

# POLITECNICO DI TORINO

MASTER's Degree in Automotive Engineering



MASTER's Degree Thesis

## State Estimation for Automotive Batteries through Extended Kalman Filter

**Supervisors**

Prof. Federico MIRETTI

Prof. Daniela Anna MISUL

**Candidate**

Federico RAVETTI - 330728

**21 November 2025**



*“I have always believed that you should never, ever give up  
and you should always keep fighting  
even when there is only the slightest chance.”*

— Michael Schumacher



# Acknowledgments

I would like to express my gratitude to my supervisor, Prof. Federico Miretti, and my co-supervisor, Prof. Daniela Anna Misul, for their availability and support during the development of this work.

My deepest and most heartfelt thanks go to my family — my mother, my father, and my brother — for all the sacrifices you have made over these years and for always believing in me, even when I doubted myself. Your love, patience, and unwavering support have been the foundation of everything I have achieved.

A special thought goes to my beloved grandmother, a constant example of strength and generosity, and to my uncles and aunts, who have always been there with affection and encouragement.

My endless gratitude goes to my girlfriend, Micaela, who has been a steady source of love, strength, and serenity. You stood by me through every challenge with kindness and patience. Without you, none of this would have carried the same meaning.

I would also like to thank my second family — your parents and your brother — for welcoming me as one of their own. Your warmth, care, and constant encouragement have been a true blessing.

My sincere appreciation also goes to all my friends and to the wonderful people I have met along this journey. Together we shared intense days, challenges, laughter, and memories that will stay with me forever.

A heartfelt acknowledgment goes to my employer, Ivan, my general director, Irene, and all my colleagues. Thank you for believing in me and for your constant motivation.

Finally, a thought filled with love and nostalgia goes to you, Grandpa and Raja. To you, who watch over me from above.



# Abstract

The performance of electric and hybrid vehicles relies on an effective battery management strategy to ensure safety, driving range, and durability. Among all monitored parameters, the state of charge (SOC) plays an essential role in optimally implementing the control logic strategies. However, since SOC cannot be directly measured, its estimation relies on algorithms capable of guaranteeing a balance between accuracy and computational efficiency.

This thesis proposes the design and implementation of an extended Kalman filter (EKF) for SOC estimation of a lithium-ion cell - LG INR18650 MJ1 - within *Matlab* and *Simulink* environments. The cell was first modelled adopting a Dual Polarisation Model (DPM) and characterised through experimental data, with model parameters optimised via Simulink Design Optimisation (SDO) toolbox. Subsequently, the EKF was developed, tuned, and validated under different operating conditions, demonstrating convergence, stability, and robustness against incorrect SOC initialisation and sensor offsets. Specifically, for both tests analysed under correct initialisation conditions, the filter achieved an estimation SOC error profile constrained within  $\pm 1\%$  window, while in the incorrect initialisation scenario, even for the most dynamic and shortest test, the mean SOC error remained limited to  $-2.38\%$ .

The study was then further extended to a configuration composed of six identical cells, denoted as multiple cells EKF *Simulink* model, enabling simultaneous SOC estimation of a battery module. The model was validated across different scenarios, proving its accuracy and convergence capability. Finally, the model was converted into C code and validated through software in the loop (SIL) approach, confirming its suitability for real-time implementation on production-level embedded hardware.



# Table of Contents

|                                                     |            |
|-----------------------------------------------------|------------|
| <b>List of Figures</b>                              | <b>VII</b> |
| <b>List of Tables</b>                               | <b>IX</b>  |
| <b>List of Listing</b>                              | <b>XI</b>  |
| <b>Acronyms</b>                                     | <b>XIV</b> |
| <b>1 Introduction</b>                               | <b>1</b>   |
| 1.1 Thesis Aim . . . . .                            | 3          |
| 1.2 Thesis Organisation . . . . .                   | 4          |
| <b>2 Cell Characterisation</b>                      | <b>5</b>   |
| 2.1 Cell Specifications . . . . .                   | 5          |
| 2.2 Dual Polarisation Model - DPM . . . . .         | 7          |
| 2.3 Characterisation Tests . . . . .                | 9          |
| 2.3.1 Capacity Test . . . . .                       | 9          |
| 2.3.1.1 C-rate Effect . . . . .                     | 11         |
| 2.3.2 Pulse Test . . . . .                          | 12         |
| 2.4 DPM Parameter Estimation . . . . .              | 15         |
| 2.4.1 Tuning A . . . . .                            | 16         |
| 2.4.2 Tuning B . . . . .                            | 17         |
| 2.4.3 Tuning C . . . . .                            | 18         |
| 2.5 DPM Validation . . . . .                        | 20         |
| <b>3 Extended Kalman Filter</b>                     | <b>27</b>  |
| 3.1 Theoretical Background . . . . .                | 27         |
| 3.1.1 Six-step Process . . . . .                    | 28         |
| 3.2 Simulink Implementation - Single Cell . . . . . | 33         |
| 3.2.1 Data Resampling . . . . .                     | 37         |
| 3.2.2 Filter Tuning . . . . .                       | 38         |
| 3.3 Simulink Model Validation . . . . .             | 40         |
| 3.3.1 Correct initial SOC . . . . .                 | 40         |



---

|          |                                                     |           |
|----------|-----------------------------------------------------|-----------|
| 3.3.2    | Incorrect initial SOC . . . . .                     | 42        |
| 3.3.3    | Current offset injection . . . . .                  | 44        |
| <b>4</b> | <b>Simulink Model Extension</b>                     | <b>47</b> |
| 4.1      | Model Extension for Multi-Cell Estimation . . . . . | 47        |
| 4.2      | Multiple cells EKF Model Validation . . . . .       | 49        |
| 4.2.1    | Scenario I . . . . .                                | 50        |
| 4.2.2    | Scenario II . . . . .                               | 50        |
| 4.2.3    | Scenario III . . . . .                              | 51        |
| 4.2.4    | Scenario IV . . . . .                               | 53        |
| <b>5</b> | <b>Code Generation</b>                              | <b>57</b> |
| 5.1      | Code Setting Parameters . . . . .                   | 57        |
| 5.2      | Simulink S-Function . . . . .                       | 59        |
| 5.3      | S-Function validation . . . . .                     | 61        |
| 5.3.1    | Scenario 1 . . . . .                                | 62        |
| 5.3.2    | Scenario 2 . . . . .                                | 64        |
| <b>6</b> | <b>Conclusions</b>                                  | <b>67</b> |
| 6.1      | Future Developments . . . . .                       | 68        |
| <b>A</b> | <b>Cad Model</b>                                    | <b>69</b> |
| <b>B</b> | <b>Matlab Code EKF</b>                              | <b>71</b> |
|          | <b>Bibliography</b>                                 | <b>75</b> |

# List of Figures

|      |                                                                   |    |
|------|-------------------------------------------------------------------|----|
| 1.1  | SOC estimation strategies . . . . .                               | 2  |
| 1.2  | Thesis flow chart . . . . .                                       | 3  |
| 2.1  | LIB operating principle . . . . .                                 | 6  |
| 2.2  | Dual polarisation cell model . . . . .                            | 8  |
| 2.3  | Capacity Test . . . . .                                           | 10 |
| 2.4  | OCV curve vs. SOC . . . . .                                       | 11 |
| 2.5  | C-rate effect on discharge OCV curve vs. cell capacity . . . . .  | 12 |
| 2.6  | Pulse Test . . . . .                                              | 13 |
| 2.7  | SOC profile during Pulse Test . . . . .                           | 14 |
| 2.8  | Mechanism for parameter estimation on Pulse Test . . . . .        | 15 |
| 2.9  | Pulse Test voltage tuning A . . . . .                             | 17 |
| 2.10 | Pulse Test voltage tuning B . . . . .                             | 18 |
| 2.11 | Pulse Test voltage tuning C . . . . .                             | 19 |
| 2.12 | Dynamic Stress Test . . . . .                                     | 20 |
| 2.13 | DST variable and fixed time interval . . . . .                    | 21 |
| 2.14 | SOC profile during DST . . . . .                                  | 22 |
| 2.15 | DPM validation - DST cell voltage . . . . .                       | 23 |
| 2.16 | Tuned OCV curve vs. SOC . . . . .                                 | 24 |
| 2.17 | Tuned DPM resistances vs. SOC . . . . .                           | 24 |
| 2.18 | Tuned DPM time constants and capacitors vs. SOC . . . . .         | 25 |
| 3.1  | EKF operating principle . . . . .                                 | 32 |
| 3.2  | Single cell <i>EKF</i> subsystems overview . . . . .              | 33 |
| 3.3  | Initialisation & Delay subsystem . . . . .                        | 34 |
| 3.4  | State Transition Matrices subsystem . . . . .                     | 35 |
| 3.5  | Prediction phase subsystem . . . . .                              | 36 |
| 3.6  | Estimation phase subsystem . . . . .                              | 36 |
| 3.7  | Resampling process at fixed time interval . . . . .               | 38 |
| 3.8  | Pulse Test Single cell EKF validation - Correct initial SOC . . . | 41 |
| 3.9  | DST Single cell EKF validation - Correct initial SOC . . . . .    | 42 |
| 3.10 | Pulse Test Single cell EKF validation - Incorrect initial SOC . . | 43 |



---

|      |                                                                   |    |
|------|-------------------------------------------------------------------|----|
| 3.11 | DST Single cell EKF validation - Incorrect initial SOC . . . . .  | 44 |
| 3.12 | DST offset current profile . . . . .                              | 45 |
| 3.13 | DST Single cell EKF validation - Offset current profile . . . . . | 46 |
|      |                                                                   |    |
| 4.1  | Multiple cells <i>EKF</i> subsystems overview . . . . .           | 48 |
| 4.2  | Validation scenario I - SOC profiles . . . . .                    | 50 |
| 4.3  | Validation scenario II - SOC profiles . . . . .                   | 51 |
| 4.4  | Validation scenario III . . . . .                                 | 53 |
| 4.5  | Validation scenario IV . . . . .                                  | 55 |
|      |                                                                   |    |
| 5.1  | Multiple cells EKF S-Function . . . . .                           | 60 |
| 5.2  | <i>Simulink</i> model for <b>S-Function</b> validation . . . . .  | 62 |
| 5.3  | SIL validation scenario 1 . . . . .                               | 63 |
| 5.4  | SIL validation scenario 2 . . . . .                               | 65 |
|      |                                                                   |    |
| A.1  | 3D Cell Cad Model . . . . .                                       | 69 |

# List of Tables

|     |                                                                  |    |
|-----|------------------------------------------------------------------|----|
| 2.1 | Cell dimension and weight . . . . .                              | 6  |
| 2.2 | Cell electrical characteristics . . . . .                        | 7  |
| 2.3 | Discharge characteristics at different C-rates . . . . .         | 12 |
| 2.4 | Resistance limits for optimisation . . . . .                     | 19 |
| 2.5 | Voltage RMSE for different tunings . . . . .                     | 20 |
| 3.1 | <i>EKF</i> mask parameters . . . . .                             | 33 |
| 3.2 | Validation process for different scenarios . . . . .             | 46 |
| 4.1 | Multiple cells <i>EKF</i> mask parameters . . . . .              | 49 |
| 4.2 | Validation scenarios applied to the multiple cells EKF . . . . . | 50 |
| 4.3 | Validation scenario III - Initial SOC's . . . . .                | 51 |
| 4.4 | Validation scenario III - Summary results . . . . .              | 53 |
| 4.5 | Validation scenario IV - Initial SOC's . . . . .                 | 54 |
| 4.6 | Validation scenario IV - Summary results . . . . .               | 55 |
| 5.1 | S-Function mask parameters . . . . .                             | 59 |
| 5.2 | SIL validation scenarios . . . . .                               | 62 |
| 5.3 | Cell voltages offsets . . . . .                                  | 64 |
| B.1 | EKF Matlab function Input & Output . . . . .                     | 71 |



# Listings

|     |                                                  |    |
|-----|--------------------------------------------------|----|
| 2.1 | DST fixed time conversion . . . . .              | 20 |
| 2.2 | DST parameter estimation . . . . .               | 22 |
| 3.1 | OCV vs. SOC derivative . . . . .                 | 31 |
| B.1 | MATLAB EKF function for SOC estimation . . . . . | 72 |



# Acronyms

**BEV** Battery Electric Vehicle

**BMS** Battery Management System

**CC** Constant Current

**CV** Constant Voltage

**DEKF** Dual Extended Kalman Filter

**DPM** Dual Polarisation Model

**DST** Dynamic Stress Test

**ECM** Equivalent Circuit Model

**EKF** Extended Kalman Filter

**EM** Electric Motor

**ESR** Equivalent Series Resistance

**EU** European Union

**HEV** Hybrid Electric Vehicle

**HPPC** Hybrid Pulse Power Characterisation

**ICCT** International Council on Clean Transportation

**ICE** Internal Combustion Engine

**IEA** International Energy Agency

**KF** Linear Kalman Filter

**LIB** Lithium Ion Battery



**LUT** Look-Up Table

**NMC** Nickel-Manganese-Cobalt

**OCV** Open Circuit Voltage

**OEM** Original Equipment Manufacturer

**PHEV** Plug-in Hybrid Electric Vehicle

**RMSE** Root Mean Square Error

**S&H** Sample-And-Hold

**SDO** Simulink Design Optimisation

**SIL** Software-In-the-Loop

**SOC** State Of Charge

**SOH** State Of Health





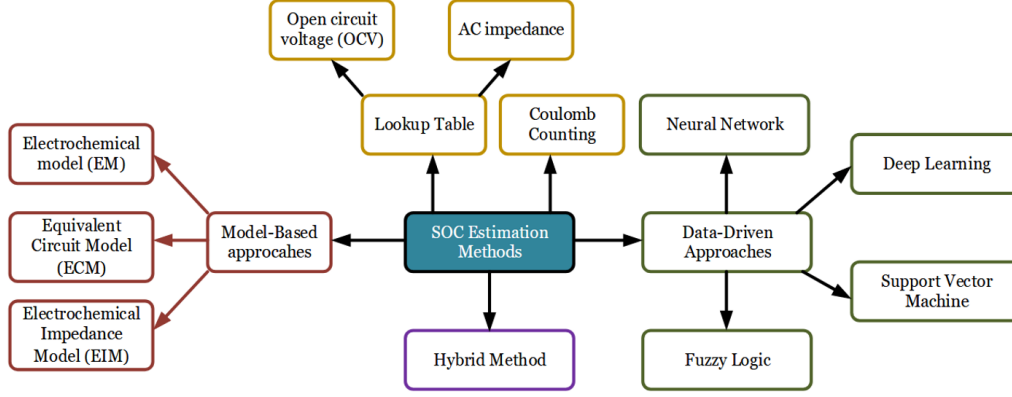
# Chapter 1

## Introduction

In recent years, due to growing environmental concerns about climate change and air quality, the automotive sector has experienced a significant transformation, especially at the powertrain system level. In fact, to comply with the stringent emission reduction targets imposed by the European Union (EU), OEMs have been forced to replace part of their traditional ICE fleets with vehicles equipped with a battery pack and one or more electric motors (EM). According to international authorities such as IEA and ICCT, this ongoing transition towards electrification is confirmed by the increased market penetration of BEVs, PHEVs, and HEVs [1, 2].

In this content, one of the most relevant variables is the battery State Of Charge (SOC), which represents the remaining usable capacity within the battery pack relative to its maximum available capacity at a given time instant. Being a non directly measurable quantity, an accurate estimation is fundamental, as it significantly impacts vehicle performance, safety, and driving range. In addition, a reliable estimation of the SOC is essential for the proper application of the optimal energy management control logic, both to improve driving efficiency and to extend the battery lifespan, considering that, on average, the battery accounts for 30% to 50% of the total vehicle cost [3].

Considering its central role, several SOC estimation strategies have been developed over the years, as illustrated in Figure 1.1 [4].



**Figure 1.1:** SOC estimation strategies

The Coulomb counting method, representing a traditional approach, estimates the SOC by integrating the current applied to the cell over time. Despite its simplicity, its accuracy strongly depends on the precise knowledge of the initial SOC, a rigorous current sensor calibration, and the continuous update of the maximum available cell capacity. Another classical approach is the Look-Up Table method, which estimates the SOC by exploiting its relationship with external parameters such as OCV or impedance to estimate SOC. This strategy requires intensive experimental characterisation tests to establish the relationship and, being applicable only to static cells, is unsuitable for online SOC estimation. For these reasons, more advanced approaches have been introduced and adopted for real-time applications. The model-based methods estimate the SOC by solving differential equations of the model developed to reproduce the cell dynamics. Typically, these techniques are combined with filtering algorithms to improve accuracy, reliability, and mitigate measurement and modelling errors. Instead, data-driven methods, contrarily to model-based, do not require any knowledge of the internal system processes, as they exploit machine learning algorithms trained on large experimental datasets. Although particularly useful in the case of absence or limited information about cell internal characteristics or chemical reactions, they are highly dependent on the quality of the training data. Finally, the hybrid methods, recently introduced, combine the advantages of model-based techniques with those of data-driven models, offering improved accuracy and robustness in SOC estimation [4].

Among the various techniques, in this thesis work, a model-based strategy was adopted. Specifically, an Extended Kalman Filter (EKF) built on a Dual Polarisation circuit model (DPM) was designed, as it provides the best compromise between accuracy, data requirements, computational effort, and real-time applicability.

In order to achieve the thesis objectives, all the data analysed were extracted from the results of the laboratory tests carried out by third parties on LG

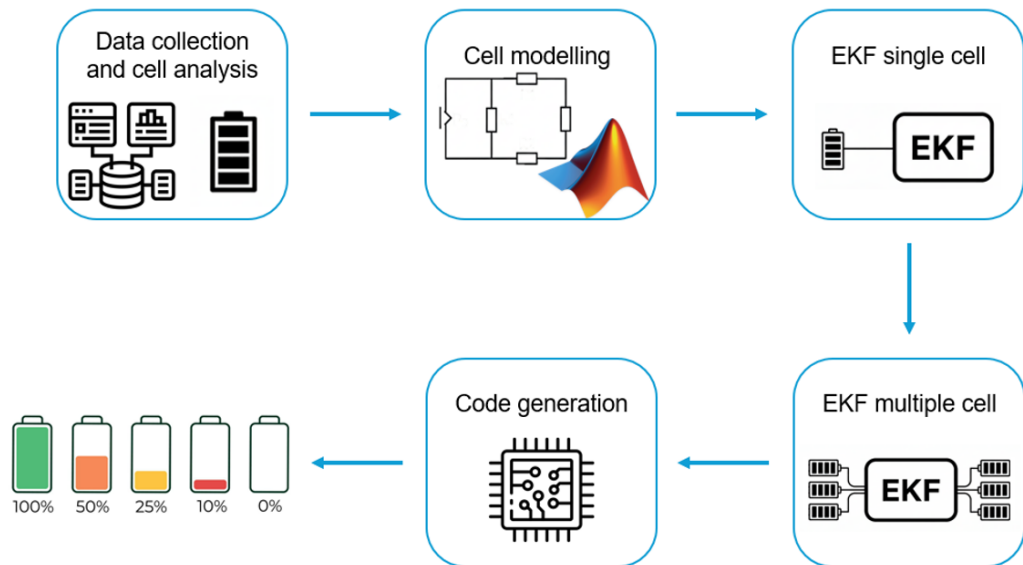
INR18650 MJ1 cell [5]. Although the database includes outcomes at two different ambient temperatures, specifically  $25^{\circ}\text{C}$  and  $45^{\circ}\text{C}$ , only the former were considered, as they are representative of standard environmental conditions.

## 1.1 Thesis Aim

This thesis work presents several objectives that can be summarised as follows:

1. Development of a mathematical model able to describe the dynamic behaviour of the adopted lithium-ion cell under operating conditions representative of a new generation powertrains (BEV, PHEV or HEV);
2. Implementation and validation of an extended Kalman filter for state of charge estimation in *Simulink* environment, based on the developed cell model;
3. Extension of the developed EKF from a single cell model to a model describing a battery module composed of six identical cells. This objective represents the main novelty with respect to the articles presented in the literature. Indeed, while existing works analyse and consider the EKF mainly at single cell level, the proposed thesis work aims at implementing a model capable of simultaneously estimating SOC of multiple cells;
4. Generation and software in the loop (SIL) validation of the C code derived from the designed model, intended to be deployed on real hardware with production-level specifications.

Figure 1.2 presents the flow chart adopted to achieve the thesis objectives:



**Figure 1.2:** Thesis flow chart



## 1.2 Thesis Organisation

To facilitate the reading and in order to follow the process path adopted to achieve the thesis objectives, the work is subdivided as follows:

- *Chapter 2: Cell Characterisation* provides a detailed description of the cell under examination and the model adopted for its characterisation;
- *Chapter 3: Extended Kalman Filter* presents the theoretical principles supporting the implementation of the *Simulink* model for single cell SOC estimation;
- *Chapter 4: Simulink Model Extension* proposes the extension of the single-cell EKF *Simulink* model previously developed to a battery module composed of six identical cells, and its validation across different operating scenarios;
- *Chapter 5: Code Generation* provides a detailed description of the procedure adopted for generating the C code of the implemented model and presents its validation through the SIL approach;
- *Chapter 6: Conclusions* closes the thesis work by summarising the relevant results obtained and offering, to the readers, suggestions for possible future developments.

# Chapter 2

## Cell Characterisation

The current chapter focuses on presenting the analysis and procedures carried out for the characterisation of the cell employed in this thesis, the LG INR18650 MJ1. This process, summarised below, constitutes an essential step in the development of the EKF:

1. Overview of cell main characteristics;
2. Description of the adopted cell model;
3. Description of the tests analysed for model parametrisation;
4. Estimation of the model parameters through SDO;
5. Analysis of the results obtained from parameters estimation.

### 2.1 Cell Specifications

The LG INR18650 MJ1 is an electrochemical device able to transform the chemical energy contained in its active material into electrical energy adopted to power the vehicle. In addition, being a secondary cell, the conversion process is reversible, allowing the cell to store energy during braking manoeuvres through regenerative braking.

The cylindrical cell, with its main dimensions reported in Table 2.1 and a 3D model shown in Appendix A, operates on lithium-ion technology and is composed of [6, 7]:

- *Positive electrode* made of NMC, specifically NMC811;
- *Negative electrode* consisting of Silicon-Graphite (Si-Gr), with a silicon content of 3.5% by weight. The addition of silicon is intended to enhance both volumetric and gravimetric energy. Nevertheless, due to high volume expansion when reacting with lithium, its content results to be a trade-off between energy performances and structural stability;

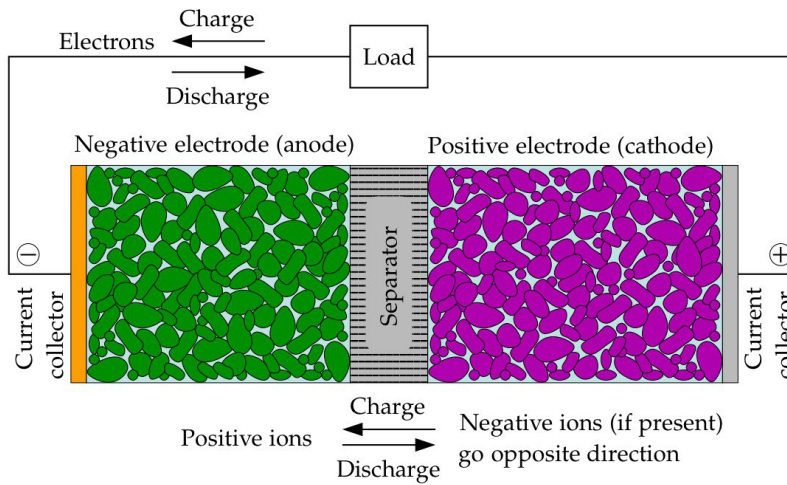
- *Electrolyte*, a non-aqueous organic liquid that enables charge transport between the two electrodes;
- *Separator*, a porous membrane that electrically insulates the positive and negative sides of the cell;
- *Current collectors* which represent key components as they act as conductive interfaces with the external circuit and help minimise the internal resistance.

| Description | Symbol | Value | Unit of Measure |
|-------------|--------|-------|-----------------|
| Diameter    | $d$    | 18.5  | $mm$            |
| Height      | $h$    | 65.1  | $mm$            |
| Weight      | $w$    | 49.0  | $g$             |

**Table 2.1:** Cell dimension and weight

The interaction among all the components described enables the electrochemical processes of LIB cell, governed by reduction-oxidation (*redox*) reactions, to take place. Specifically, during discharge, the negative electrode undergoes oxidation reactions, releasing electrons to the external circuit. The resulting charge imbalance generated by the oxidation is compensated by the migration of lithium-ions from the negative towards the positive electrode (reduction). Similar phenomena occur during charge, where the current supplied from the external circuit reverses the mechanism.

The components of the cell, as well as its operating principle, previously explained, are shown in Figure 2.1 [8].



