium-ion batteries via long short-term memory network. *Ieee Access*, 7:53792–53799.

[Yoshida et al., 2006] Yoshida, T., Takahashi, M., Morikawa, S., Ihara, C., Katsukawa, H., Shiratsuchi, T., and Yamaki, J.-i. (2006). Degradation mechanism and life prediction of lithium-ion batteries. *Journal of The Electrochemical Society*, 153(3):A576.

[Yu et al., 2017] Yu, Q., Xiong, R., and Lin, C. (2017). Online estimation of state-of-charge based on the h infinity and unscented kalman filters for lithium ion batteries. *Energy Procedia*, 105:2791–2796.

[Zhang et al., 2018] Zhang, C., Allafi, W., Dinh, Q., Ascencio, P., and Marco, J. (2018). Online estimation of battery equivalent circuit model parameters and state of charge using decoupled least squares technique. *Energy*, 142:678–688.

[Zhang et al., 2020] Zhang, D., Couto, L. D., Gill, P., Benjamin, S., Zeng, W., and Moura, S. J. (2020). Interval observer for soc estimation in parallel-connected lithium-ion batteries. In *2020 American Control Conference (ACC)*, pages 1149–1154. IEEE.

[Zhang et al., 2021] Zhang, D., Couto, L. D., Gill, P. S., Benjamin, S., Zeng, W., and Moura, S. J. (2021). Thermal-enhanced adaptive interval estimation in battery packs with heterogeneous cells. *IEEE Transactions on Control Systems Technology*, pages 1–14.

[Zhao et al., 2017] Zhao, Y., Liu, P., Wang, Z., Zhang, L., and Hong, J. (2017). Fault and defect diagnosis of battery for electric vehicles based on big data analysis methods. *Applied Energy*, 207:354–362.

[Zhou et al., 2020] Zhou, W., Huang, R., Liu, K., and Zhang, W. (2020). A novel interval-based approach for quantifying practical parameter identifiability of a lithium-ion battery model. *International Journal of Energy Research*, 44(5):3558–3573.

[Ziegler and Trancik, 2021] Ziegler, M. S. and Trancik, J. E. (2021). Re-examining rates of lithium-ion battery technology improvement and cost decline. *Energy & Environmental Science*, 14(4):1635–1651.