**Figure 2.1:** LIB operating principle

In conclusion of the cell data analysis, the main electrical characteristics required for the next steps, such as nominal capacity and voltage limits, are listed in Table 2.2 [6].

| Description        | Symbol        | Value                | Unit of Measure |
|--------------------|---------------|----------------------|-----------------|
| Nominal Capacity   | $Q_{nom}$     | 3.50                 | Ah              |
| Nominal Energy     | $E_{nom}$     | 12.72                | Wh              |
| Nominal Voltage    | $V_{nom}$     | 3.64                 | V               |
| Standard Charge    | $i_{stCh}$    | $0.5 \cdot C = 1750$ | mA              |
| Maximum Voltage    | $V_{max}$     | 4.20                 | V               |
| Standard Discharge | $i_{stDisch}$ | $0.2 \cdot C = 700$  | mA              |
| Minimum Voltage    | $V_{min}$     | 2.50                 | V               |

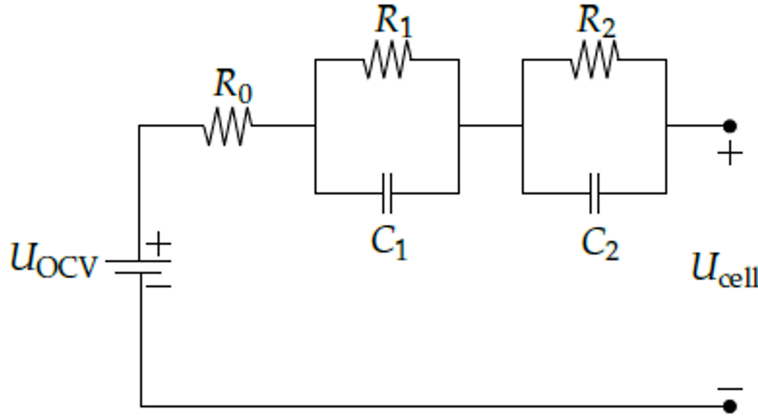
**Table 2.2:** Cell electrical characteristics

## 2.2 Dual Polarisation Model - DPM

Modelling a cell or a battery is an extremely challenging task due to the complexity and non-linearity of the electrochemical processes described in the previous Section 2.1. Their behaviour depends on several factors such as SOC, temperature, ageing, and charging/discharging rate. For these reasons, over the years, different techniques and approaches have been proposed, depending also on the goals to be achieved [9]. Among these, electrochemical models describe, through differential equations, the chemical reactions of the cell faithfully. However, they are extremely difficult to configure as they require a large amount of input data and, due to their low computational efficiency, are not appropriate for real-time applications. In contrast, the electrical equivalent circuit models (ECM) approximate the cell behaviour using basic circuit elements such as resistors, capacitors, and voltage sources. Although these models are less accurate, particularly in capturing the non-linearity of electrochemical processes, their simplicity, standard test instrumentation requirements, and high computational efficiency make them widely adopted in BMS [10]. Therefore, this type of model is employed in the development of the EKF for SOC estimation.

Within the various types of circuit-based models, the simplest results to be the *RINT* made up of a voltage source and a series resistance, and is only able to capture the steady-state behaviour of the cell. To overcome this limitation, resistor-capacitor groups can be introduced in the model to reproduce the dynamic behaviour of the electrochemical processes [11]. Specifically, the chosen configuration, reported in Figure 2.2, was the *Dual Polarisation Model*

(DPM), an evolution of the *Thevenin* model, consisting of a voltage source in series with a resistance and two resistor-capacitor (RC) networks.



**Figure 2.2:** Dual polarisation cell model

The main components of the DPM are the following:

- *OCV*: represents the cell terminal voltage measured under no-load conditions, therefore, when no current is applied to the cell. As will be demonstrated in the next Sections, it is strongly dependent on SOC, thereby making the voltage source variable [12];
- $R_0$ : also known as *equivalent series resistance* (ESR), is mounted directly in series to the voltage source and is intended to describe the instantaneous voltage drop occurring when a current is applied to the cell. The resistor models almost all the components involved in the operating process described in the previous Section, such as the electrolyte, the active materials in the electrodes and the current collectors. Moreover, it is important to mention that the resistor causes ohmic power losses dissipated as heat, reducing the cell efficiency [8, 13];
- RC groups capture the dynamics of the electrochemical processes occurring inside the cell. The first group describes the activation polarisation, also denoted as charge-transfer polarisation, which represents the voltage required to increase the rate of the chemical reaction. The second network, instead, models the voltage associated with the formation of concentration gradients of charge carriers in the electrolyte. This process is known as concentration polarisation, diffusion polarisation, or mass-transport polarisation. The two mentioned phenomena exhibit different speeds: the first, being faster, is characterised by a smaller time constant  $\tau_1$ , while the second, slower, presents a larger time constant  $\tau_2$  ( $\tau_i [s] = R_i \cdot C_i$ ) [13].

It is worth noting that all these components are influenced not only by SOC, but also by operating temperature. Furthermore, the equivalent series resistance is strongly dependent on the SOH, showing an increasing trend during cell ageing, while the RC networks are affected both by the current magnitude and direction [13, 14, 15]. Nevertheless, to limit the complexity of the model, these factors were neglected, maintaining only the dependence of the parameters on SOC.

The DPM can be formulated in both continuous and discrete time. However, since the current application involves the implementation in a BMS operating at fixed time intervals, the discrete-time Equations are reported [16]:

$$\begin{cases} V_{cell}(k) = OCV(SOC(k)) - V_{RC_1}(k) - V_{RC_2}(k) - R_0 \cdot I_{cell}(k) \\ V_{RC_i}(k) = \exp(-\frac{dt}{\tau_i}) \cdot V_{RC_i}(k-1) + R_i \cdot (1 - \exp(-\frac{dt}{\tau_i})) \cdot I_{cell}(k-1) \end{cases} \quad (2.1)$$

In set of Equations 2.1, the variables not previously defined are:

- $V_{cell}$  is the measured output voltage at the cell terminals expressed in  $V$ ;
- $V_{RC_i}$  is the voltage drop across the  $i$  -  $th$  RC group expressed in  $V$ ;
- $R_i$  is the resistance of the  $i$  -  $th$  RC group expressed in  $\Omega$ ;
- $I_{cell}$  is the current, expressed in  $A$ , applied to the cell considered positive during discharge and negative in charging mode;
- $k$  is the actual time step;
- $dt$  is the sampling time expressed in  $s$ .

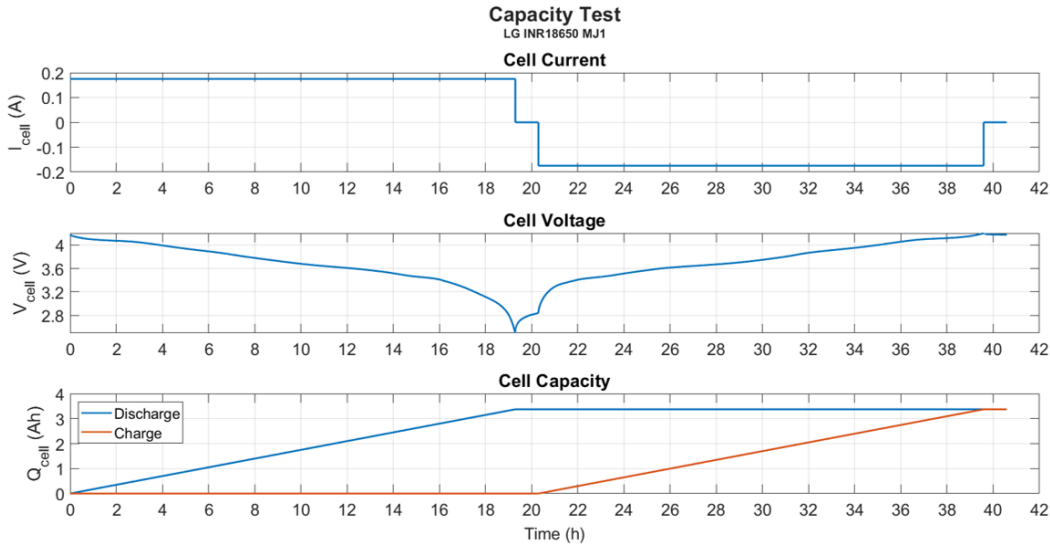
## 2.3 Characterisation Tests

In order to parametrise the components of the proposed DPM, two tests, performed using standard instrumentation, were required. As highlighted in Section 1, these tests were not physically conducted but only their results, reported in the adopted dataset, were analysed [5].

### 2.3.1 Capacity Test

The first examined test, denoted as *Capacity Test*, consists of a full charge or discharge of the cell under constant current (CC) with an extremely low C-rate. This procedure allowed the extraction of the OCV at the cell's terminals, as well as the usable capacity thanks to the minimum influence of ohmic losses and polarisation effects on the cell model.

The test provided in the database, shown in Figure 2.3, includes a discharge phase from maximum to minimum voltage (make reference to Table 2.2), followed by a rest period and a subsequent charging phase restoring back the cell to its maximum voltage. Both the active phases are carried out at  $C/20$ , corresponding to  $175\text{ mA}$ . As visible from Figure 2.3, during the resting period, despite the absence of current applied to the system, the terminal voltage increases from the minimum value. This behaviour, caused by hysteresis phenomenon, cannot be modelled by the DPM and consequently was excluded from the extraction of the OCV curve as a function of SOC [8].

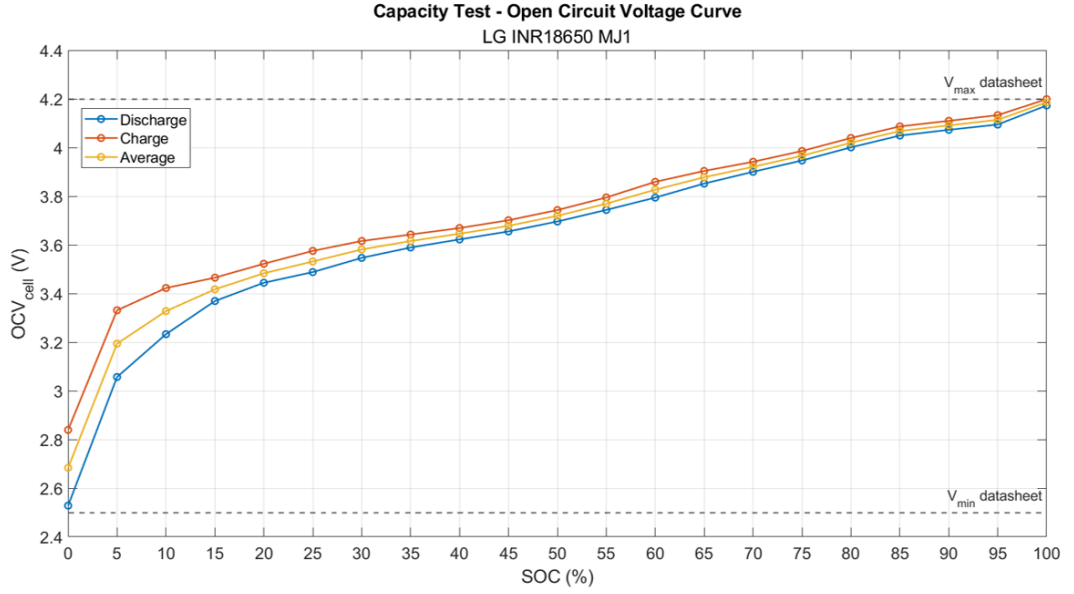


**Figure 2.3:** Capacity Test

To extract the OCV curve, the two active phases were first isolated. For each of them, the corresponding SOC was evaluated by normalising the extracted cell capacity ( $Q_{cell}$ ) with respect to the phase-specific maximum capacity ( $Q_{max}$ ), as reported in Equation 2.2.

$$SOC^{Discharge/Charge} = \frac{Q_{cell}^{Discharge/Charge}}{Q_{max}^{Discharge/Charge}} \quad (2.2)$$

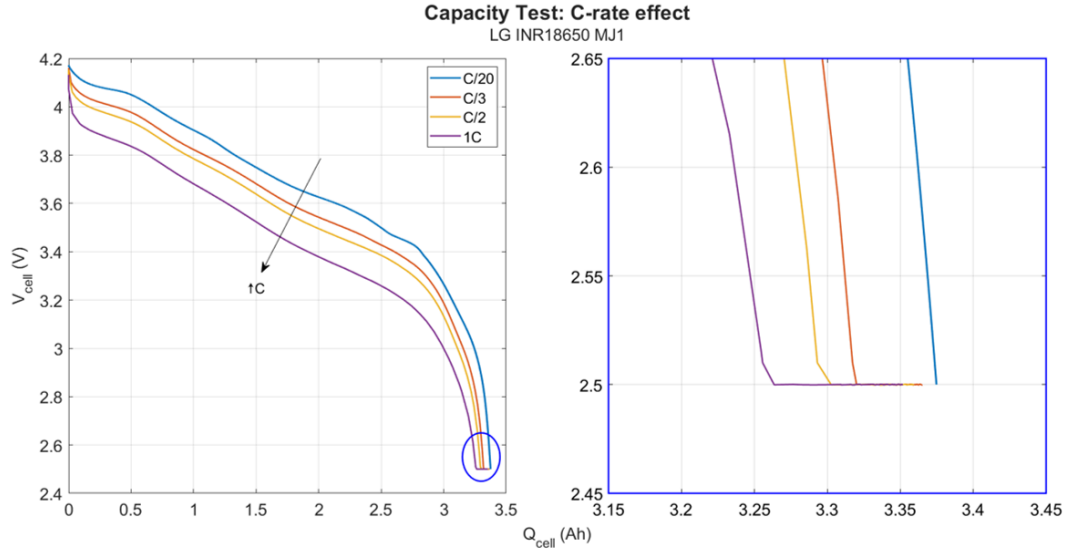
This approach allowed to cover the entire SOC range (0% – 100%) both in discharge and charge. Afterwards, the cell terminal voltage was collected at 5% SOC for both phases and then averaged pointwise. The resulting curve, illustrated in Figure 2.4 in yellow, was adopted for estimating the DPM parameters.



**Figure 2.4:** OCV curve vs. SOC

### 2.3.1.1 C-rate Effect

To better understand the influence of operating conditions on cell behaviour during the test, the effect of current magnitude was inspected. The dataset also includes entire capacity tests performed at four different C-rate levels. However, for graphical reasons, only the discharge stages were considered, although similar considerations could be drawn from the analysis of the charge periods. As evident in Figure 2.5, an increment in the C-rate leads to a reduction in the measured cell voltage. This phenomenon, attributed to increased ohmic losses and polarisation effects, is represented in the DPM by voltage drops, as described by Equation 2.1. In addition, as shown in the zoomed view of the final stage of the discharge phase, the maximum usable capacity ( $Q_{max}^{Discharge}$ ) decreases as the current magnitude increases. Therefore, although higher C-rate levels reduced the test duration, the resulting data were not reliable for model parametrisation.



**Figure 2.5:** C-rate effect on discharge OCV curve vs. cell capacity

Table 2.3, reported below, further supports the discussed observations:

| C-rate | $ I_{\text{cell}} $ (mA) | Discharge Time (h) | $Q_{\text{max}}^{\text{Discharge}}$ (Ah) |
|--------|--------------------------|--------------------|------------------------------------------|
| C/20   | 175                      | 19.3               | 3.375                                    |
| C/3    | 1167                     | 3.1                | 3.365                                    |
| C/2    | 1750                     | 2.3                | 3.359                                    |
| 1C     | 3500                     | 1.4                | 3.352                                    |

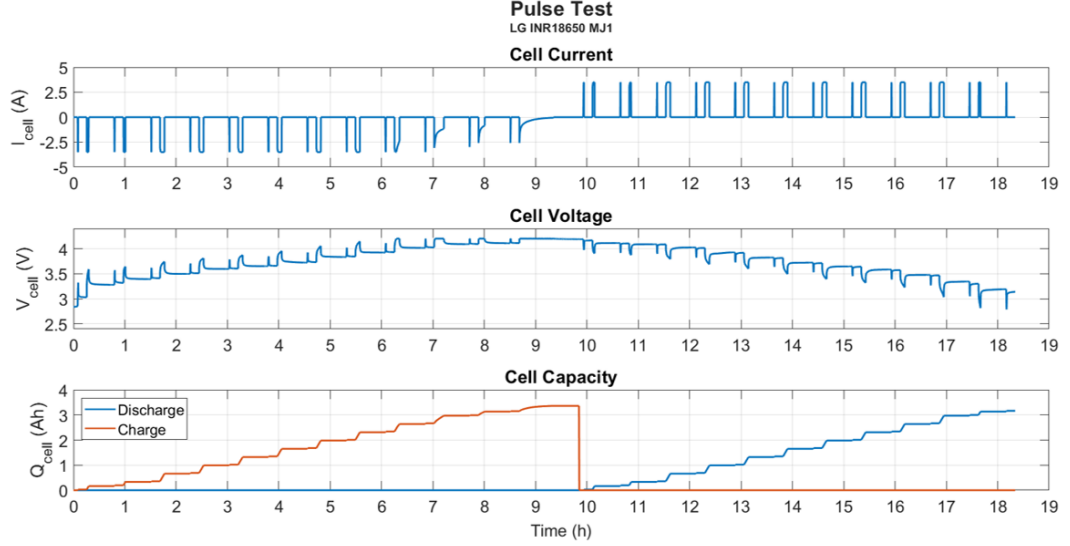
**Table 2.3:** Discharge characteristics at different C-rates

### 2.3.2 Pulse Test

The second test examined, denoted as *Pulse Test*, is used to characterise the resistive and capacitive components of the presented DPM. Typically, to achieve this goal, the Hybrid Pulse Power Characterization (HPPC) is performed. The latter consists of a 1C discharge pulse, followed by a short rest period, a subsequent 1C charge pulse, another rest, and a final 1C discharge pulse intended to reduce by 10% the cell SOC. The entire process is repeated until the cell is fully discharged, ensuring the characterisation of the model over the entire SOC range [14].

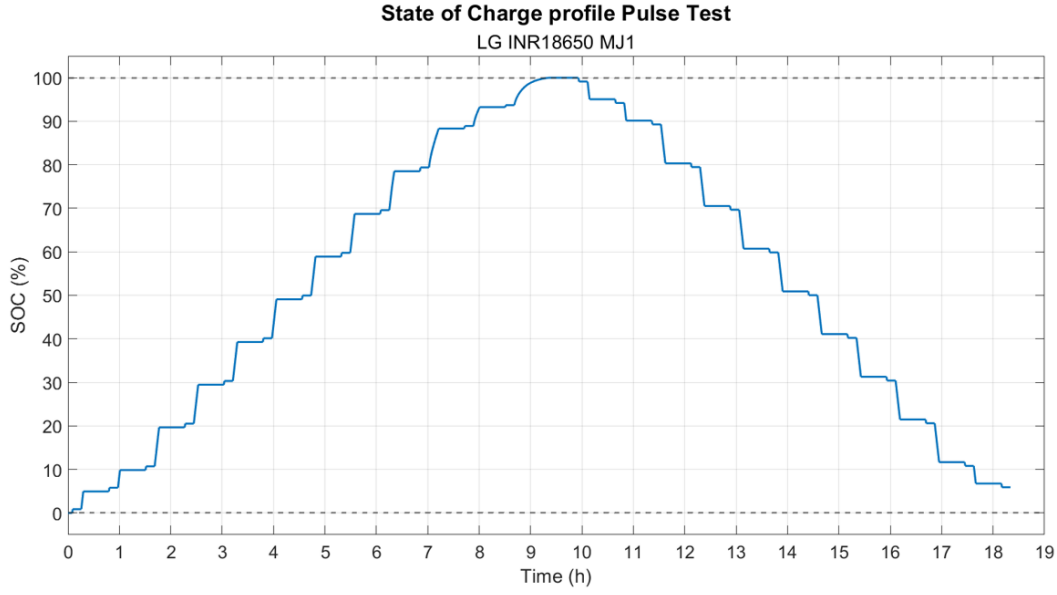
However, in this work, the dataset provides a different pulse test as illustrated in Figure 2.6. Although it differs from the standard test, the outcome results to be comparable, ensuring a perfect description of the model parameters. In particular, the procedure consists of a series of 1C charge pulses separated by relaxation periods to guarantee the chemical stability of the cell. It is important to point out that, towards the end of the charging phase, the current rate is

progressively reduced from 1C to 0.7C to prevent damage to the cell caused by exceeding the upper voltage limit (see Table 2.2). Subsequently, several discharge pulses at 1C, always separated by rest periods, are conducted until the cell approaches the lower voltage limit. At this point, the test is concluded since the application of a further discharge pulse, even if at a lower C-rate, could bring the cell voltage below the lower threshold.



**Figure 2.6:** Pulse Test

The extracted cell capacity, reported in the lower plot of Figure 2.6, was used to determine the SOC profile over the test duration. To perform this operation, the net capacity was evaluated as the difference between the red curve and the blue one during charging. Conversely, in the discharge phase, the opposite difference was considered and subtracted from the SOC reached at the end of charge phase in order to maintain it positive. Figure 2.7 reports the resulting profile and, in accordance with what was previously explained regarding the test stop condition, the SOC does not reach zero at the end of the discharge stage.



**Figure 2.7:** SOC profile during Pulse Test

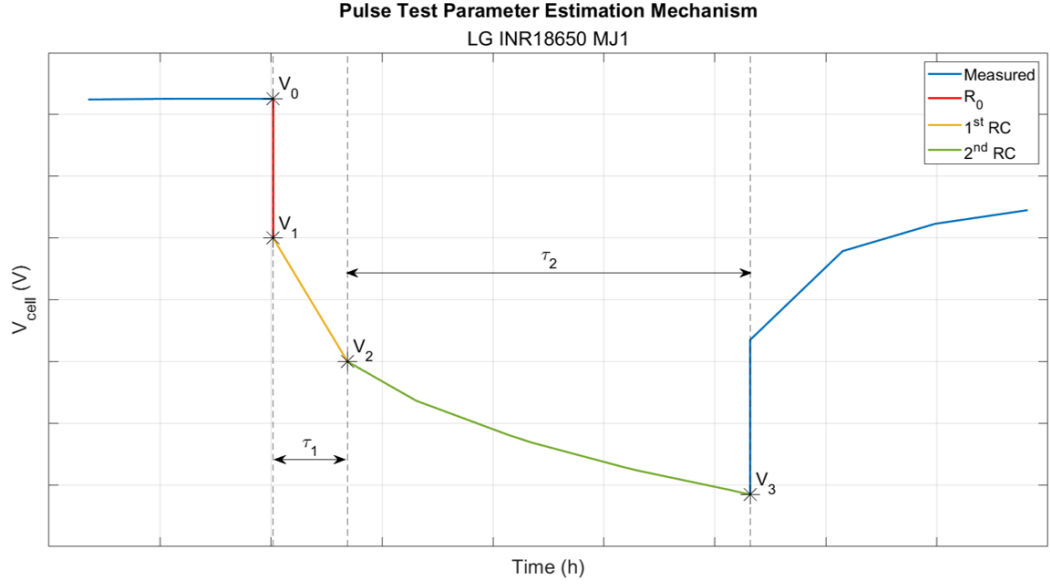
By combining the *Pulse Test* current and voltage data with the SOC profile, it is possible to analyse the dependence of the model parameters on SOC according to the following mechanism. At each pulse, either discharge or charge, an instantaneous voltage variation, respectively, a decrement or an increment, occurs caused by the ESR. By evaluating this voltage difference and knowing the current applied, it is possible to estimate the value of  $R_0$  at that specific SOC level through Equation 2.3. Subsequently, the two polarisation effects intervene contributing to a further, but smoother, voltage variation. The rate of change in this phase is governed by the speed of these two effects, which, as mentioned, is higher for the activation process ( $\tau_1$ ) and lower for the diffusion one ( $\tau_2$ ). This smoother voltage modification allows the characterisation of the two RC networks of the DPM according to Equations 2.4 and 2.5 [14, 15]. Figure 2.8 illustrates the mechanism for a single discharge pulse, but, as expected, the same concepts can be applied during charging.

$$R_0 = \frac{V_1 - V_0}{I_{cell}} \quad (2.3)$$

$$R_1 = \frac{V_2 - V_1}{I_{cell}} \Rightarrow C_1 = \frac{\tau_1}{R_1} \quad (2.4)$$

$$R_2 = \frac{V_3 - V_2}{I_{cell}} \Rightarrow C_2 = \frac{\tau_2}{R_2} \quad (2.5)$$

It is important to highlight that the voltage subscripts in the previous Equations directly correspond to the marker points reported in Figure 2.8.



**Figure 2.8:** Mechanism for parameter estimation on Pulse Test

## 2.4 DPM Parameter Estimation

The last step to conclude the cell characterisation concerned the identification of the most accurate estimation of the DPM parameter profiles ( $R_0$ ,  $R_1$ ,  $R_2$ ,  $\tau_1$ ,  $\tau_2$ ) as a function of SOC. To achieve this goal, the *Simulink* Design Optimisation (SDO) was employed.

Initially, a continuous-time *Simulink* model was created to describe the evolution of the DPM state. In particular, the state of this model, which will be reused for the implementation of the EKF, were the State of Charge and the voltages across the two RC networks, whose dynamic behaviours are described by Equations 2.6.

$$\begin{cases} \dot{SOC} = \frac{-I_{cell} \cdot dt}{Q_{nom}} \\ \dot{V}_{RC_i} = \frac{R_i \cdot I_{cell} - V_{RC_i}}{\tau_i} \end{cases} \quad (2.6)$$

The inputs of the model were the voltage, current, and SOC profile obtained during the *Pulse Test*, while the output was the simulated cell voltage evaluated through first equation of the DPM formulation (make reference to Eq. 2.1). It is important to note that, although this equation was defined in the discrete-time domain, it remained valid in this continuous *Simulink* model.

For what concerns the cell model parameters to be tuned, and required by all reported equations, a 1D LUT was implemented for each of them, with SOC breakpoints discretised every 5%, consistently with the OCV curve (see Figure 2.4).

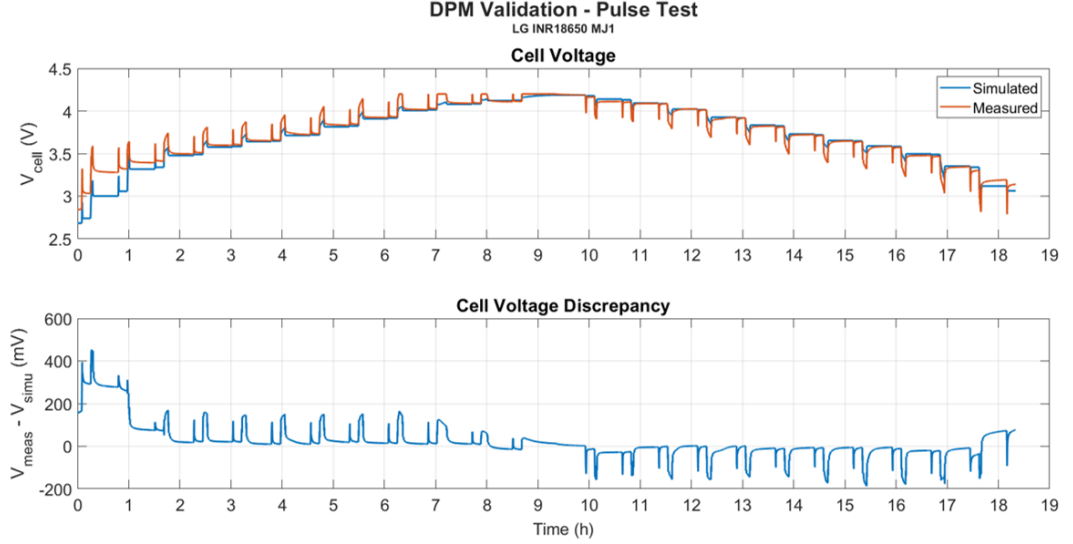
Subsequently, it was necessary to define all the options for the proper initial-

isation of the SDO [17]. The parameter estimation problem was formulated as a Nonlinear Least Squares (`lsqnonlin`) optimisation, where the objective function minimised the sum of the squared difference between the measured voltage and the voltage simulated by the *Simulink* model. [18].

At this stage, with the SDO objective and the problem setup clearly defined, it is necessary to explain the working principle of the optimisation process. The produced *Matlab* script initially runs a simulation of the *Pulse Test* with an initial guess of the DPM parameters defined by the user. Then, the voltage error during the test is evaluated and, on the basis of this result, the software iteratively relaunches the test by adjusting the parameters within a predefined range. This process continues until no further voltage error reduction can be achieved by modifying the parameters. Therefore, in order to define the best estimation, it was possible to control only the initial guess and the range, for each parameter, in which the optimisation process could search. A total of thirteen different tunings were attempted, but only the most relevant are reported, each of them accompanied by the identification code, the description of the main features, and the simulated voltage profile.

### 2.4.1 Tuning A

The first tuning, identified by letter A, was the simplest possible SDO trial. In particular, the OCV curve was taken from the *Capacity Test* (yellow curve Figure 2.4), whereas all the resistive elements ( $R_0$ ,  $R_1$ ,  $R_2$ ) were left free to vary between 0 and 10  $\Omega$ . On the other hand, the time constant ranges were independently defined to better model the polarisation effects. Specifically, the first  $\tau$  was allowed to change from 0 to 1000 seconds, while the second ranged from 1000 up to  $1 \cdot 10^7$  seconds. The simulated voltage, obtained after the optimisation process, is shown in Figure 2.9.



**Figure 2.9:** Pulse Test voltage tuning A

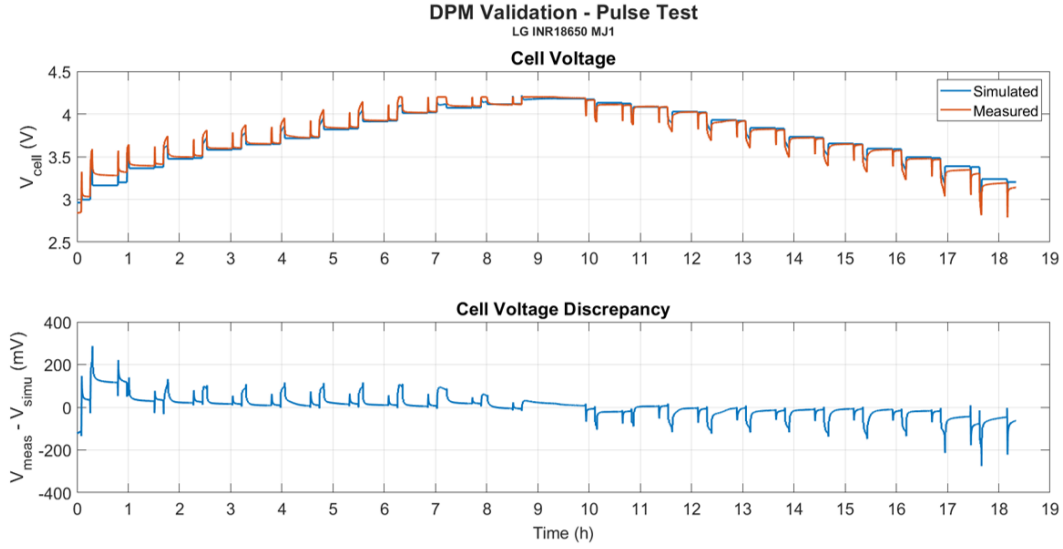
Analysing this voltage profile, several important characteristics can be observed. The first consideration regards the initial simulated voltage value. Indeed, since the validation process was initiated from the first available time, the initial voltage value was imposed equal to the OCV evaluated at zero SOC. This assumption, valid for all other proposed tunings, was made by considering the null initial current, thus making OCV a coherent starting condition and avoiding an unfeasible value.

The second feature concerns the root mean square error (RMSE), evaluated by Equation 2.7, which is equal to 84.2 mV. This value reflects the strong deviation of the simulated voltage from the measured one, especially in the charging phase. This discrepancy arises mainly due to the OCV curve resulting underestimated with the *Capacity Test* at low SOC. To overcome this issue, the second tuning was set.

$$V_{RMSE} (V) = \sqrt{\frac{1}{N} \cdot \sum_{k=1}^N (V_{cell,k}^{Sim} - V_{cell,k}^{Meas})^2} \quad (2.7)$$

### 2.4.2 Tuning B

The second tuning, associated with letter B, was defined by imposing the same ranges for all the resistors and time constants of the previous tuning. The key difference was in the OCV curve, which was allowed to vary during the optimisation process. The only constraint to be respected was the compliance with the lower and upper voltage limits specified in the datasheet (Table 2.2). Consequently, the OCV was limited between 2.5 V and 4.2 V. The simulated *Pulse Test* voltage is reported in Figure 2.10.



**Figure 2.10:** Pulse Test voltage tuning B

By observing the proposed voltage profiles, it is clearly evident that, considering the OCV curve as a tunable parameter, an improvement in the quality of the DPM is achieved. Indeed, compared to the previous tuning, the simulated voltage is closer to the measured one, not only during the resting period but also during the charging and discharging pulses. This improved behaviour is due to the higher OCV values at low SOC obtained during the tuning process. However, according to Equation 2.1, this causes a slight deviation at the end of the discharge phase where the simulated voltage exceeds the measured one in contrast to Tuning A. Despite this, the second tuning process ensured a reduction of the voltage RMSE ( $V_{RMSE}$ ) up to  $47.8\text{ mV}$ , corresponding to a decrease of 43.2% compared to Tuning A.

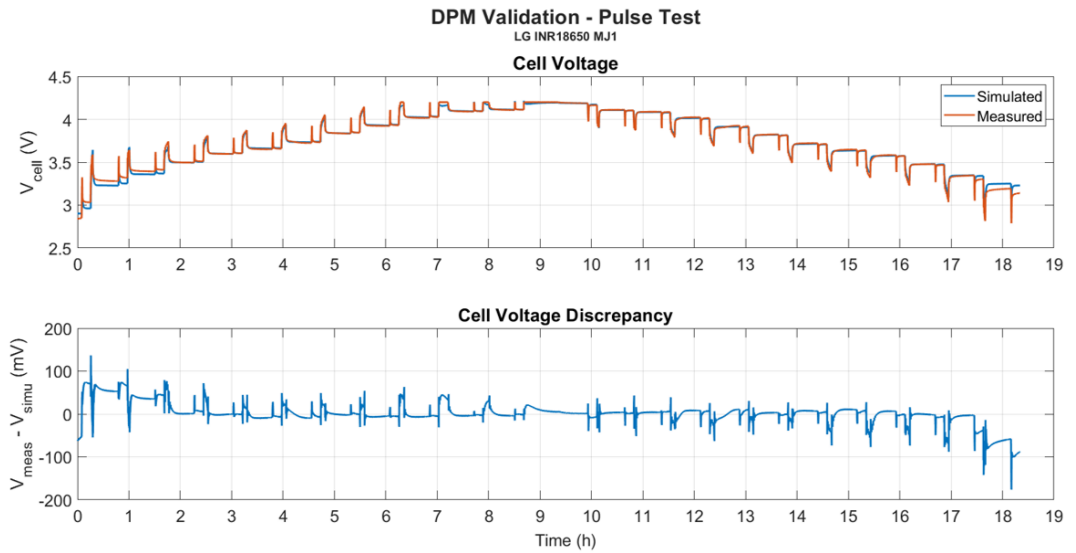
### 2.4.3 Tuning C

The third tuning, denoted by letter C, was derived after an analysis of the influence of each parameter on the simulated voltage. The outcome of this investigation revealed that, for each resistive parameter and at each SOC level, specific lower and upper bounds must be defined. The admissible intervals for the optimisation process, reported in Table 2.4, were established based on the most frequent values extracted from the conducted sensitivity analysis. Instead, for what concerns the OCV, it was allowed to be tuned as in the previous case (Tuning B), while the first time constant spanned from 0 to 500 seconds, and the second between 500 to 10000 seconds.

As in the previous cases, the resulting voltage profile is illustrated below, in Figure 2.11.

| Parameter           | Limit | SOC Level (%) |     |    |         |    |    |     |
|---------------------|-------|---------------|-----|----|---------|----|----|-----|
|                     |       | 0             | 5   | 10 | 15 ÷ 85 | 90 | 95 | 100 |
| $R_0$ (m $\Omega$ ) | Lower | 45            | 40  | 35 | 32      | 35 | 35 | 40  |
|                     | Upper | 60            | 55  | 50 | 38      | 40 | 45 | 50  |
| $R_1$ (m $\Omega$ ) | Lower | 60            | 40  | 35 | 5       | 5  | 5  | 10  |
|                     | Upper | 100           | 80  | 60 | 30      | 30 | 30 | 35  |
| $R_2$ (m $\Omega$ ) | Lower | 100           | 40  | 30 | 30      | 35 | 35 | 60  |
|                     | Upper | 300           | 100 | 80 | 60      | 60 | 80 | 100 |

**Table 2.4:** Resistance limits for optimisation



**Figure 2.11:** Pulse Test voltage tuning C

By examining Figure 2.11 and comparing it to the previous ones, it is immediately apparent that the current tuning produces the most accurate DPM parameters estimation for describing the cell behaviour. As a matter of fact, the simulated voltage is almost overlapped during whole test, even in the highly dynamic voltage variations caused by the current pulses. This effect translates into a voltage RMSE reduction by 46.7% compared to Tuning B, obtaining a final value of just 25.5 mV.

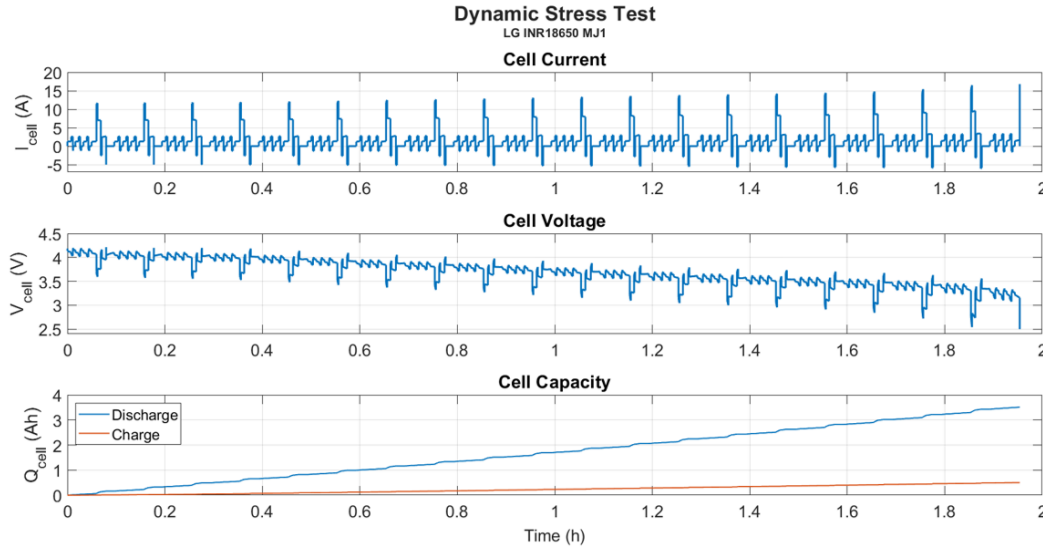
Table 2.5 summarises the voltage RMSE obtained from the three proposed tuning processes considering the entire test duration. Moreover, it underlines the importance of proper input definition to the *Matlab* script to effectively guide the SDO towards the optimal solution.

| Tuning Code | $V_{RMSE}$ (mV) | $V_{RMSE}$ reduction (%) |
|-------------|-----------------|--------------------------|
| A           | 84.2            | /                        |
| B           | 47.8            | 43.2                     |
| C           | 25.5            | 46.7                     |

**Table 2.5:** Voltage RMSE for different tunings

## 2.5 DPM Validation

To effectively consider Tuning C as the optimal parameters estimation for modelling the cell, it was necessary to validate the DPM on a more dynamic and aggressive test than the one adopted for the characterisation. Specifically, the test considered, available in the database and reported in Figure 2.12, is named *Dynamic Stress Test*. The term *Dynamic* can be visualised in the current profile which exhibits frequent alternations of charging and discharging phases at different C-rates. Consequently, the voltage, related to the current by Equation 2.1, appears extremely irregular and sawtoothed.



**Figure 2.12:** Dynamic Stress Test

It is important to highlight that, in order to properly validate the model, the test data, sampled at variable time steps, were first converted to fixed time steps. In particular, to capture even the smallest voltage and current variations, an extremely small time interval (1 ms) was adopted. The result of this conversion process, performed with the *Matlab* script 2.1, is depicted in Figure 2.13.

```

1 disTime = 0:0.001:contTime(end)
2

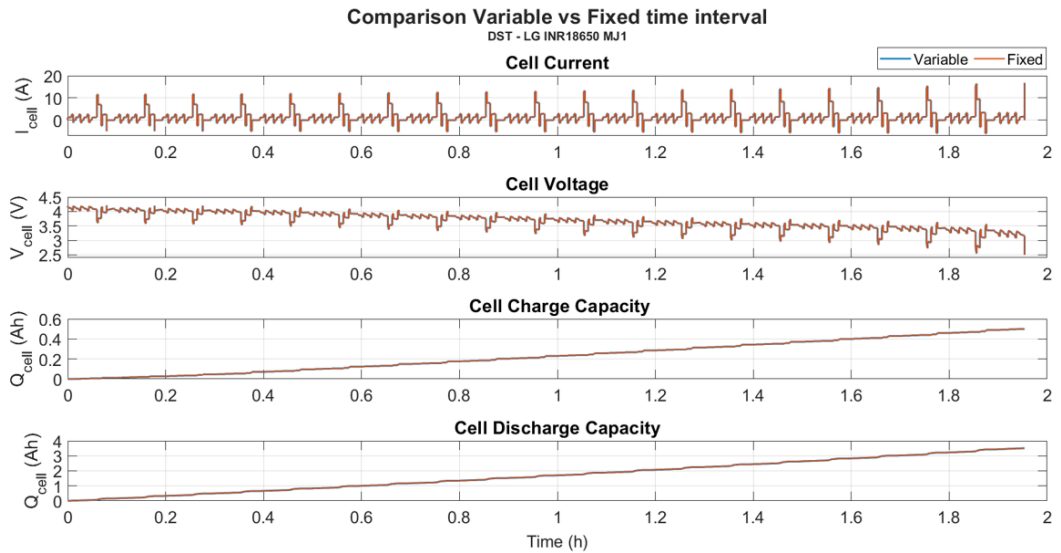
```

```

3 stringMethods = ["Linear", "Makima","Nearest"];
4 selectMethod = stringMethods(1);
5
6 disCurrent = interp1(contTime, contCurrent,
    disTime,selectMethod,"extrap");
7 disVoltage = interp1(contTime, contVoltage,
    disTime,selectMethod,"extrap");
8 disCapCharge = interp1(contTime, contChCap,
    disTime,selectMethod,"extrap");
9 disCapDischarge = interp1(contTime, contDischCap,
    disTime,selectMethod,"extrap");

```

**Listing 2.1:** DST fixed time conversion



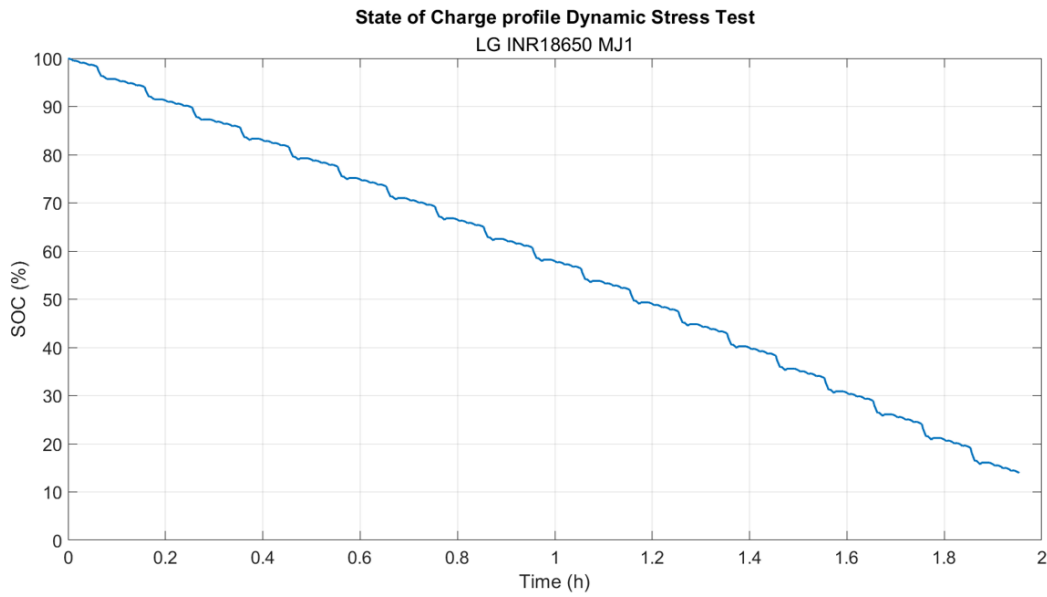
**Figure 2.13:** DST variable and fixed time interval

As noticeable, for all the electrical quantities shown, the curves representing the variable sampling time data are not visible, as they are perfectly overlapped by the fixed ones.

After this preliminary step, the SOC profile during the test was evaluated, by Equation 2.8, as the difference between the previous SOC ( $SOC_{DST}(k-1)$ ) and the ratio of net ( $Q_{net}$ ) to nominal ( $Q_{nom}$ ) capacity. The first element in the Equation was initially imposed equal to 1, corresponding to 100%, in accordance with the database experiment setup. In fact, before being subjected to the *Dynamic Stress Test*, the cell was completely charged with a CC phase at 1C until the upper voltage limit was reached. Subsequently, a CV phase was carried out until the current dropped to zero [6].

$$\begin{cases} SOC_{DST}(k) = SOC_{DST}(k-1) - \frac{Q_{net}(k)}{Q_{nom}} \\ Q_{net}(k) = Q_{Discharge}(k) - Q_{Charge}(k) \end{cases} \quad (2.8)$$

The resulting SOC curve throughout the DST is shown in Figure 2.14. The profile highlights the dynamic nature of the test with alternations of discharging and charging phases. For this reason, the DST was suitable for the validation process because, if the parameters had been accurately estimated, the model would have been able to replicate the cell behaviour by following the measured voltage.



**Figure 2.14:** SOC profile during DST

The evaluation of this variable proved to be crucial as it was adopted to calculate the resistances and time constants values of the DPM over the entire test. Precisely, this operation was carried out in *Matlab* adopting the `interp1` function, with the reference values for the interpolation taken from the results obtained with Tuning C.

```

1 %SOCbrk = 0:0.05:1, discretised at 5% SOC
2 %NameParameter_TuningC = outcome profile of the
   optimisation process for each parameter (array 1x21)
3 %SOC_DST = value calculated with Equation 2.8
4 %NameParameter_DST = vector of parameters during DST
5
6 R0_DST = interp1(SOCbrk, R0_TuningC, SOC_DST) %[ohm]
```

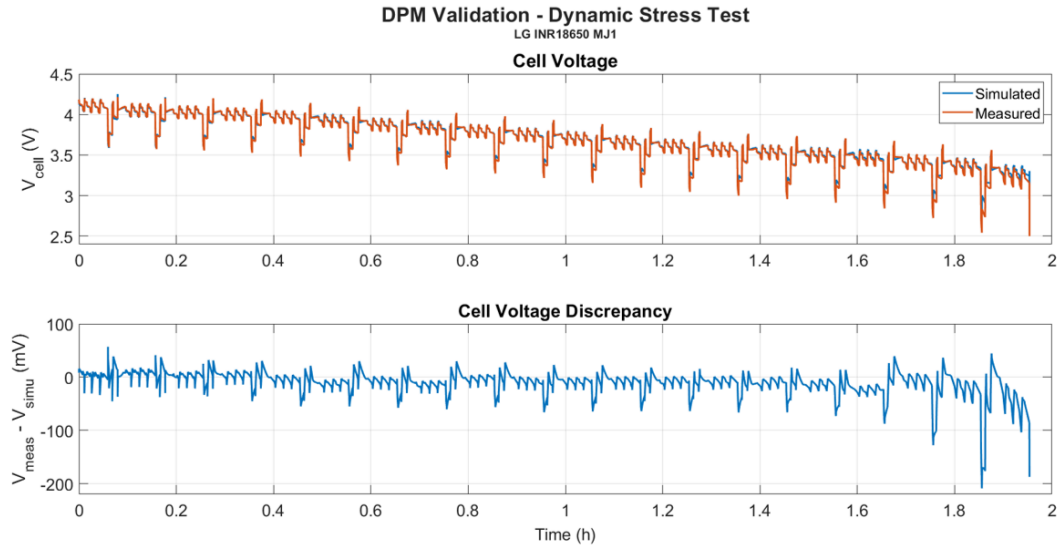
```

7  R1_DST = interp1(SOCbrk, R1_TuningC, SOC_DST) %[ohm]
8  R2_DST = interp1(SOCbrk, R2_TuningC, SOC_DST) %[ohm]
9  tau1_DST = interp1(SOCbrk, tau1_TuningC, SOC_DST) %[s]
10 tau2_DST = interp1(SOCbrk, tau2_TuningC, SOC_DST) %[s]

```

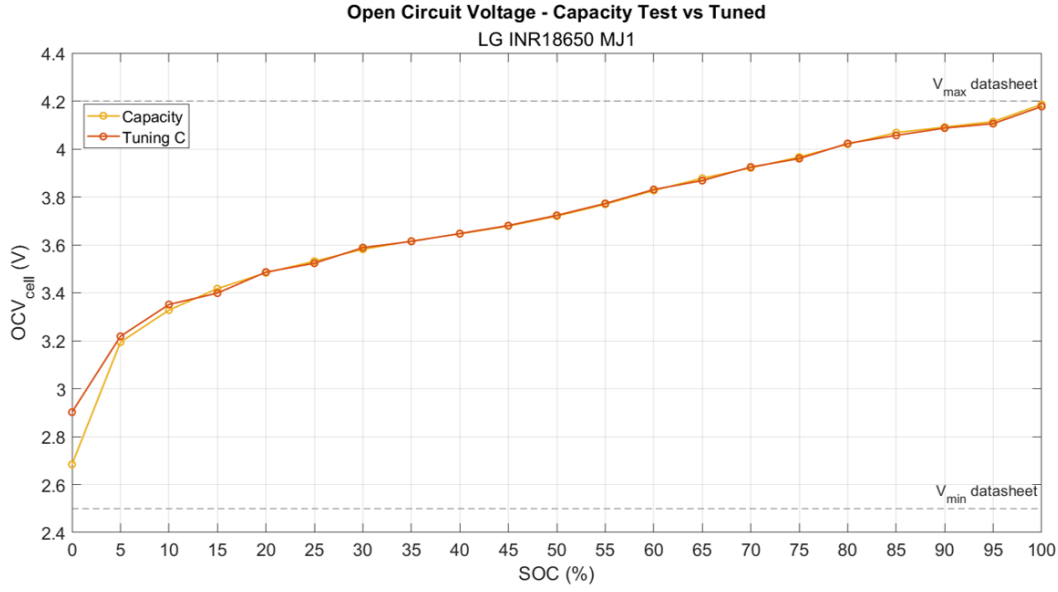
**Listing 2.2:** DST parameter estimation

These model parameters were inserted into the set of Equations 2.1 to evaluate the cell voltage, illustrated in Figure 2.15, at any millisecond of the test. As visible, for the entire test duration, the blue curve, representing the simulated voltage, accurately tracks the measured one, confirming the high quality estimation achieved with Tuning C. The only drawback concerns the reliability of the parameters at low SOC where, as in Figure 2.11, the largest deviations occur. Nevertheless, the voltage RMSE, calculated with Equation 2.7, is only 23.8 mV, thus establishing Tuning C as the proper candidate for the cell characterisation.



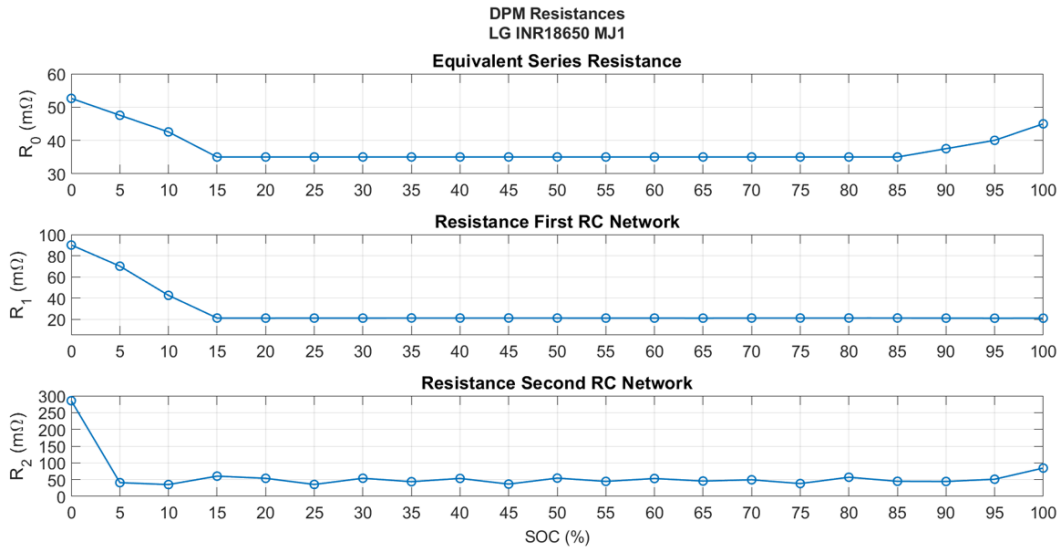
**Figure 2.15:** DPM validation - DST cell voltage

Once the validation stage was completed, the behaviours of the tuned DPM parameters as a function of SOC were examined. The first element analysed was the OCV curve, which, even if allowed to be tuned, was required to maintain an increasing trend consistent with what was extracted from *Capacity Test*. Figure 2.16 compares the OCV profile obtained during the optimisation process (red) with the averaged one from the *Capacity Test* (yellow). As noticeable, the optimised OCV curve respects the constraints on shape and trend but, coherently with what was discussed in Tuning B, it appears higher at low SOC where the *Capacity Test* underestimated the values. On the contrary, at high SOC an overestimation was achieved during the test. Instead, in the usable cell range (20% ÷ 80% SOC) the two curves perfectly match.



**Figure 2.16:** Tuned OCV curve vs. SOC

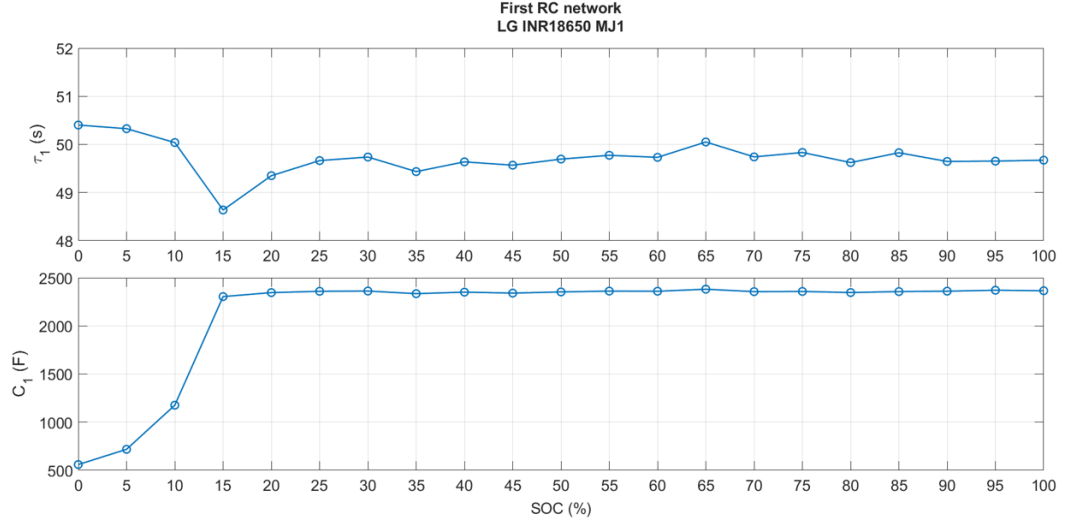
Afterwards, the attention was shifted to the profiles of the tuned DPM resistances ( $R_0, R_1, R_2$ ), which are illustrated in Figure 2.17. As visible, all three curves exhibit a *U*-shaped profile with an increase at low and high SOC and a plateau in the usable cell range. This behaviour is consistent with the outcomes found in the literature [19, 20, 21, 22].



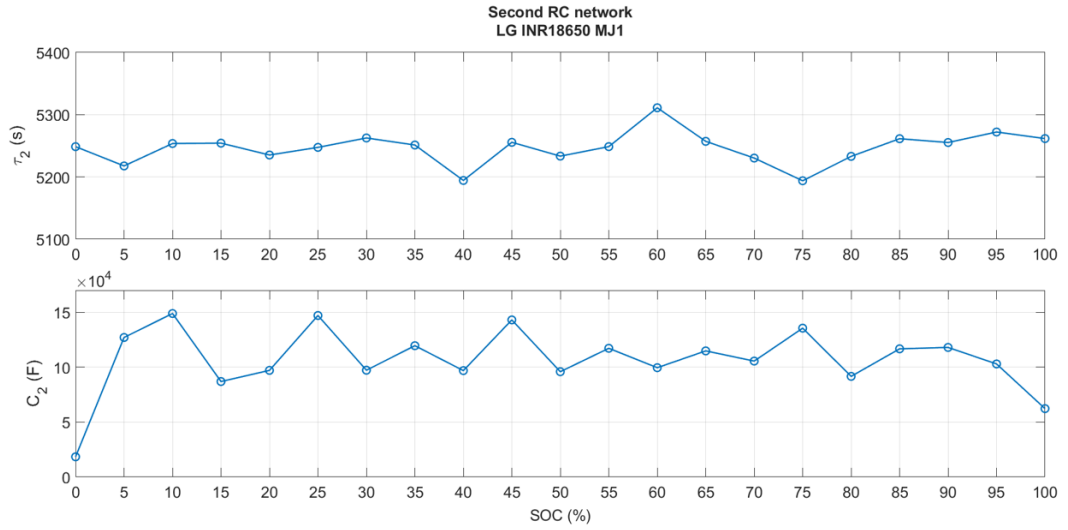
**Figure 2.17:** Tuned DPM resistances vs. SOC

Finally, the tuned profiles of the two time constants were examined and adopted to compute the curves of the two capacitive elements of the DPM. This calculation was performed, at each SOC level, by dividing each time constant by its corresponding resistance ( $C_i = \tau_i / R_i$ ). Figure 2.18 displays the results for both RC networks, with the first depicted in the upper part and the second

in the lower one. Both tuned capacitive elements exhibit an increasing trend at low SOC followed by a more stable region for  $C_1$  and a more oscillatory behaviour for  $C_2$ . These resulting profiles are consistent, as for the resistances, with the literature [19, 22].



(a) Time constant and capacitor first RC network vs. SOC



(b) Time constant and capacitor second RC network vs. SOC

**Figure 2.18:** Tuned DPM time constants and capacitors vs. SOC



# Chapter 3

## Extended Kalman Filter

The chapter aims to explain the theory behind the Extended Kalman Filter for battery or cell state estimation, along with its implementation and validation in *Simulink* environment.

### 3.1 Theoretical Background

The EKF, as discussed in section 1, proves to be the most suitable method for estimating the SOC in the context of this thesis, as it provides an optimal balance between accuracy and computational effort [16].

The EKF, an extension of the Linear Kalman Filter (KF), is a recursive algorithm adopted for non-linear systems to estimate internal state variables, not directly measurable, such as the cell SOC. The main assumption of the filter is related to the noises assumed to be described by uncorrelated Gaussian distributions with zero mean and known covariance matrices. The operating principle consists of predicting the evolution of the state through a mathematical model describing the system, followed by a correction based on the measurements acquired by the sensors. This update phase is fundamental as it balances, through the Kalman Gain, the uncertainties or noises associated with both the model and the measurements, ensuring the most accurate probabilistic estimate of the system state. The described process remains the same for both KF and EKF, the key difference lies in the handled systems. Indeed, while the KF can be applied only to linear systems represented in state-space with constant matrices, the EKF is able to manage non-linear systems, such as the cell model employed in this work, by linearising them at each time step [23, 24]. Before presenting the equations adopted for the development of the EKF, some important notations must be introduced:

- The superscript “ - ” indicates a predicted quantity based only on past measurements;

- The superscript “+” indicates an estimated quantity based on both past and present measurements;
- The hat “^” on a variable indicates either a predicted or estimated value of that variable. For instance  $\hat{x}^-$  represents the predicted states, while  $\hat{y}$  the predicted output;
- The tilde “~” on a variable indicates an error evaluated as the difference between the true value and the predicted or estimated one. For example  $\tilde{x}^- = x - \hat{x}^-$  is the error of the predicted state, while  $\tilde{y} = y - \hat{y}$  is the output error, denoted as innovation term [23];
- The variable “P” represents the correlation between the two arguments in its subscript:

$$P_{xy} = \mathbb{E}[xy^T] \quad \text{and} \quad P_{xx} = \mathbb{E}[xx^T] \quad (3.1)$$

### 3.1.1 Six-step Process

The EKF principle operates on six distinct steps, which are presented below with a brief generic description and a more detailed focus on the current application. It is important to recall, as described in section 2.4, the state of the DPM adopted for the cell description [16, 21, 23]:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} SOC \\ V_{RC_1} \\ V_{RC_2} \end{bmatrix}$$

#### Phase 0: Initialisation

This stage is aimed at setting, for the first time instant, the initial state values ( $\hat{x}_0^+$ ) and the associated covariance matrix ( $P_{\hat{x}_0}^+$ ). This matrix represents the uncertainty level of this initial estimation and, as will be demonstrated later, its proper definition is crucial since it affects the behaviour of the filter.

#### Phase 1: Prediction

This phase constitutes one of the two main steps in the described filter’s working principle and is structured as follows:

##### 1.a) State Prediction

The filter predicts, through the mathematical model describing the system dynamics, the state at the current time step ( $k$ ) based on measurements and estimated state at the previous time step ( $k-1$ ). Equation 3.2 defines

the general formulation of this phase, where  $f$  is the non-linear function describing the state evolution,  $u$  is the input applied to the system, and  $w$  is the process noise characterised, as previously explained, by zero mean ( $\bar{w}$ ) and covariance  $Q_x$ :

$$\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}, \bar{w}_{k-1}) \quad (3.2)$$

It is important to highlight that, in this thesis, the function  $f$  was linearised around the previous estimated state every time step, thus the state evolution in discrete time was described by the first Equation in 2.6 and the second in Equation 2.1. As a consequence, it was possible to express the predicted state in state-space form with time-varying matrices:

$$\hat{x}_k^- = \hat{A}_{k-1} \cdot \hat{x}_{k-1}^+ + \hat{B}_{k-1} \cdot u_{k-1} \quad (3.3)$$

$$\hat{A}_{k-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \exp\left(-\frac{dt}{\tau_1}\right) & 0 \\ 0 & 0 & \exp\left(-\frac{dt}{\tau_2}\right) \end{bmatrix} \quad (3.4)$$

$$\hat{B}_{k-1} = \begin{bmatrix} -\frac{dt}{Q_{nom}} \\ R_1 \left(1 - \exp\left(-\frac{dt}{\tau_1}\right)\right) \\ R_2 \left(1 - \exp\left(-\frac{dt}{\tau_2}\right)\right) \end{bmatrix} \quad (3.5)$$

In the presented equations, the system input  $u$  was represented by the current applied to the cell, considered positive during discharge and negative in charge. Instead, all the DPM parameters were SOC dependent in accordance with the results of Tuning C.

### 1.b) Error Covariance Prediction

After the system state prediction, the accuracy level is evaluated by combining the error covariance matrix at previous time step, the transition matrix  $\hat{A}$ , and the process noise covariance as follows:

$$P_{\tilde{x}_k}^- = \mathbb{E}[(\tilde{x}_k^-) \cdot (\tilde{x}_k^-)^T] = \hat{A}_{k-1} \cdot P_{\tilde{x}_{k-1}}^+ \cdot \hat{A}_{k-1}^T + Q_x \quad (3.6)$$

The proposed formulation, typically adopted in KF, results valid in the current application since the state evolution was linearised.

### 1.c) System Output Prediction

The filter predicts the system output value  $y_k$ , which represents the expected measurement, based on the mathematical model describing its evolution and the predicted state. Equation 3.7 describes the general formulation of this phase, where  $h$  is the non-linear function modelling the output evolution,  $u$  is the system input and  $v$  is the measurement noise characterised by zero mean ( $\bar{v}$ ) and covariance  $R$ :

$$\hat{y}_k = h(\hat{x}_k^-, u_k, \bar{v}_k) \quad (3.7)$$

In the case under examination, the system output was represented by the cell voltage and described in discrete time by first Equation in 2.1 with the introduction of the measurement noise:

$$\hat{y}_k = V_{cell}(k) = OCV(\hat{x}_{1,k}^-) - \hat{x}_{2,k}^- - \hat{x}_{3,k}^- - R_0 \cdot u_k + \bar{v}_k \quad (3.8)$$

As clearly evident, due to the dependency exhibited by the OCV on the SOC, it was not possible to express it in state-space form. In fact, while the state evolution function  $f$  was expressed in state space form, the output voltage equation  $g$  was maintained in explicit form.

## Phase 2: Correction or Innovation

This phase represents the second main step in the working principle of the filter and is subdivided as follows:

### 2.a) Kalman Gain Evaluation

The Kalman gain, a fundamental component of the filter, represents a weighting matrix determining the balance between the predicted state, evaluated through the system model, and the observed measurements. It is adopted to adjust the predicted state in order to minimise the mean square estimation error, ensuring the most accurate estimation of the system state. Equation 3.9 defines the general formulation of the Kalman gain, where  $P_{\tilde{x}_k, \tilde{y}_k}^-$  is the cross-covariance matrix between state error and measurement error, and  $P_{\tilde{y}_k}$  is the measurement error covariance matrix:

$$K_k = P_{\tilde{x}_k, \tilde{y}_k}^- \cdot P_{\tilde{y}_k}^{-1} \quad (3.9)$$

In order to apply this Equation to the proposed application, several mathematical steps were necessary to evaluate the individual components. Starting from the second element, it was possible to express the output error, in linearised Jacobian matrices, by adopting a truncated Taylor-

series expansion at first order of  $y_k$  around the set point  $q_k = \{\hat{x}_k^-, u_k, \bar{v}_k\}$ :

$$\tilde{y}_k = y_k - \hat{y}_k \approx \hat{C}_k \cdot \hat{x}_k^- + \hat{D}_k \cdot \tilde{v}_k \quad (3.10)$$

This allowed to determine the measurement error covariance matrix:

$$P_{\tilde{y}_k} \approx \hat{C}_k \cdot P_{\tilde{x}_k}^- \cdot \hat{C}_k^{-1} + \hat{D}_k \cdot R \cdot \hat{D}_k^T \quad (3.11)$$

$$\hat{C}_k = \frac{\partial h(\cdot)}{\partial \hat{x}_k^-} = \left[ \frac{\partial OCV}{\partial SOC} \bigg|_{\hat{x}_{1k}^-} \quad -1 \quad -1 \right] \quad (3.12)$$

$$\hat{D}_k = \frac{\partial h(\cdot)}{\partial \hat{v}_k} = [1] \quad (3.13)$$

The derivative of the OCV with respect to the SOC was modelled through a 1D LUT following the method proposed by Plett. Therefore, the reference values adopted for the interpolation were evaluated as:

```

1      %ocvVal = row vector obtained with Tuning C
2      %dSOC = discretised SOC, equal to 5%
3      %dOCVdSOC = reference value for interpolation
4
5      dOCVdSOC = ( ocvVal(3:end) - ocvVal(1:end-2) )
6      ./ ( 2 * dSOC );
7      dOCVdSOC = [ ( ocvVal(2) - ocvVal(1) )/dSOC
      dOCVdSOC      (ocvVal(end) - ocvVal(end-1))/dSOC ];

```

**Listing 3.1:** OCV vs. SOC derivative

Instead, regarding the first term of the Kalman gain, it was derived by exploiting the assumption of independence between measurement noise and predicted state, thereby obtaining:

$$P_{\tilde{x}_k, \tilde{y}_k}^- = P_{\tilde{x}_k}^- \cdot \hat{C}_k^T \quad (3.14)$$

Combining Equations 3.14 and 3.11, the Kalman Gain for SOC estimation was expressed as:

$$K_k = P_{\tilde{x}_k}^- \cdot \hat{C}_k^T \cdot [\hat{C}_k \cdot P_{\tilde{x}_k}^- \cdot \hat{C}_k^T + R]^{-1} \quad (3.15)$$

## 2.b) State Estimation

In this step, the filter updates and corrects the predicted state knowing

the acquired measurement from the sensors ( $y_k$ ) and the Kalman gain:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \cdot \tilde{y}_k = \hat{x}_k^- + K_k \cdot (y_k - \hat{y}_k) \quad (3.16)$$

It is important to highlight that, in some cases, as proposed by Plett, the SOC ( $\hat{x}_1^+$ ) is limited to avoid unfeasible values. However, in this thesis, no constraints were imposed on the estimated SOC since these filter results will be subsequently adopted for vehicle's control where the proper saturations will be implemented.

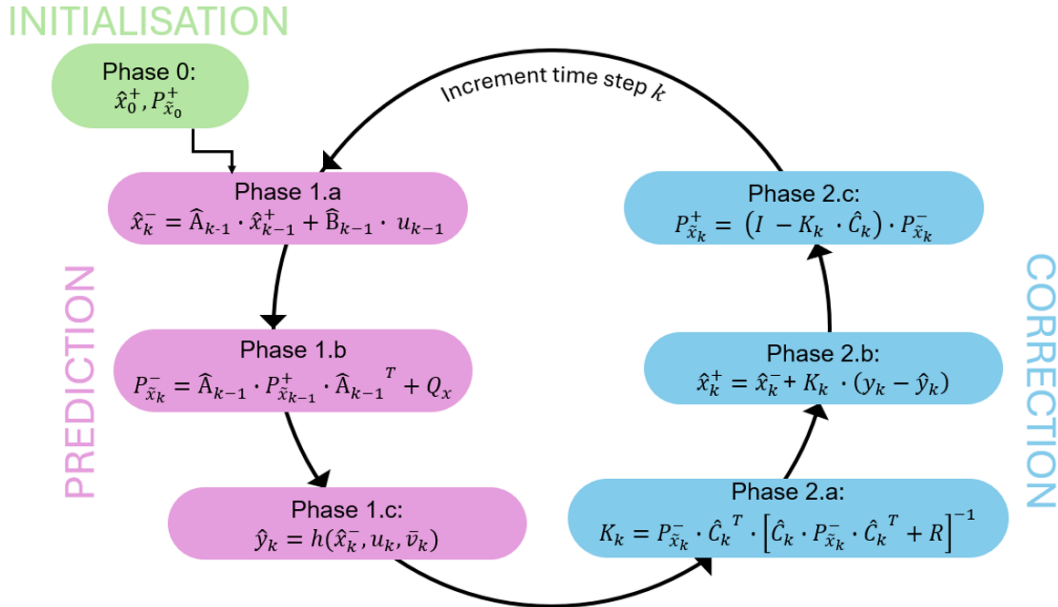
### 2.c) Error Covariance Estimation

The last step regards the update of the state covariance error matrix performed by Equation 3.17:

$$P_{\hat{x}_k}^+ = (I - K_k \cdot \hat{C}_k) \cdot P_{\hat{x}_k}^- \quad (3.17)$$

As visible, the state correction with the measurement leads to a reduction in the uncertainty level. Indeed, due to the Kalman gain contribution, the estimated error covariance matrix is lower than the predicted one.

Figure 3.1 summarises the operating principle of the filter, including the main steps and equations employed in its implementation:



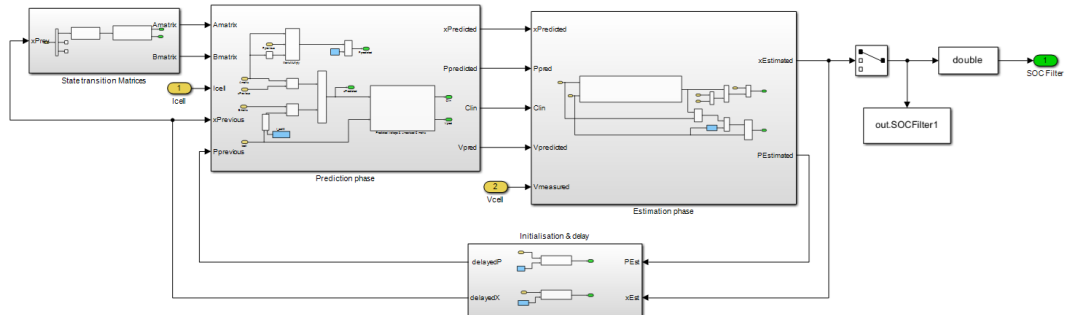
**Figure 3.1:** EKF operating principle

## 3.2 Simulink Implementation - Single Cell

The EKF for SOC estimation was first implemented in *Simulink* environment for a single cell following the theoretical formulation introduced in the previous section. Specifically, to facilitate future code generation and maintain a clear structure, a subsystem approach was adopted. Each subsystem was responsible for a specific part of the filter working principle and embedded in a main *EKF* block. This external subsystem receives as inputs the current applied to the cell ( $u$ ) and the measured voltage ( $y$ ) and outputs the estimated SOC ( $\hat{x}^+$ ). Additionally, a mask was created on this *EKF* block to easily configure all the parameters involved in the filter Equations. Table 3.1 reports the mask parameters while Figure 3.2 shows the subsystems inside the main *EKF* block:

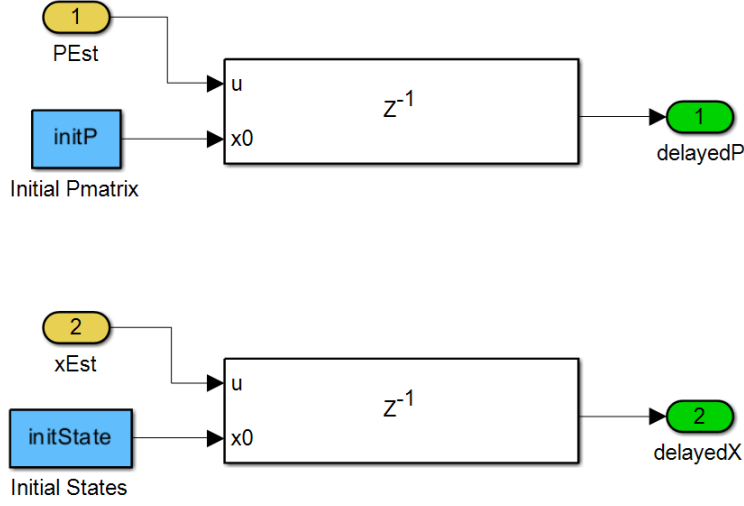
| Parameter                       | Variable Name | Dimensions |
|---------------------------------|---------------|------------|
| Initial state                   | initState     | 3x1        |
| Initial error covariance matrix | initP         | 3x3        |
| Process noise matrix            | Q             | 3x3        |
| Measurement noise matrix        | R             | 1x1        |
| Initial current                 | initCellCurr  | 1x1        |
| Cell nominal capacity           | nomCapAh      | 1x1        |
| SOC breakpoints                 | SOCbrk        | 1x21       |
| LUT values $R_0$                | R0val         | 1x21       |
| LUT values $R_1$                | R1val         | 1x21       |
| LUT values $R_2$                | R2val         | 1x21       |
| LUT values $\tau_1$             | tau1val       | 1x21       |
| LUT values $\tau_2$             | tau2val       | 1x21       |
| LUT values OCV                  | ocvVal        | 1x21       |
| LUT values OCV derivative       | dOCVdSOC      | 1x21       |

**Table 3.1:** *EKF* mask parameters



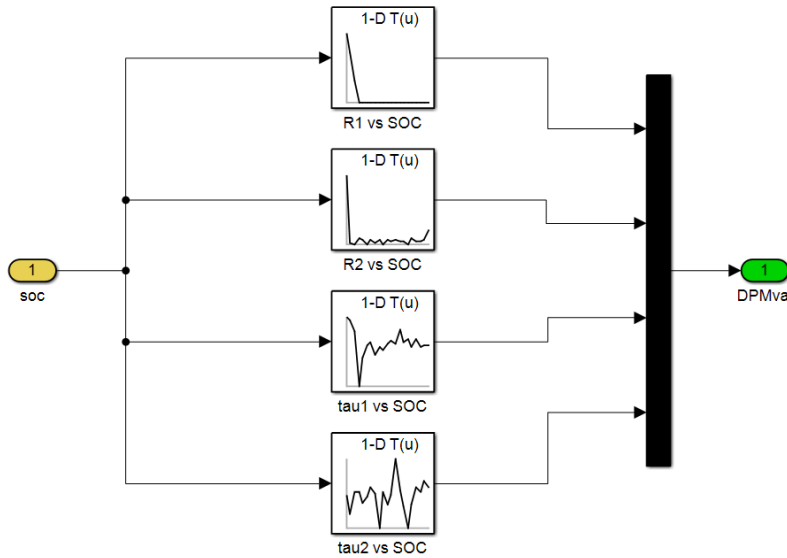
**Figure 3.2:** Single cell *EKF* subsystems overview

The estimation process starts from the lower block of Figure 3.2, denoted as *Initialisation & Delay*, responsible for setting the initial state vector and the associated error covariance matrix at the first simulation step.

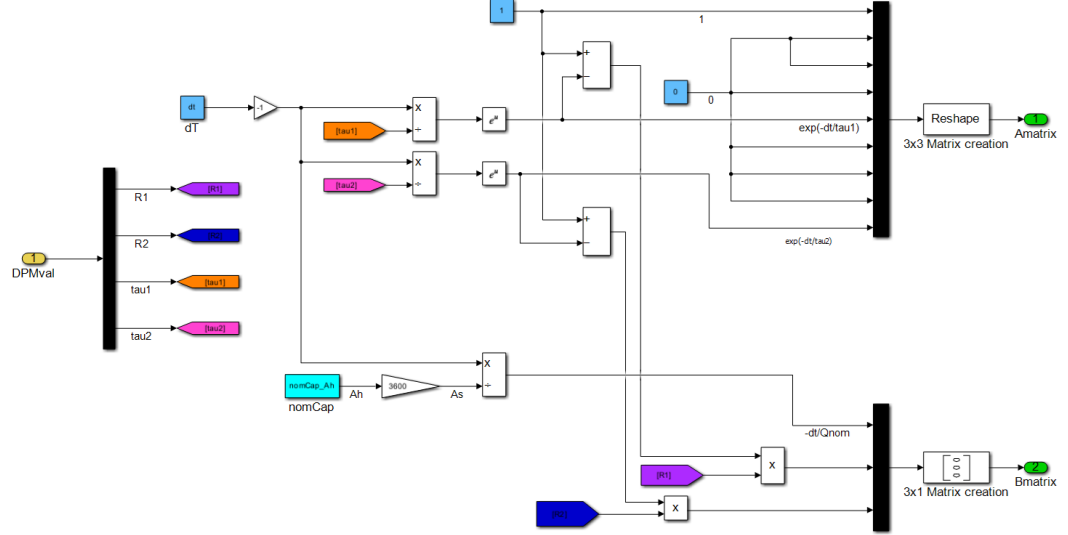


**Figure 3.3:** Initialisation & Delay subsystem

Afterwards, the initial state vector is passed to the upper left block, labelled *State Transition Matrices*, where it is adopted to extrapolate the DPM cell parameters through 1D LUT with reference values of Tuning C. These parameters are used to extract the two linearised transition matrices  $\hat{A}_{k-1}$  and  $\hat{B}_{k-1}$ . Figure 3.4 depicts the implementation of the described process:



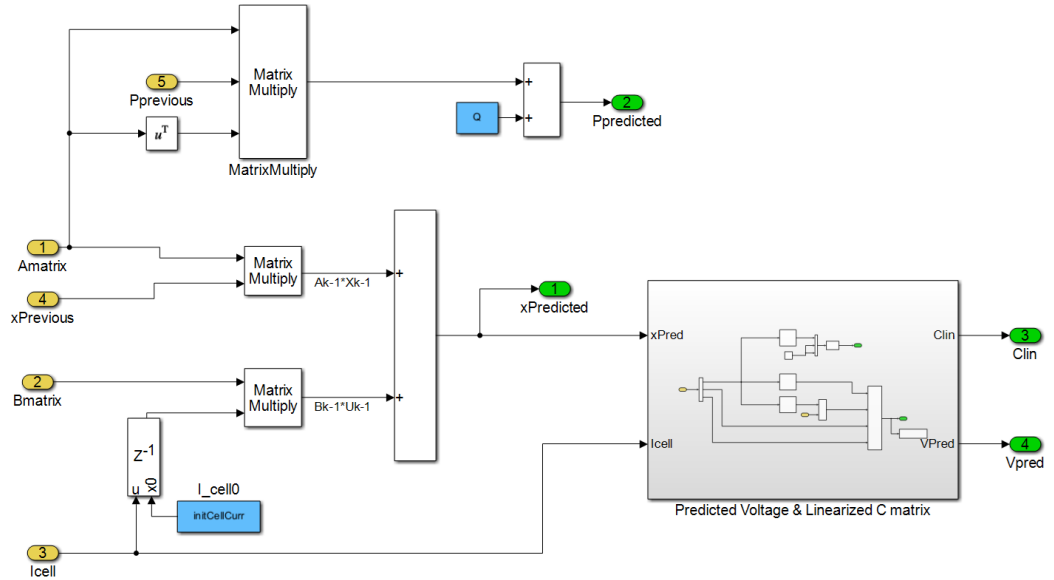
**(a)** DPM parameters at previous SOC



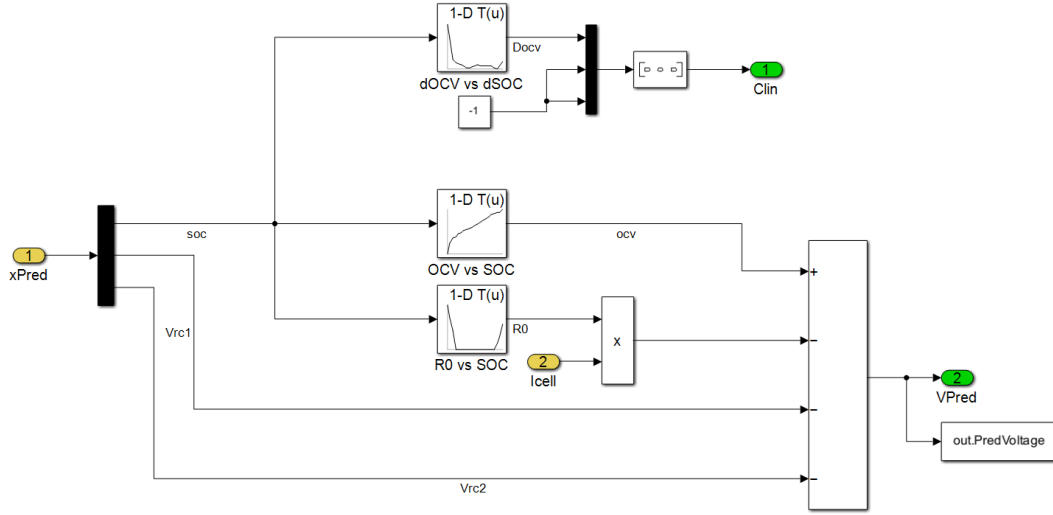
(b) Matrices creation

**Figure 3.4:** State Transition Matrices subsystem

Subsequently, the two evaluated matrices, along with the initial state vector and the error covariance matrix, are inserted into the *Prediction phase* subsystem, where steps 1.a, 1.b, and 1.c of the six-step process are executed. The upper part of Figure 3.5 illustrates the evaluation of the predicted state and the error covariance matrix associated, while the bottom part shows the cell voltage prediction and the linearised output matrices.



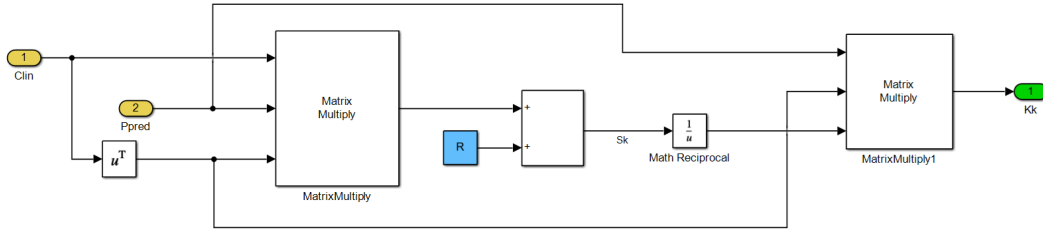
(a) State and covariance matrix prediction



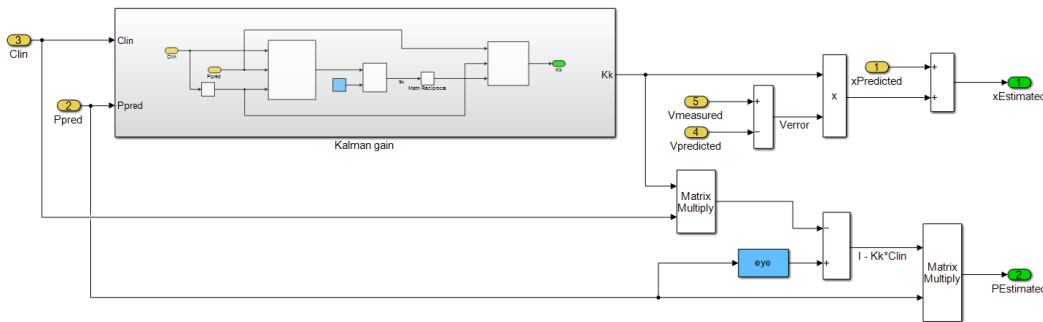
(b) Cell voltage prediction and linearised output matrices

**Figure 3.5:** Prediction phase subsystem

Finally, the estimation phase starts with the computation of the Kalman gain, as illustrated in Figure 3.6a, followed by the correction of the state and the error covariance matrix (Figure 3.6b).



(a) Kalman gain



(b) State and covariance matrix estimation

**Figure 3.6:** Estimation phase subsystem

The entire process is iteratively repeated over the entire test duration, adopting the previous state, required for the DPM parameter extraction and

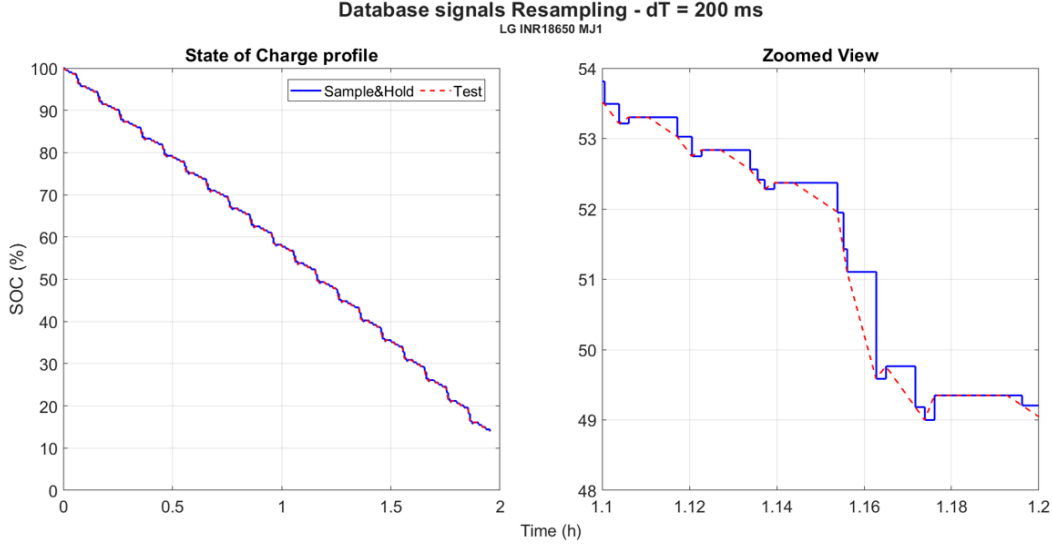
prediction phase, equal to the output of the *Initialisation & Delay* subsystem. The latter represents the estimated state delayed of one time step as reported in Equation 3.3.

It is important to highlight that all the LUTs adopted in the *Simulink* model were configured with the `clip` interpolation method. This setting was applied in order to saturate the output value to the nearest boundary whenever the SOC exceeds the upper (100%) and lower (0%) breakpoint limits, preventing unrealistic outputs.

### 3.2.1 Data Resampling

To enable future C code deployment of the proposed *Simulink* implementation and ensure compatibility with the available experimental instrumentation, the original database signals were resampled at a fixed time interval of 200 *ms*. This procedure was carried out in *Matlab* environment for both *Pulse* and *Dynamic Stress* test, and was applied to all relevant electrical quantities, including cell current, voltage, charge, and discharge capacities. It is important to note that this resampling process differs from the one previously proposed in Code 2.1. While the latter was conceived to approximate the test behaviour as accurately as possible, the present method was developed to implement a sample-and-hold (S&H) mechanism, simulating fixed-time acquisition of real instrumentation. The *Matlab* code adopted to perform this resampling was similar to the previous one (Listing 2.1) with two main differences. First, the time interval was adjusted from 1 *ms* to 200 *ms*, second the interpolation method was changed to `previous` in order to obtain the zero S&H mechanism.

Figure 3.7 illustrates the SOC profile during the DST, evaluated by Equation 2.8. The red dashed line corresponds to the SOC derived from the raw capacities of the database, whereas the blue line represents the SOC profile obtained after the resampling process. As visible in the full-scale view on the left, the two curves appear perfectly overlapped. However, analysing the right plot, which is a zoomed view, an important difference emerges. The resampled signal maintains a constant value over the time interval imposed before being updated, introducing a stair-wise behaviour. Despite this, the resulting profile still closely traces the original trend, thus representing the overall cell behaviour throughout the entire test duration.



**Figure 3.7:** Resampling process at fixed time interval

The aforementioned considerations, although focused only on the SOC profile during the DST, equally apply to all other electrical quantities and to both experimental tests.

### 3.2.2 Filter Tuning

As previously mentioned, one of the key steps in the filter implementation was the tuning. Particularly, it was necessary to define four quantities: the initial state  $\hat{x}_0^+$ , the associated error covariance matrix  $P_{\hat{x}_0}^+$ , the process noise covariance matrix  $Q_x$ , and the measurement noise covariance matrix  $R$ .

The first parameter, represented by a column vector, was the easiest to be defined. For both experimental tests previously described, the initial SOC as well as the voltage drop across the two RC networks were already known. Specifically, the voltages  $V_{RC1}$  and  $V_{RC2}$  were always set equal to zero since, in both tests, the initial current value was zero, while the SOC was imposed equal to 0% for the *Pulse Test* and to 100% for the *DST*.

The second parameter was more complex to be defined as the uncertainty level associated with the state initialisation was not physically measurable. In order to solve this tedious task, several approaches were proposed in the literature over the years, some of them based on analytical formulations and others on heuristic assumptions [23, 25]. For the case under examination, the trial-and-error approach, proposed by Plett, was employed to overcome the lack of robustness exhibited by some analytical methods. Specifically, it was assumed that the state estimation errors were uncorrelated, resulting in a diagonal matrix. It is important to note that higher the value assigned to a state on the diagonal, the greater the associated uncertainty. Therefore,

the proper initialisation was crucial as it directly affects the filter behaviour, indeed, if the values are imposed too low, the filter may overly rely on the model prediction.

The process noise covariance matrix  $Q_x$  captures the uncertainties in the state equations that arise from an improper cell parametrisation or imperfections in the cell model structure. It determines the precision in the state prediction (step 1.a) obtained relying only on the equations describing the state evolution of the cell model. In the literature, two main approaches can be found: one relies on a constant matrix defined by the filter designer, while the other defines a time-varying matrix computed at each time step from the covariance of the model parameters [26]. In this thesis, the first approach was adopted to reduce the computation effort and facilitate subsequent C code implementation.

The last parameter, represented by the measurement noise covariance matrix  $R$ , quantifies the uncertainties associated with the sensor measurements. It essentially determines how much the filter relies on the measurement during the state estimation (step 2.b) with respect to the model prediction. In general, the larger the matrix, the less reliable the measurements and, as a consequence, the greater the reliance on the model prediction. In the literature, the matrix  $R$  is typically determined based on the square root of the measurement error. However, due to the absence of this information in the database, the matrix  $R$ , which, in this case, was a scalar value, was determined by a trial-and-error approach.

On the basis of these theoretical considerations, and after several tuning attempts, the initial error covariance matrix ( $P_{\tilde{x}_0}^+$ ) and the two noise covariance matrices ( $Q_x$ ,  $R$ ) were defined in order to obtain an estimated SOC profile consistent with the one obtained by the Coulomb counting method. The final tuned parameters are reported below:

$$P_{\tilde{x}_0}^+ = \begin{bmatrix} 0.09 & 0 & 0 \\ 0 & 0.02 & 0 \\ 0 & 0 & 0.02 \end{bmatrix}$$

$$Q_x = \begin{bmatrix} 1.5 \cdot 10^{-10} & 0 & 0 \\ 0 & 9.7 \cdot 10^{-4} & 0 \\ 0 & 0 & 6.6 \cdot 10^{-7} \end{bmatrix}$$

$$R = 0.15$$

From the reported matrices, some important considerations can be drawn. The first concerns the diagonal element of the matrix  $P_{\hat{x}_0}^+$  corresponding to the uncertainty level of the SOC. This value was intentionally imposed higher compared to those corresponding to the two RC networks in order to mitigate possible small inaccuracies in the measured capacity adopted to compute the SOC profile throughout the tests. The second aspect involves the first diagonal element of the process noise covariance matrix  $Q_x$ , which was set lower compared to the other two elements. This choice was made to prevent an oscillatory behaviour in the estimated SOC profile at the expense of a slight reduction in responsiveness.

### 3.3 Simulink Model Validation

In order to proceed with the subsequent C code generation, the EKF *Simulink* model was subjected to an aggressive validation phase to verify its robustness, accuracy, and suitability for real-time application. Specifically, three different scenarios were considered:

1. Correct initial SOC;
2. Incorrect initial SOC;
3. Current offset injection.

#### 3.3.1 Correct initial SOC

The first scenario considered was the most conservative since the filter was initialised according to the procedure described in the *Filter Tuning* section. In this case, the initial filter SOC was set equal to the value provided by the database.

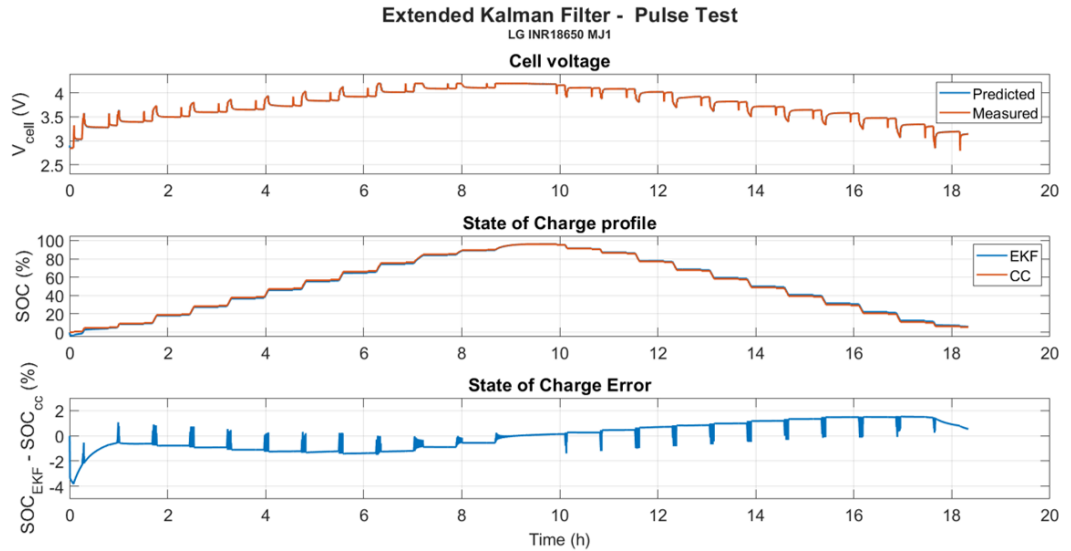
##### Pulse Test

The first experimental test analysed was the *Pulse Test* with an initial SOC equal to 0%. Figure 3.8 shows the results of the *Simulink* EKF model simulation. The upper graph depicts, in blue, the predicted cell voltage computed through the blocks reported in Figure 3.5b representing Equation 3.8, and, in red, the measured cell voltage reported in the database. As clearly visible, apart from the initial instants, where a small discrepancy appears due to the filter convergence, the two curves remain perfectly overlapped over the entire test duration, indicating a proper DPM parametrisation. To further support this consideration, the voltage RMSE throughout the entire test duration, evaluated

with Equation 2.7, is only 3.31 mV.

The central graph reports two SOC profiles. The blue represents the estimated SOC obtained through the EKF *Simulink* model, while the red curve corresponds to the SOC computed with Coulomb counting method. Although the latter closely resembles the profile proposed in Figure 2.7, a difference in the calculation approach is present. Indeed, while in Figure 2.7, the profile was obtained by dividing the net capacity by the maximum value of the corresponding phase (charging or discharging), in the current Figure, the profile was evaluated by dividing the net capacity by the nominal one ( $Q_{nom}$ ). As evident, the estimated SOC profile, despite the initial deviations from the reference curve, rapidly converges and accurately tracks it, demonstrating the effectiveness of the filter tuning.

The bottom graph, supporting the aforementioned considerations, illustrates the difference between the estimated SOC and the reference profile. As can be observed, the error remains enclosed in a window of  $\pm 1\%$  throughout the entire test duration, with slightly larger deviations occurring at the initial and final stages. The final deviation is mainly attributed to the limitations of the DPM in capturing the dynamic behaviour of the cell at low SOC, whereas the initial one arises from a combination of this model limitation and the filter convergence dynamics.



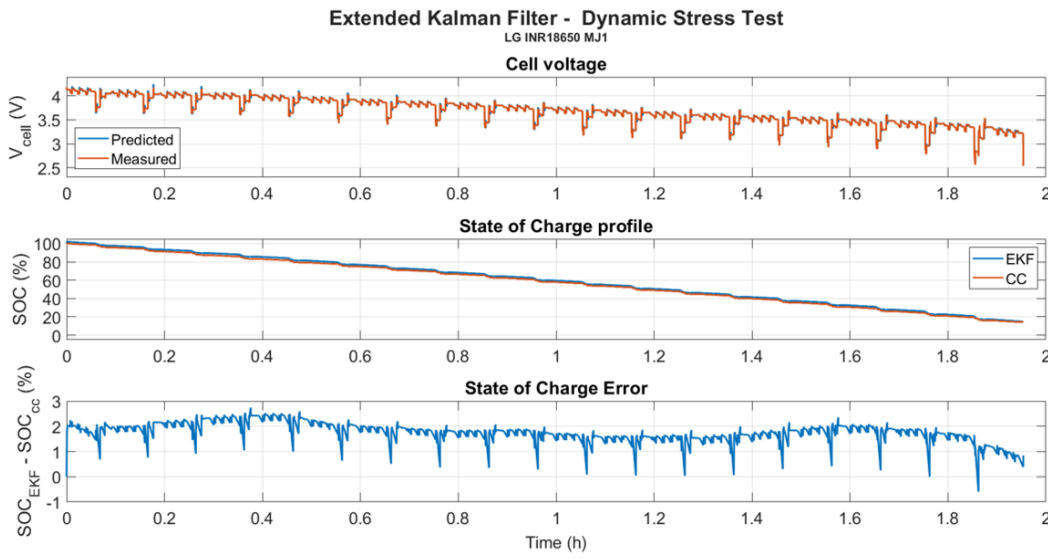
**Figure 3.8:** Pulse Test Single cell EKF validation - Correct initial SOC

### Dynamic Stress Test

The second experimental test analysed was the DST with an initial SOC equal to 100%. Figure 3.9, subdivided in the same way as Figure 3.8, depicts the outcome of the *Simulink* EKF model simulation.

As in the *Pulse Test* validation, the predicted cell voltage accurately traces the measured profile during the entire test duration. The main difference lies in the voltage RMSE, equal to  $9.69\text{ mV}$ , resulting slightly higher compared to the previous case. This increase is attributed to the greater dynamics of the DST, which makes the cell modelling by the DPM more challenging.

The SOC profile and the corresponding error plots emphasise the proper filter tuning. Indeed, the estimated SOC curve closely follows the reference profile during the entire test duration without exhibiting significant drift and deviation. However, due to the more aggressive dynamic load variations of the DST, the error appears more jagged compared to the *Pulse Test*, leading to an average SOC error of  $1.60\%$ .



**Figure 3.9:** DST Single cell EKF validation - Correct initial SOC

### 3.3.2 Incorrect initial SOC

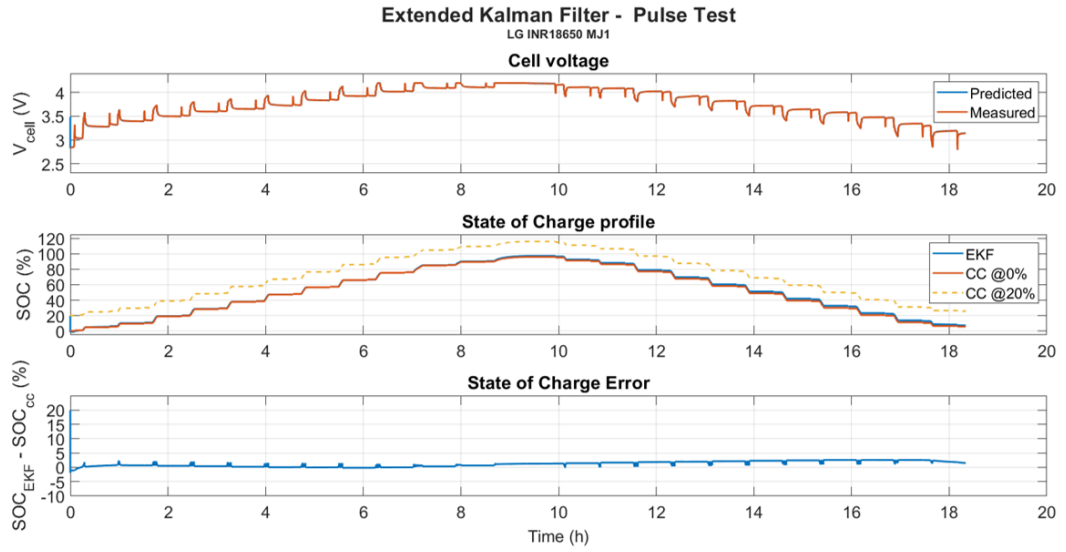
The second scenario analysed aimed to verify the filter capability to converge to the true SOC profile when the initialisation of the cell state (phase 0) was incorrectly set. It should be noted that, throughout the proposed scenarios, only the first element, corresponding to the SOC, was allowed to vary, while the two voltages across the RC networks were maintained equal to zero.

#### Pulse Test

For the considered test, being characterised by relatively smooth dynamics, the initial SOC was deliberately set to  $20\%$ , a value significantly different from the actual initial state. As illustrated in Figure 3.10, which presents the results of this scenario obtained with the EKF *Simulink* model, the filter demonstrates a strong capability to rapidly converge to the actual profiles of both measured

voltage and SOC (red curves). Indeed, despite the evident deviation at the beginning of the test caused by the significant mismatch between the true and imposed initial SOC, the voltage RMSE remains only  $3.33\text{ mV}$ , and the average SOC error is limited to  $1.18\%$ , with the largest contribution deriving from the initial stage.

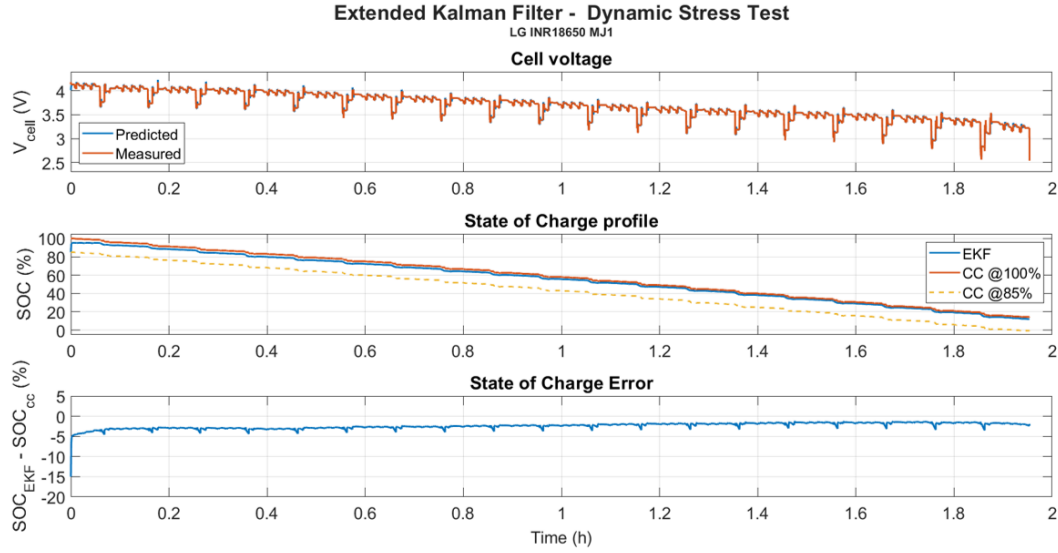
Furthermore, the central graph of Figure 3.10 reveals the main limitation of the Coulomb counting method. The yellow dashed line, representing the SOC profile obtained by the integration method with an incorrect initial state, fails to converge to the true SOC, exhibiting a persistent shift throughout the entire test duration.



**Figure 3.10:** Pulse Test Single cell EKF validation - Incorrect initial SOC

### Dynamic Stress Test

For the DST, the initial SOC value was intentionally imposed equal to  $85\%$ . By analysing Figure 3.11, similar considerations to those extracted from the *Pulse test* validation can be drawn. In this case, however, the higher dynamics and the shorter duration of the test result in slower filter convergence, as confirmed by a voltage RMSE of  $9.35\text{ mV}$  and an average SOC error of  $-2.38\%$ . Despite these phenomena, the filter still manages to achieve convergence of both the cell voltage and SOC profiles, indicating the effectiveness of the tuning. In addition, the central plot further highlights the main drawback of the Coulomb counting method with a SOC profile dropping below the lower cell threshold of  $0\%$ .

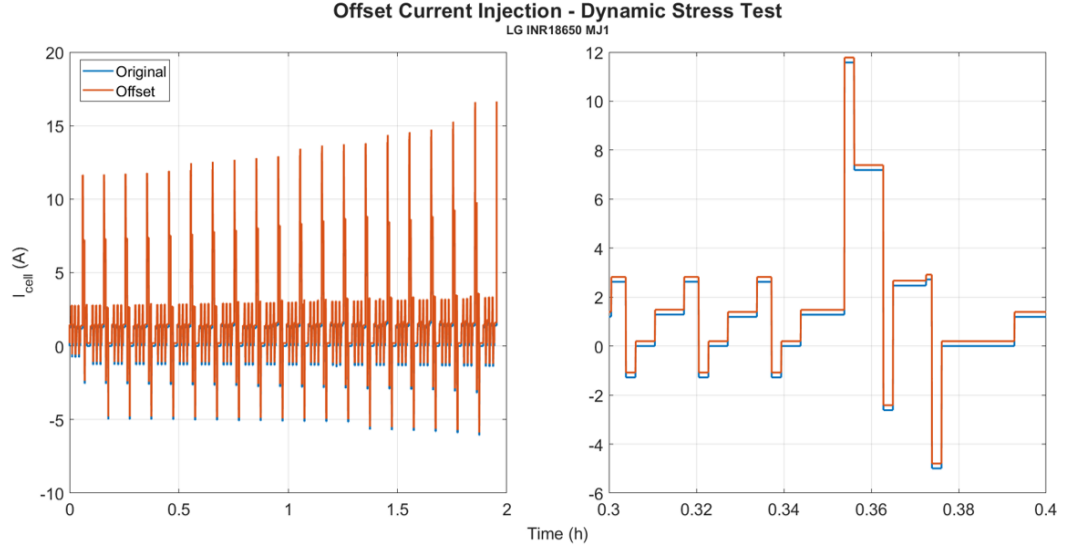


**Figure 3.11:** DST Single cell EKF validation - Incorrect initial SOC

### 3.3.3 Current offset injection

The last scenario considered was the most aggressive and challenging condition for the filter. It consisted of injecting a positive current offset of 0.2 A into the original current profile of the experimental test. This introduction results in an upward shift of the experimental current profile, thus intensifying the discharge effect experienced by the cell and, simultaneously, reducing the charge contribution. To further increase the complexity for the filter, this scenario was performed on the DST, which, due to its highly dynamic load variations, increases the risk of exceeding the lower cell threshold of 0%.

Figure 3.12, which illustrates both the current profile reported in the database (labelled **Original**) and the modified one, supports the previous considerations.

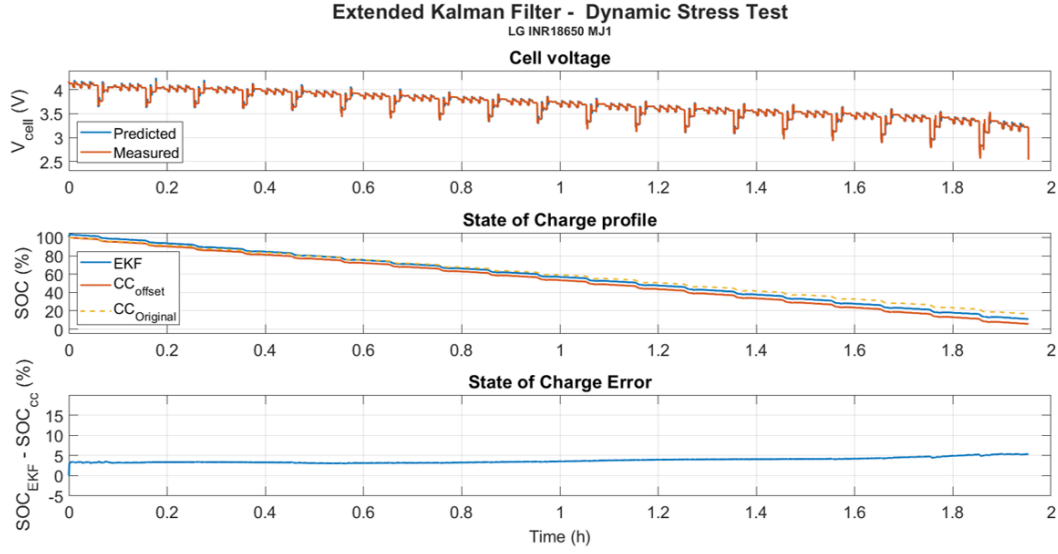


**Figure 3.12:** DST offset current profile

From Figure 3.13, which presents the simulation results for this scenario, the advantages of the EKF emerge. Indeed, the Coulomb counting offset SOC profile, represented by the red line and computed according to Equation 3.18 due to the absence of measured capacity for this modified current profile, drops below the lower cell physical limit. Contrarily, the SOC estimated by the filter, depicted in blue, remains within the admissible cell range and, thanks to the innovation step with the measured voltage feedback, closely follows the **CC Original** curve despite the modified current profile. Towards the end of the test, the combined effect of DPM model limitations and highly dynamic discharge current profile hinders the filter convergence, leading to a greater deviation of the EKF curve from the reference. Nonetheless, the EKF curve remains closer to the reference profile throughout the entire test duration with respect to the **CC Offset** curve, highlighting the filter robustness compared to the simple integration method.

A final important consideration concerns the error graph, which, differently from the previous validation cases where it represented the deviation of the EKF curve from the reference, in this scenario, illustrates the correction effect introduced by the filter.

$$SOC_{cc}(k) = \text{CC Offset} = SOC_{cc}(k-1) - \frac{dt \cdot I_{cell}(k)}{Q_{nom}} \quad (3.18)$$



**Figure 3.13:** DST Single cell EKF validation - Offset current profile

Table 3.2 summarises the results in terms of voltage RMSE and average SOC error obtained from the EKF *Simulink* model validation process under the three different scenarios analysed.

| Validation scenario      | Test  | $V_{RMSE}$ (mV) | $\overline{SOC}_{error}$ (%) |
|--------------------------|-------|-----------------|------------------------------|
| 1: Correct initial SOC   | Pulse | 3.31            | -0.05                        |
|                          | DST   | 9.69            | 1.60                         |
| 2: Incorrect initial SOC | Pulse | 3.33            | 1.18                         |
|                          | DST   | 9.35            | -2.38                        |
| 3: Current offset        | DST   | 9.75            | /                            |

**Table 3.2:** Validation process for different scenarios

# Chapter 4

## Simulink Model Extension

The current chapter is dedicated to describing the procedure followed to extend the EKF *Simulink* model, previously described and developed for single-cell SOC estimation, to a configuration capable of simultaneously estimating SOC for six cells. Furthermore, a preliminary validation phase is included to verify the correctness of the model before code generation.

### 4.1 Model Extension for Multi-Cell Estimation

In order to ensure the deployment of the SOC estimator at battery module level, the EKF *Simulink* model, initially designed and validated for a single cell, as discussed in Sections 3.2 and 3.3, was extended to six cells. Figure 4.1 illustrates the subsystems contained within the main EKF masked block. Each subsystem, labelled from **Cell 1** to **Cell 6**, reproduces the structure reported in Figure 3.2, thus receiving as inputs the current and the measured voltage of the considered cell and providing the estimated SOC profile as output. It is worth noting that, being the battery module composed of series-connected cells, the input current is common to all members, whereas the measured voltage remains specific to each individual cell.

Furthermore, in order to reflect real operating conditions where each cell may start from a different initial state, the variable `initState`, defined within the subsystem *Initialisation & Delay*, was assigned for each cell and set as a tunable parameter in the mask. Table 4.1 reports the relevant parameters of the mask created on the main EKF subsystem:

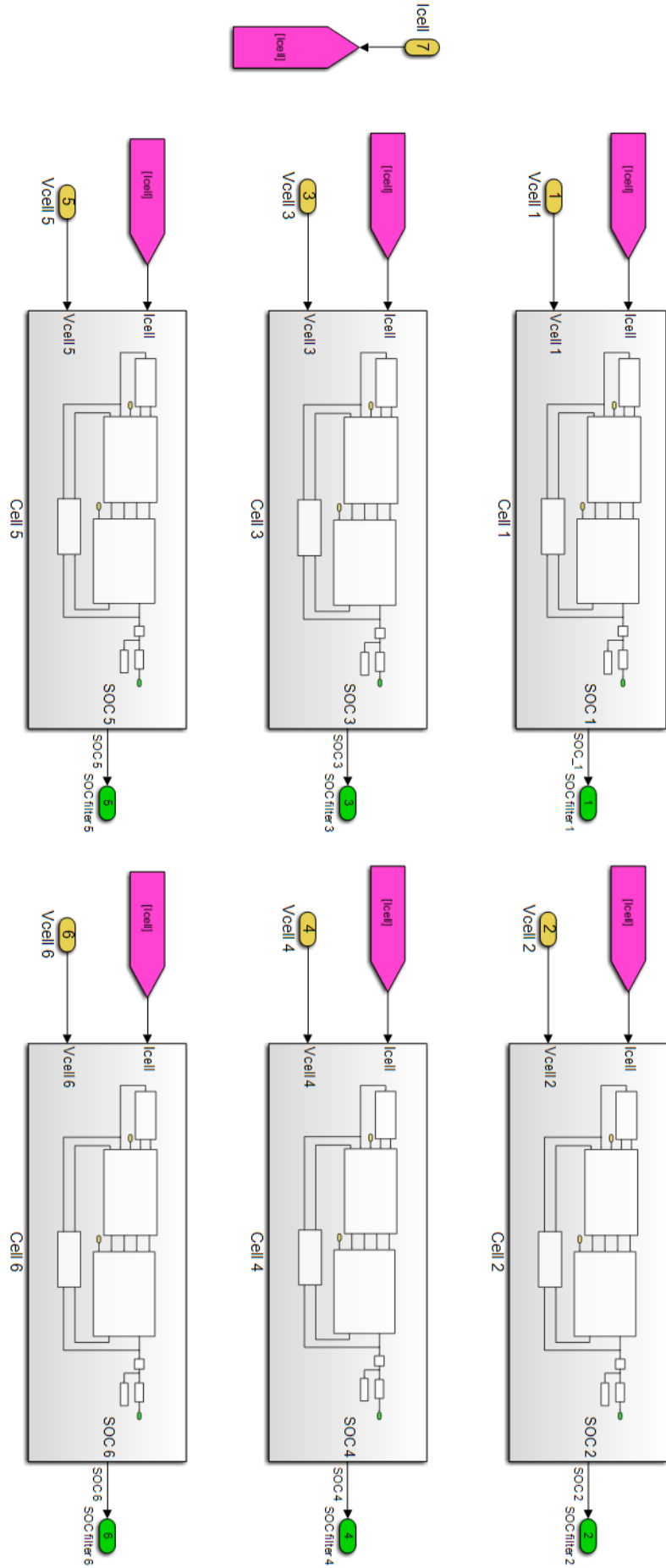


Figure 4.1: Multiple cells *EKF* subsystems overview

| Parameter                       | Variable Name | Dimensions |
|---------------------------------|---------------|------------|
| Initial state Cell 1            | initState1    | 3x1        |
| Initial state Cell 2            | initState2    | 3x1        |
| Initial state Cell 3            | initState3    | 3x1        |
| Initial state Cell 4            | initState4    | 3x1        |
| Initial state Cell 5            | initState5    | 3x1        |
| Initial state Cell 6            | initState6    | 3x1        |
| Initial error covariance matrix | initP         | 3x3        |
| Process noise matrix            | Q             | 3x3        |
| Measurement noise matrix        | R             | 1x1        |
| Initial current                 | initCellCurr  | 1x1        |
| Cell nominal capacity           | nomCapAh      | 1x1        |
| SOC breakpoints                 | SOCbrk        | 1x21       |
| LUT values $R_0$                | R0val         | 1x21       |
| LUT values $R_1$                | R1val         | 1x21       |
| LUT values $R_2$                | R2val         | 1x21       |
| LUT values $\tau_1$             | tau1val       | 1x21       |
| LUT values $\tau_2$             | tau2val       | 1x21       |
| LUT values OCV                  | ocvVal        | 1x21       |
| LUT values OCV derivative       | dOCVdSOC      | 1x21       |

**Table 4.1:** Multiple cells *EKF* mask parameters

Analysing Table 4.1, it can be observed that all the DPM parameters and the covariance matrices adopted by the filter logic are unique and defined for a single cell. This configuration was ensured by the composition of the battery module under consideration, made up of the same cell type (LG INR18650 MJ1).

## 4.2 Multiple cells EKF Model Validation

In order to verify the correctness of the multiple cells EKF *Simulink* model, a validation phase was conducted based on the previously analysed tests. Specifically, the multiple cells model, illustrated in Figure 4.1, was subjected to an extremely rigorous validation process, similar to the one conducted for the single cell model (see Section 3.3). However, to maintain a focused and clear thesis structure, the discussion is limited only to the cases reported in Table 4.2:

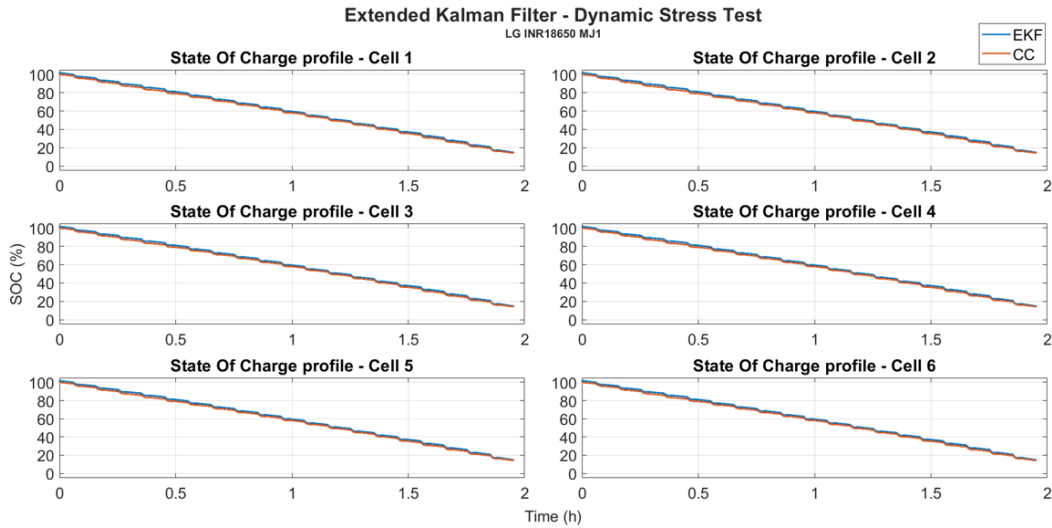
| Scenario | Description                                                   |
|----------|---------------------------------------------------------------|
| I        | DST with correct initial SOC                                  |
| II       | Pulse test with common incorrect initial SOC for all cells    |
| III      | Pulse test with different incorrect initial SOC for each cell |
| IV       | DST with different incorrect initial SOC for each cell        |

**Table 4.2:** Validation scenarios applied to the multiple cells EKF

### 4.2.1 Scenario I

Within the proposed validation scenario, the initial SOC<sub>0</sub>s of the six cells (first element from `initState1` to `initState6`) were imposed equal to the correct test value, corresponding to  $SOC_0 = 1 = 100\%$ . In addition, the measured voltage profile of each cell was set equal to the single cell database profile, shown in the central plot of Figure 2.12.

Consequently, as a result of this configuration, and as shown in Figure 4.2, all cells, being subjected to the same input conditions, exhibit identical SOC profiles. Moreover, since these inputs correspond to those imposed in Subsection 3.3.1, the SOC<sub>0</sub>s reported in Figure 4.2 coincide with the single cell result depicted in Figure 3.9.

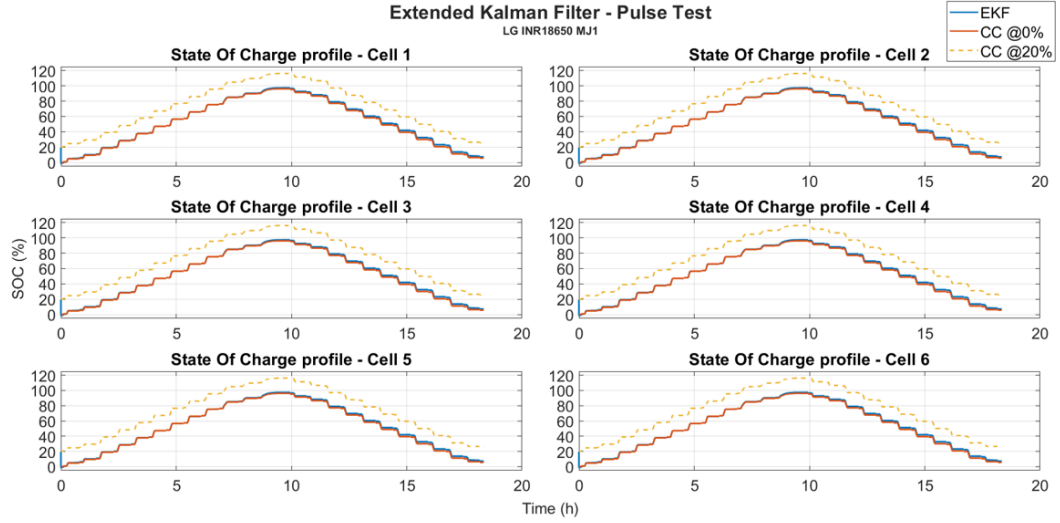


**Figure 4.2:** Validation scenario I - SOC profiles

### 4.2.2 Scenario II

For the second scenario considered, the initial SOC<sub>0</sub>s of the six cells (first element from `initState1` to `initState6`) were all imposed to  $SOC_0 = 0.2 = 20\%$ . In addition, following the same procedure adopted in the previous validation scenario, the measured voltage of each cell was set equal to the single cell

database profile. Consequently, since the input conditions correspond to those imposed in Subsection 3.3.2, all SOC profiles depicted in Figure 4.3 are identical to each other and coincide with the one reported in Figure 3.10.



**Figure 4.3:** Validation scenario II - SOC profiles

Although the proposed discussion focuses only on SOC, the cell voltages predicted by the filter were consistent with the profiles depicted in the upper plot of Figure 3.9 and Figure 3.10, respectively.

### 4.2.3 Scenario III

This scenario was developed in order to reproduce a more realistic operating condition, in which cells within the battery pack may have different initial SOC levels. Specifically, being the test analysed the *Pulse*, the first elements of the initial state vector `initState` (from 1 to 6) were imposed in the neighbourhood of the actual value of 0%, as reported in Table 4.3. As clearly visible, to increase the difficulty of filter convergence and to test its adaptability, the initial  $SOC_{s_0}$  were randomly assigned, covering a wide deviation from the actual test value.

|             | Cell 1 | Cell 2 | Cell 3 | Cell 4 | Cell 5 | Cell 6 |
|-------------|--------|--------|--------|--------|--------|--------|
| $SOC_0$ (%) | 10     | 15     | 20     | 5      | 0      | 25     |

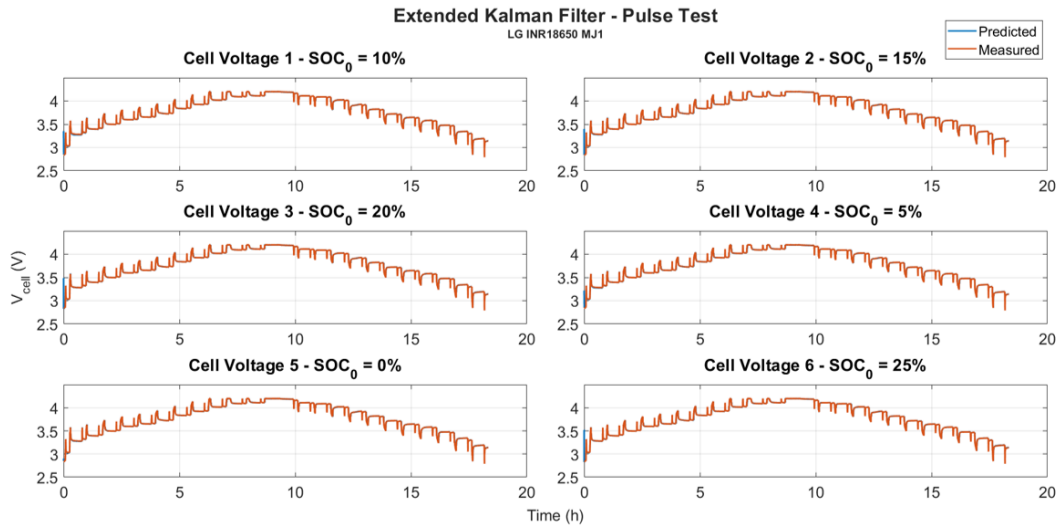
**Table 4.3:** Validation scenario III - Initial SOC

Figure 4.4 reports the results obtained by simulating the multiple cells EKF *Simulink* model under the considered configuration. The upper plot (Figure 4.4a) illustrates the cell voltage profiles, while the lower one (Figure 4.4b) depicts the estimated SOC for all six cells.

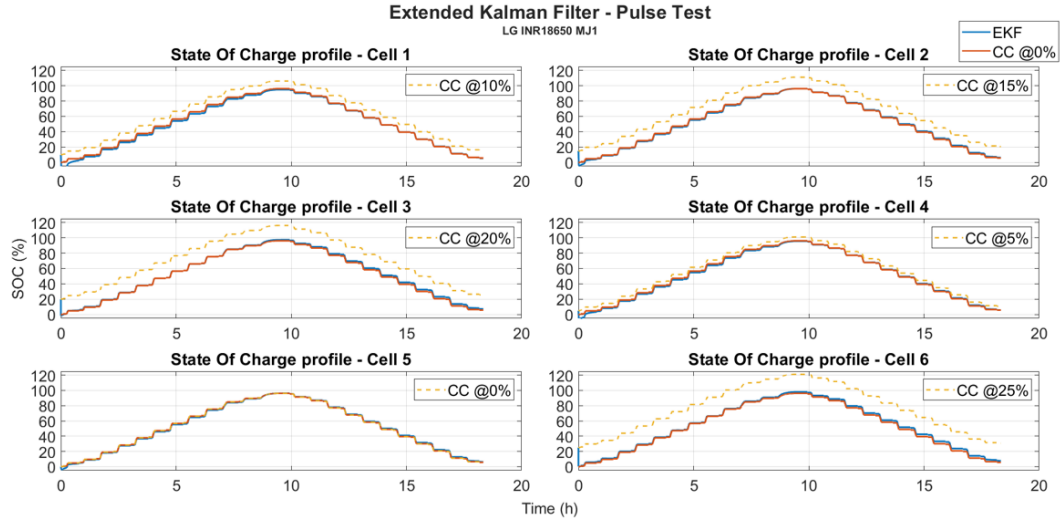
Starting from Figure 4.4a, it can be observed that, regardless of the initialisation deviation, the predicted cell voltage by the filter closely overlaps with the measured throughout the entire test. The only visible discrepancies, attributable to filter convergence, occur in the initial simulation stages, where an increased initial SOC leads to a higher initial predicted cell voltage. This behaviour, consistent with the monotonic trend of the OCV curve, implies a longer convergence time and, thus, a more evident mismatch between the voltage profiles.

By examining Figure 4.4b, the effectiveness of the filter over the Coulomb counting method is clearly demonstrated. Indeed, thanks to the optimal filter tuning, the estimated SOC consistently converges towards the reference trajectory (red curve labelled as CC @0%) across all six cells, regardless of the imposed initial condition. This behaviour highlights the ability of the filter to compensate for initialisation errors and to provide an accurate tracking of the SOC evolution.

The voltage and SOC profiles of cells 3 and 5, being subjected to the same initial conditions applied in Section 3.3, correspond, respectively, to those reported in Figure 3.10 and Figure 3.8.



(a) Validation scenario III - Cell voltage profiles



(b) Validation scenario III - SOC profiles

**Figure 4.4:** Validation scenario III

To conclude the analysis of this scenario, Table 4.4 summarises the results in terms of voltage RMSE and mean SOC error for each cell. The outcomes confirm the considerations discussed above. Indeed, the cells initialised around 10% present the largest  $V_{RMSE}$ , highlighting the limitations of the DPM in describing accurately the cell behaviour at low SOC. Concerning SOC, the mean error, evaluated by averaging the difference between the filter estimated and Coulomb counting SOC, inevitably increases with the initialisation deviation. Nonetheless, it is worth noting that the main contribution arises from the initial simulation steps, before the filter reaches convergence.

|                              | Cell 1 | Cell 2 | Cell 3 | Cell 4 | Cell 5 | Cell 6 |
|------------------------------|--------|--------|--------|--------|--------|--------|
| $V_{RMSE}$ (mV)              | 4.17   | 3.51   | 3.33   | 3.64   | 3.31   | 3.32   |
| $\overline{SOC}_{error}$ (%) | -1.41  | -0.11  | 1.18   | -0.67  | -0.05  | 1.73   |

**Table 4.4:** Validation scenario III - Summary results

#### 4.2.4 Scenario IV

The last scenario considered was intended, similarly to the previous one, to reproduce a realistic operating condition with different initial SOC on a dynamic cycle. Specifically, the EKF *Simulink* model was tested on the DST by imposing the first elements of the initial state vector `initState` (from 1 to 6) in the neighbourhood of the actual value of 100%, as reported in Table 4.5. This scenario constitutes an extremely challenging validation case for the EKF, as it combines the highly dynamic behaviour of the DST with its short duration.



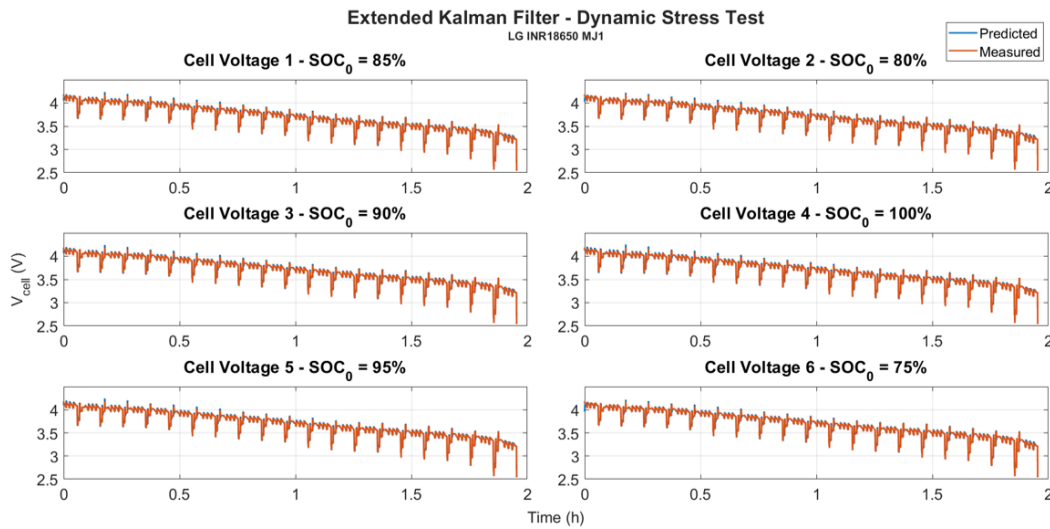
|             | Cell 1 | Cell 2 | Cell 3 | Cell 4 | Cell 5 | Cell 6 |
|-------------|--------|--------|--------|--------|--------|--------|
| $SOC_0$ (%) | 85     | 80     | 90     | 100    | 95     | 75     |

**Table 4.5:** Validation scenario IV - Initial SOC's

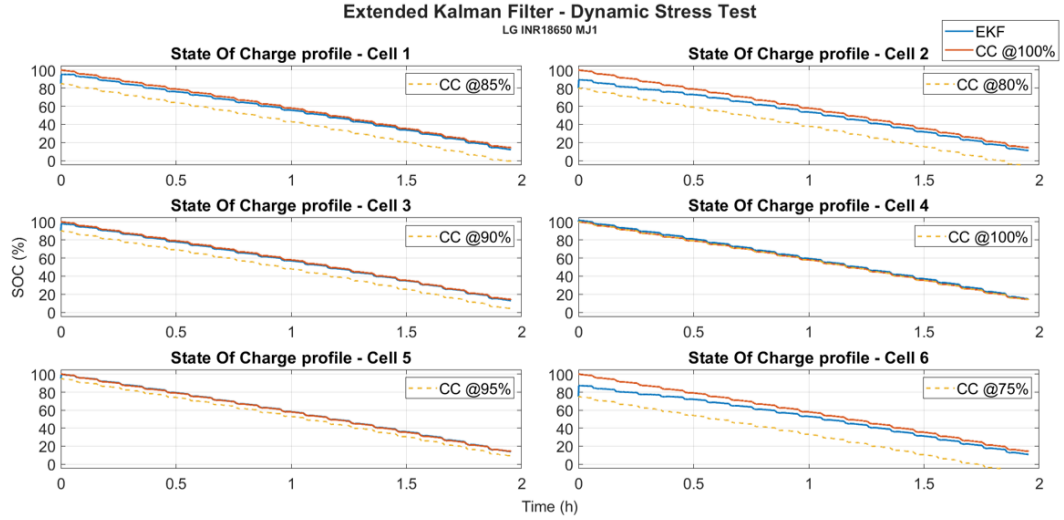
Figure 4.5 illustrates the simulation outcomes for the considered scenario. The analysis of Figure 4.5a reveals that the predicted cell voltage profiles, despite initial errors caused by incorrect SOC initialisations, closely follow the measured ones throughout the entire test.

As shown in Figure 4.5b, the filter demonstrates convergence capability towards the reference profiles (red curves labelled as CC @100%) for all six cells, regardless of the imposed initial value. However, convergence proves to be more challenging for cells 2 and 5, mainly due to three factors:

- Large discrepancy between actual and imposed initial SOC conditions;
- Relatively short test duration;
- Reduced sensitivity of the DPM parameters in the usable operating region (20% ÷ 80% SOC). This last aspect plays a crucial role in the filter operating behaviour. In fact, within this region, the OCV curve, governing the cell voltage, exhibits a limited slope profile. This phenomenon implies that even a significant SOC variation is translated into a minimal modification of the cell voltage. As a consequence, since the voltage profile results almost overlapped with the measured one, the SOC filter correction effect with the feedback measurement strongly diminishes, making convergence slower.



**(a)** Validation scenario IV - Cell voltage profiles



(b) Validation scenario IV - SOC profiles

**Figure 4.5:** Validation scenario IV

Similarly to Scenario III, Table 4.6 shows how cells closer to the upper SOC limit (100%) exhibit a higher voltage RMSE, thereby highlighting the DPM limitations. Furthermore, the SOC error increases with the initialisation error, with the major contribution deriving from the initial simulation steps.

|                               | Cell 1 | Cell 2 | Cell 3 | Cell 4 | Cell 5 | Cell 6 |
|-------------------------------|--------|--------|--------|--------|--------|--------|
| $V_{RMSE} (mV)$               | 9.35   | 9.32   | 9.41   | 9.69   | 9.50   | 9.40   |
| $\overline{SOC}_{error} (\%)$ | -2.38  | -5.37  | -0.98  | 1.60   | 0.26   | -6.16  |

**Table 4.6:** Validation scenario IV - Summary results

On the basis of the results discussed in the analysed scenarios, the multiple cells EKF *Simulink* model can be considered reliable and robust, thus allowing the subsequent thesis steps.



# Chapter 5

## Code Generation

The current chapter is dedicated to describing the procedure adopted for generating C code of the EKF *Simulink* model capable of simultaneously estimating the SOC for six cells.

### 5.1 Code Setting Parameters

After extending and validating the multiple cells EKF *Simulink* model, attention was shifted to properly configure the C code generation process [27]. Specifically, this section presents the main parameters and options defined within the *Configuration Parameters* window to guarantee both the applicability and correctness of the generated code. To facilitate reading, the configuration settings are presented in dedicated subsections, following the same structure adopted in *Simulink*. It should be noted that only the configuration windows in which modifications were applied are reported, whereas all others maintain their default settings.

#### Solver

Within the solver window, the parameters were imposed equal to those adopted during the *Simulink* validation process (Section 3.3), thus:

- Simulation Start time: 0 s;
- Simulation Stop time: `cycleTime(end)`;
- Solver type selection: Fixed-time;
- Solver: Discrete (no continuous states);
- Fixed-step size: `dt = 200 ms`;



## Hardware Implementation

To guarantee consistency with the hardware requirements and suitability for deployment on a real platform, the settings within this window were configured to:

- Device vendor: Freescale;
- Device type: 32-bit PowerPC;
- Device details: default values;

## Code Generation

For the current window, the parameters were set to:

- System Target File: `ert.tlc`;
- Description: Embedded Coder;
- Language: C;
- Language Standard: C99 (ISO);
- Generate code only: cleared;
- Toolchain: Automatically locate an installed toolchain, MinGW64;
- Build configuration: Faster Builds.

## Optimisation

Within the *Optimisation* subsection of the code generation settings, the *Default parameter behaviour* option was configured to allow user-defined parameters and to enable their modification during run mode, without requiring code regeneration. Specifically, all the initial state of the cells and the initial current value were set as `Tunable` and defined as `ExportedGlobal` variables.

After setting up the proposed configuration for the multiple cells EKF *Simulink* model shown in Figure 4.1, it was necessary to ensure that all inputs and outputs could be accessed and called by the functions created during the generation process. Specifically, to achieve this aim, before generating the code and within the Embedded Coder environment, all signals were selected and added to the code mapping interface through the "Add signal to code mapping" command and configured as `ExportedGlobal`. Only at the end of this preliminary setup process, the C code was generated by selecting the "Build This Subsystem" command from the context menu that appears when right-clicking on the main EKF masked block.

## 5.2 Simulink S-Function

The procedure described in Section 5.1 ensured the generation of C code designed for deployment on real hardware with production-level specifications and suitable for experimental validation. However, since these experimental tests will be conducted by third parties, in the present thesis work, the multiple cells EKF *Simulink* model was exported as **S-Function** and validated through the SIL approach. Although the generated C code was not directly employed in this thesis, the procedure description was included, as it was faithfully followed to provide the code to the final users.

To achieve the **S-Function** generation, the multiple cells EKF *Simulink* model was configured with the solver and optimisation settings previously defined. Subsequently, by executing the command "Generate S-Function" directly from the external mask of the model, the **S-Function** was obtained [28]. This process allowed the creation of a *Simulink* model equivalent to the original one (Figure 4.1) in which the EKF logic was no longer implemented through blocks, but instead expressed directly in C code. The resulting model, subdivided into three subsystems as illustrated in Figure 5.1, was equipped with an external mask containing only the parameters imposed to **Tunable**, as reported in Table 5.1. Specifically, the first and third subsystems were involved in managing inputs and outputs of the system, respectively, while the green one was the system's core, implementing the EKF logic for six cells SOC estimation. It is important to point out that the outputs of this model were intentionally named **SOC Code** (from 1 to 6) to prevent variable overwriting when running both models simultaneously.

| Parameter            | Variable Name | Dimensions |
|----------------------|---------------|------------|
| Initial state Cell 1 | initState1    | 3x1        |
| Initial state Cell 2 | initState2    | 3x1        |
| Initial state Cell 3 | initState3    | 3x1        |
| Initial state Cell 4 | initState4    | 3x1        |
| Initial state Cell 5 | initState5    | 3x1        |
| Initial state Cell 6 | initState6    | 3x1        |
| Initial current      | initCellCurr  | 1x1        |

**Table 5.1:** S-Function mask parameters

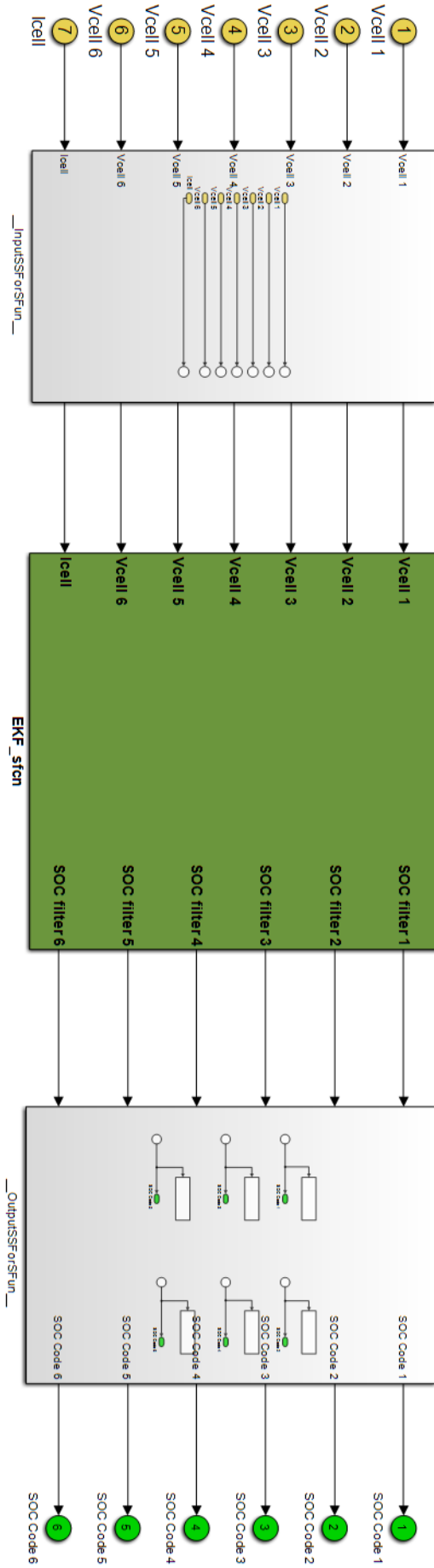


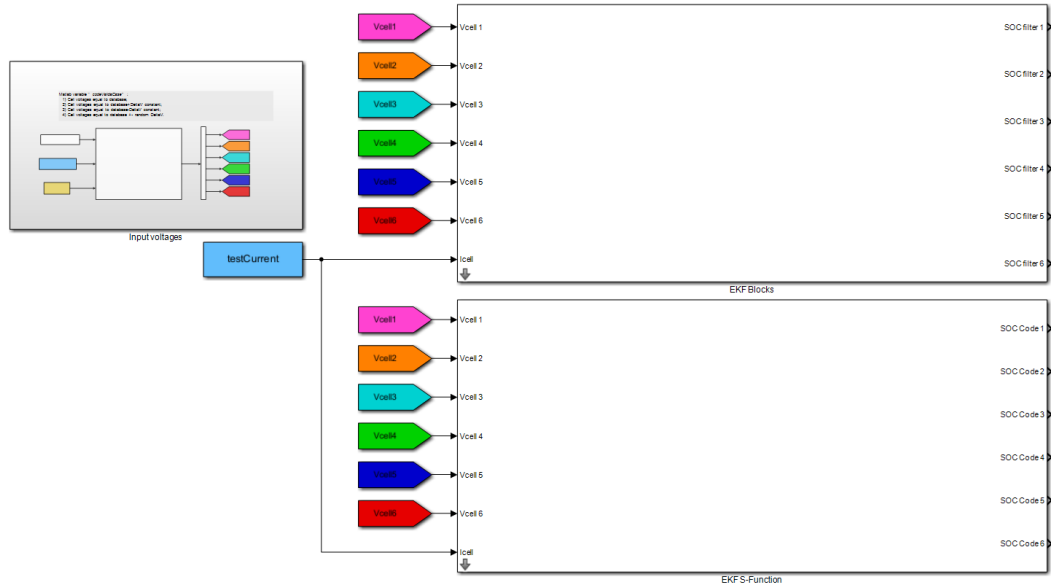
Figure 5.1: Multiple cells EKF S-Function

### 5.3 S-Function validation

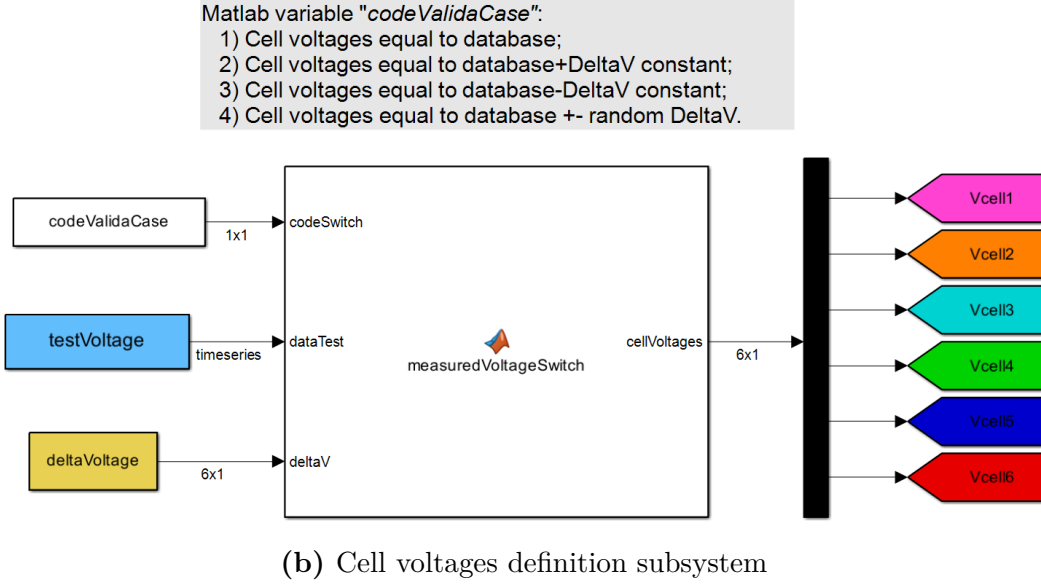
Once the **S-Function** was successfully generated, particular attention was devoted to its validation. This step was fundamental to ensure that the generated code faithfully reproduced the behaviour of the multiple cells EKF *Simulink* model (Figure 4.1), thus guaranteeing consistency and correctness.

To achieve this objective, a dedicated *Simulink* model was developed, as illustrated in Figure 5.2a, comprising:

- The multiple cells EKF model implemented through *Simulink* blocks, referred to **EKF Blocks** and depicted in Figure 4.1;
- The corresponding multiple cells EKF model implemented through C code, denoted as **EKF S-Function** and previously shown in Figure 5.1;
- An auxiliary subsystem, labelled **Input voltages**, responsible for defining the voltage profile associated with each cell. The internal structure is further detailed in Figure 5.2b.



(a) Block-diagram and S-Function EKF implementation



**Figure 5.2:** *Simulink* model for S-Function validation

Among the several validation cases that could be executed and performed through the proposed *Simulink* model, only two representative scenarios, reported in Table 5.2, were selected, in order to avoid redundancy and maintain a concise thesis structure.

| Scenario | Description                                                     |
|----------|-----------------------------------------------------------------|
| 1        | Pulse test with correct initial SOC and reference cell voltages |
| 2        | DST test with correct initial SOC and different cell voltages   |

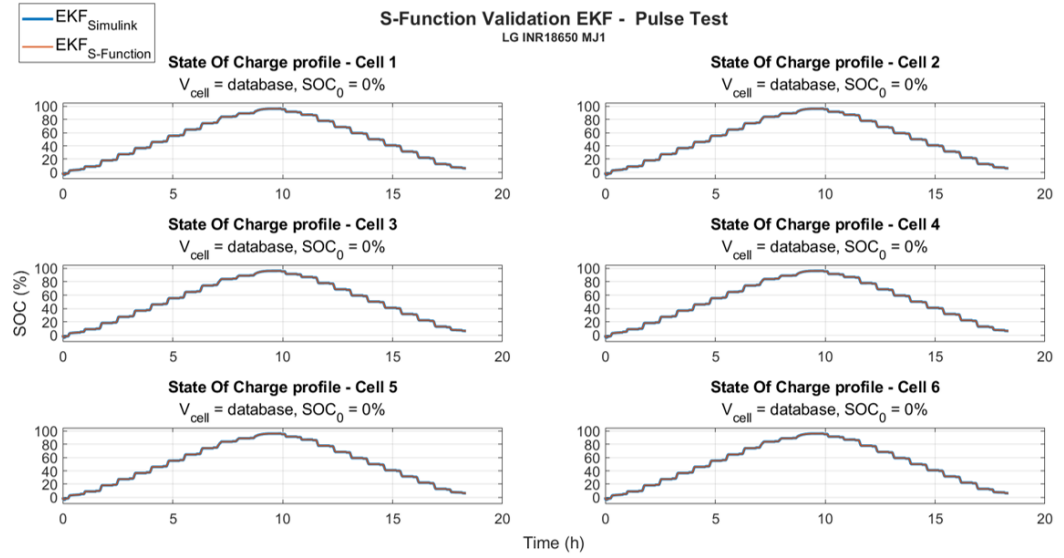
**Table 5.2:** SIL validation scenarios

### 5.3.1 Scenario 1

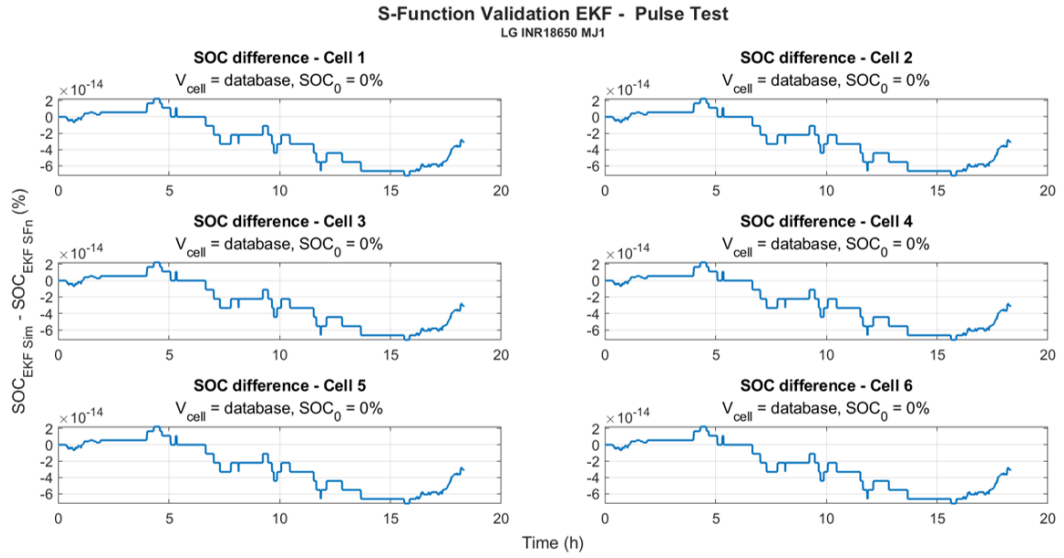
The first scenario analysed was the simplest code validation case considered, specifically designed to verify the proper functionality of the *Simulink* model depicted in Figure 5.1 and, thus, of the generated and refined S-Function. Specifically, the model was subjected to the *Pulse Test*, with the initial SOC of each cell set equal to the actual value ( $SOC_0 = 0\%$ ) and the measured cell voltages imposed according to the reference data contained in the database. The profiles obtained through the simulation are illustrated in Figure 5.3. By examining Figure 5.3a, it can be observed that the SOC profiles estimated by the block-based model (depicted in blue) and by the generated S-Function (red) perfectly overlap for all cells throughout the entire test, confirming the correctness and reliability of the generated C code. Furthermore, since all six cells were subjected to identical input conditions, their SOC profiles coincide

with each other and match the one previously shown in the central plot of Figure 3.8.

Analysing Figure 5.3b, which displays the difference between the estimated SOC profiles obtained from the two models for each cell, a minimal discrepancy can be observed. However, since the deviation is on the order of  $10^{-14}\%$ , it can be reasonably attributed to numerical rounding and solver precision effects, thereby confirming the consistency and equivalence of the two implementations.



(a) SIL validation scenario 1 - SOC profiles



(b) SIL validation scenario 1 - SOC discrepancy

Figure 5.3: SIL validation scenario 1

### 5.3.2 Scenario 2

The second scenario, implemented on the DST with correct initial SOC ( $SOC_0 = 100\%$ ) set for all cells, was designed to replicate realistic operating conditions of the battery module, in which individual cells may exhibit slightly different measured voltages. Specifically, to emulate this behaviour, it was decided to deliberately introduce a random voltage variation to the measured voltages present in the database.

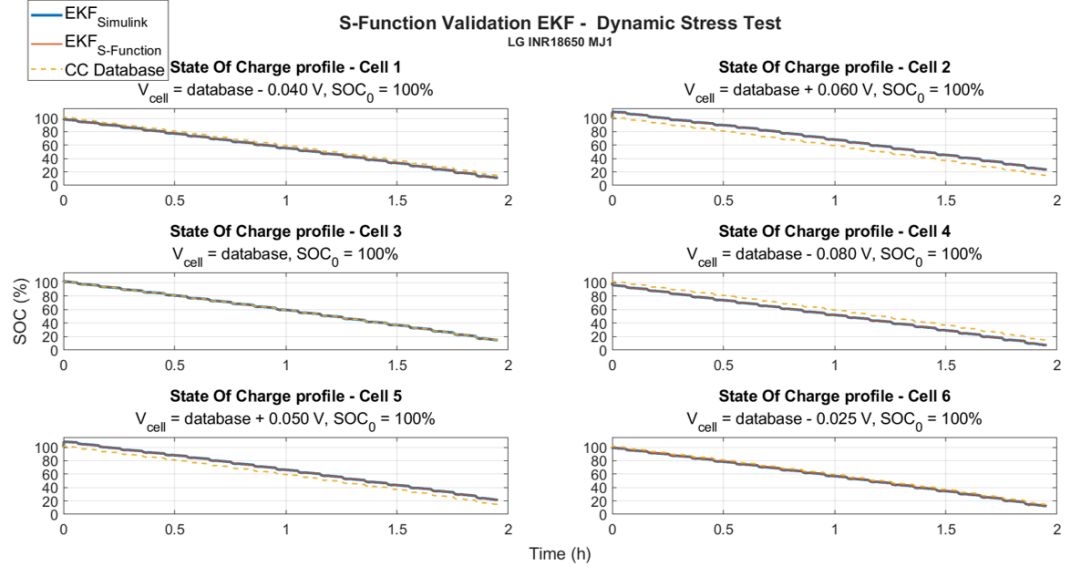
The implementation of this configuration was enabled by the subsystem shown in Figure 5.2b, arranged as reported in Table 5.3.

| Variable                    | Value             |
|-----------------------------|-------------------|
| <code>codeValidaCase</code> | 4                 |
| <code>deltaVoltage</code>   | Cell 1 $-40$ mV   |
|                             | Cell 2 $+60$ mV   |
|                             | Cell 3 $\pm 0$ mV |
|                             | Cell 4 $-80$ mV   |
|                             | Cell 5 $+50$ mV   |
|                             | Cell 6 $-25$ mV   |

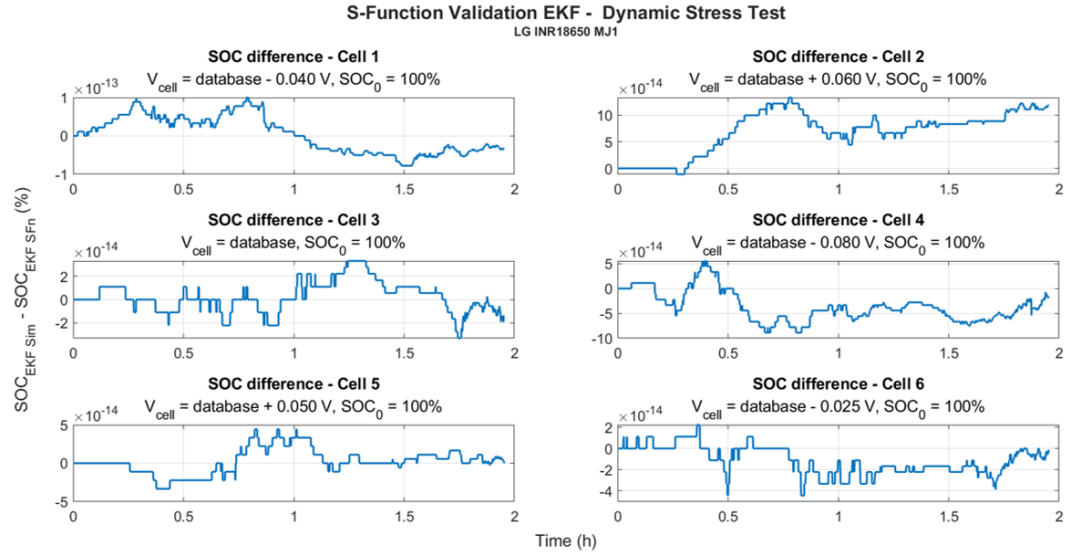
**Table 5.3:** Cell voltages offsets

The results obtained from the simulation are illustrated in Figure 5.4, organised coherently with the previous scenario. The upper plot reports the SOC evolution estimated by both the block-based model and the generated **S-Function** for each cell, whereas the lower plot depicts the corresponding discrepancy between the two estimations. By examining Figure 5.4a, it can be observed that the designed EKF, through the measured cell voltage feedback, is capable of adjusting the SOC evolution profile. In particular, cells 2 and 5, where a positive offset was introduced, exhibit slightly higher SOC profile compared to cell 3, considered as reference condition ( $SOC_0 = 100\%$  and  $V_{cell} = database$ ). Conversely, cells 1, 4, and 6, subjected to a negative offset, display a slightly lower evolution. This behaviour is consistent with the expected cell's electrochemical response, as described in the DPM analysis (Section 2.2). Furthermore, the plot clearly shows the perfect overlap between the SOC profiles obtained from the block-based model and those produced by the generated **S-Function**, confirming the equivalence of the two implementations. This observation is further supported by the negligible discrepancy reported in

Figure 5.4b, which, as in the previous validation case, can be only attributed to floating-point and numerical precision effects.



(a) SIL validation scenario 2 - SOC profiles



(b) SIL validation scenario 2 - SOC discrepancy

**Figure 5.4:** SIL validation scenario 2

The proposed validation case concludes this thesis work, demonstrating the effectiveness and correct operation of the developed EKF, its proper extension to battery module application, and the accurate generation of the corresponding C code.



# Chapter 6

## Conclusions

The entire work presented in this thesis focused on the development of an extended Kalman filter (EKF) designed to estimate the state of charge (SOC) of lithium-ion cells, with specific application to a battery module composed of six identical LG INR18650 MJ1.

The process started from the electrochemical characterisation of a single cell, adopting experimental data retrieved from a public database. To perform this operation, an equivalent circuit model (ECM) based on a Dual Polarisation Model (DPM) was implemented and accurately tuned through Simulink Design Optimisation (SDO) to reproduce the cell voltage dynamics faithfully. Specifically, a voltage RMSE of 25.5 *mV* was achieved during the *Pulse Test* and 23.8 *mV* during the Dynamic Stress Test (DST), highlighting the fidelity of the developed model. Building upon this model, the EKF algorithm was designed and tuned in *Matlab* and *Simulink* environments at single cell level. The logic developed was then extended to a multiple cells configuration capable of simultaneously estimating the SOC of each element within the battery module. The filter validation phase, conducted across several test scenarios, demonstrated:

- Under correct initialisation, the estimated SOC profile remained enclosed within an error window of just  $\pm 2\%$  for both the *Pulse Test* and DST;
- In case of incorrect initialisation, the estimated SOC profile rapidly converged to the reference trajectory during all considered cases. Specifically, for an initial deviation of 25%, the mean SOC error obtained during the *Pulse Test* was 1.73%, and  $-6.16\%$  in the DST, with the largest contribution occurring in the first instants before complete filter convergence.

Overall, the EKF exhibited, due to the optimal tuning, fast convergence, robustness, reliability, and high consistency between the estimated and reference SOC profiles, highlighting the advantages of this estimation technique over the traditional Coulomb counting method.

Finally, the multiple cells *Simulink* model was prepared for embedded deployment through C-Code generation, and its correctness was verified by adopting the Software-in-the-loop (SIL) validation approach. The achieved results confirmed the functional equivalence between the block-based and **S-Function** models, thus demonstrating the proper configuration settings adopted for code generation.

All the discussed considerations validate the robustness, reliability, and real-time integration within a production-level BMS of the developed project.

## 6.1 Future Developments

Although all objectives defined in this thesis were successfully achieved, the proposed work could be further enhanced through the following activities:

- **Temperature modelling:** incorporate the temperature effect into the DPM and consequently on the EKF. Specifically, all 1D LUTs should be replaced by 2D LUTs, introducing both SOC and temperature as variables to describe the dynamic behaviour of the cell. However, to implement this extension, it would be necessary either to access a new experimental database including a wider temperature range or to perform dedicated laboratory tests. This requirement arises from the limited temperature data available in the adopted database, which is therefore inadequate to accurately characterise the thermal effect on the electrochemical behaviour of the cell;
- **State Of Health (SOH) modelling:** extend the proposed EKF to a structure composed of two filters operating simultaneously for each cell, referred to as dual extended Kalman filter (DEKF). In particular, the outer filter would be responsible for estimating the slow-varying variables, such as the ESR ( $R_0$ ) and the nominal capacity ( $Q_{nom}$ ), while the inner one would estimate the SOC, following the same logic implemented in this work. This extension would enable optimal monitoring of cell performance degradations over time, reflected as a gradual decrease in nominal capacity and an increase in ESR [14, 16, 21];
- **Passive cell balancing modelling:** introduce in the EKF resistive elements to emulate real-world behaviour where all the cells within the battery module are maintained at a uniform SOC level, preventing performance degradation and extending lifespan;
- **Real experimental tests:** conduct real laboratory tests to validate both the EKF and the generated code behaviour under different scenarios.

# Appendix A

## Cad Model

As mentioned in section 2.1, to provide a visual representation of the cell's shape and dimensions, a 3D cad model is proposed and shown in Figure A.1. It is worth noting that the latter was not designed and created from scratch, but rather sourced from *3D Cad Browser* [29].



**Figure A.1:** 3D Cell Cad Model



# Appendix B

## Matlab Code EKF

The current chapter is devoted to presenting the *Matlab* function developed in parallel to the proposed *Simulink* model (Figure 3.2) for the implementation of the single cell EKF SOC estimation, whose inputs and outputs are reported in the following Table B.1.

|        | Variable Name   | Description                                            |
|--------|-----------------|--------------------------------------------------------|
| Input  | states          | Vector of estimated states                             |
|        | initStates      | Initial states as column vector                        |
|        | initP           | Initial covariance matrix                              |
|        | prevP           | Covariance matrix at previous time step                |
|        | Qprocess        | Process noise matrix                                   |
|        | Rmeasure        | Measurement noise matrix                               |
|        | inpCurr         | Vector of input current (A)                            |
|        | voltage         | Vector of measured voltage (V)                         |
|        | dati            | Matlab cell containing all the ECM and cell parameters |
|        | index           | Actual time step                                       |
| Output | predStates      | Predicted states at current time step                  |
|        | predVoltage     | Predicted measured voltage (V)                         |
|        | estimatedStates | Estimated states at current time step                  |
|        | prevP           | Covariance matrix at current time step                 |
|        | prevP           | Updated previous covariance matrix                     |

**Table B.1:** EKF Matlab function Input & Output



```
1 function [predStates, predVoltage, estimatedStates,
    estimatedP, prevP] =
    ExtendedKalmanFilterFcn(states, initState, initP,
    prevP, Qprocess, Rmeasure, inpCurr, voltage, dati,
    index)
2
3 Solutions to avoid errors:
4
5 1. Insert the first prevP as zeros(size(initP));
6
7 2. Initialise in the main for loop the first index as prevP =
    initP.
8
9 % Data extraction
10 R0Lut = dati{1,2};
11 R1Lut = dati{1,4};
12 R2Lut = dati{1,6};
13 tau1Lut = dati{1,8};
14 tau2Lut = dati{1,10};
15 Qnom = dati{1,12}; %[Ah]
16 brkSoc = dati{1,14};
17 ocvLut = dati{1,16};
18 ocvSocLut = dati{1,18};
19 dt = dati{1,20}; %[s]
20
21 %Phase 0: Initialisation
22 For the first time instant use the initial values of
    states, covariance matrix and current
23 if index ~= 1
24     previousSOC = states(:,index-1);
25     previousCurrent = inpCurr(index-1);
26 else
27     previousSOC = initState;
28     previousCurrent = inpCurr(1);
29     prevP = initP;
30 end
31
32 %Phase 1: Prediction phase
33 %1.a) State Prediction
34
35 %Definition of the parameters @ previous SOC
36 R1 = extrapolationFcn(brkSoc, R1Lut, previousSOC(1));
37 R2 = extrapolationFcn(brkSoc, R2Lut, previousSOC(1));
```



```
34     tau1 = extrapolationFcn(brkSoc, tau1Lut,
previousSOC(1));
35     tau2 = extrapolationFcn(brkSoc, tau2Lut,
previousSOC(1));
36
37     % Definition of the state transition matrices
38     Amatrix = [1 0 0;
39               0 exp(-dt/tau1) 0;
40               0 0 exp(-dt/tau2)];
41
42     Bmatrix = [-dt/(Qnom*3600);
43               (1 - exp(-dt/tau1))*R1;
44               (1 - exp(-dt/tau2))*R2];
45
46     predStates = Amatrix*previousSOC +
Bmatrix*previousCurrent;
47
48 %1.b) Error Covariance Prediction
49     predP = Amatrix*prevP*Amatrix' + Qprocess;
50
51 %1.c) System Output Prediction: variable calculated @
predicted state
52     ocv = extrapolationFcn(brkSoc, ocvLut,
predStates(1));
53     R0 = extrapolationFcn(brkSoc, R0Lut, predStates(1));
54
55     %Vcell = OCV - VRC1 -VRC2 - R0*I
56     predVoltage = ocv - predStates(2) - predStates(3) -
R0*inpCurr(index);
57
58 %Phase 2: Correction Phase
59 %2.a) Kalman Gain Evaluation
60     derOCV = extrapolationFcn(brkSoc, ocvSocLut,
predStates(1));
61
62     %Cmatrix
63     Chat = [derOCV -1 -1];
64
65     Sk = Chat*predP*Chat' + Rmeasure;
66     KalGain = predP*Chat'/Sk;
67
68 %2.b) State Estimation
69     estimatedStates = predStates +
```



```
70 KalGain*(voltage(index) - predVoltage);
71
72 %If necessary apply SOC saturation between -5% &
73 105% as proposed by Plett Vol.2
74 % states(1) = min(1.05, max(states(1),-0.05));
75
76 %2.c) Error Covariance Estimation
77 estimatedP = (eye(size(predP,1)) -
78 KalGain*Chat)*predP;
79
80 %Update the next covariance matrix
81 prevP = estimatedP;
82
83 This function replicates the clip operation performed by the
84 corresponding Simulink block:
85
86 function valueExtrapolated =
87 extrapolationFcn(LUTbrk, LUTval, queryVal)
88
89 %LUTbrk = double containing the breakpoints;
90 %LUTval = values for the extrapolation;
91 %queryVal = SOC value for the extrapolation. Can
92 be either predicted or estimated;
93
94 %Output = valueExtrapolated
95
96 if queryVal < LUTbrk(1) %if SOC < 0 --> use
97 first value of LUTval
98 valueExtrapolated = LUTval(1);
99
100 elseif queryVal > LUTbrk(end) %if SOC > 1 -->
101 use last value of LUTval
102 valueExtrapolated = LUTval(end);
103
104 else
105 valueExtrapolated = interp1(LUTbrk,LUTval,
106 queryVal, 'linear', 'extrap');
107 end
108 end
109 end
```

Listing B.1: MATLAB EKF function for SOC estimation

# Bibliography

- [1] *Global EV Outlook 2025 – Analysis*. IEA. URL: <https://www.iea.org/reports/global-ev-outlook-2025> (visited on 08/24/2025) (cit. on p. 1).
- [2] *European Market Monitor: Cars and vans (June 2025)*. International Council on Clean Transportation. URL: <https://theicct.org/publication/european-market-monitor-cars-and-vans-june-2025/> (visited on 08/24/2025) (cit. on p. 1).
- [3] Boucar Diouf. “The electric vehicle transition”. In: *Environmental Science: Advances* 3.2 (2024), pp. 332–345. ISSN: 2754-7000. DOI: 10.1039/D3VA00322A. (Visited on 08/24/2025) (cit. on p. 1).
- [4] Dickson N. T. How, M. A. Hannan, M. S. Hossain Lipu, and Pin Jern Ker. “State of Charge Estimation for Lithium-Ion Batteries Using Model-Based and Data-Driven Methods: A Review”. In: *IEEE Access* 7 (2019), pp. 136116–136136. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2942213. (Visited on 08/24/2025) (cit. on pp. 1, 2).
- [5] Khiem Trad. *Lifecycle ageing tests on commercial 18650 Li ion cell @ 25°C and 45°C*. 2021. DOI: 10.4121/13739296.V1. URL: [https://data.4tu.nl/articles/\\_/13739296/1](https://data.4tu.nl/articles/_/13739296/1) (cit. on pp. 3, 9).
- [6] Yoido-Dong Youngdungpo-Gu. *Rechargeable Lithium Ion Battery Model: INR18650 MJ1 3500mAh*. Tech. rep. LG - Chem, 2016. URL: [https://power.tenenergy.com/content/datasheet/30708\\_datasheet.pdf](https://power.tenenergy.com/content/datasheet/30708_datasheet.pdf) (visited on 06/04/2025) (cit. on pp. 5, 7, 21).
- [7] Nigel. *LG INR 18650 MJ1*. Battery Design. Mar. 2023. URL: <https://www.batterydesign.net/lg-18650/> (visited on 06/06/2025) (cit. on p. 5).
- [8] Gregory L. Plett. *Battery management systems: Volume 1: battery modeling*. Artech House power engineering and power electronics. Boston: Artech House, 2015. ISBN: 978-1-63081-023-8 (cit. on pp. 6, 8, 10).



- [9] M.R. Jongerden and Boudewijn R.H.M. Haverkort. *Battery Modeling*. CTIT Technical Report Series TR-CTIT-08-01. University of Twente, Jan. 2008 (cit. on p. 7).
- [10] Marian Tomasov, Martina Kajanova, Peter Bracinik, and David Motyka. “Overview of Battery Models for Sustainable Power and Transport Applications”. In: *Transportation Research Procedia* 40 (2019), pp. 548–555. ISSN: 23521465. DOI: 10.1016/j.trpro.2019.07.079. (Visited on 06/06/2025) (cit. on p. 7).
- [11] Marcus Auch, Timo Kuthada, Sascha Giese, and Andreas Wagner. “Influence of Lithium-Ion-Battery Equivalent Circuit Model Parameter Dependencies and Architectures on the Predicted Heat Generation in Real-Life Drive Cycles”. In: *Batteries* 9.5 (May 17, 2023), p. 274. ISSN: 2313-0105. DOI: 10.3390/batteries9050274. (Visited on 06/07/2025) (cit. on p. 7).
- [12] Merve Tekin and M. Ihsan Karamangil. “Development of dual polarization battery model with high accuracy for a lithium-ion battery cell under dynamic driving cycle conditions”. In: *Heliyon* 10.7 (Apr. 2024), e28454. ISSN: 24058440. DOI: 10.1016/j.heliyon.2024.e28454. (Visited on 06/11/2025) (cit. on p. 8).
- [13] Davide Cittanti, Alessandro Ferraris, Andrea Airale, Sabina Fiorot, Santo Scavuzzo, and Massimiliana Carello. “Modeling Li-ion batteries for automotive application: A trade-off between accuracy and complexity”. In: *2017 International Conference of Electrical and Electronic Technologies for Automotive*. IEEE, June 2017, pp. 1–8. ISBN: 978-88-87237-26-9. DOI: 10.23919/EETA.2017.7993213. (Visited on 06/10/2025) (cit. on pp. 8, 9).
- [14] Sara Ha, Gabriele Pozzato, and Simona Onori. “Electrochemical characterization tools for lithium-ion batteries”. In: *Journal of Solid State Electrochemistry* 28.3 (Mar. 2024), pp. 1131–1157. ISSN: 1432-8488, 1433-0768. DOI: 10.1007/s10008-023-05717-1. (Visited on 06/10/2025) (cit. on pp. 9, 12, 14, 68).
- [15] Suguna Thanagasundram, Raghavendra Arunachala, Karl Löffler, Tanja Teutsch, and Andreas Jossen. “A Cell Level Model for Battery Simulation”. In: (Nov. 2012) (cit. on pp. 9, 14).
- [16] Carlo Taborelli and Simona Onori. “State of charge estimation using extended Kalman filters for battery management system”. In: *2014 IEEE International Electric Vehicle Conference (IEVC)*. Florence: IEEE, Dec. 2014, pp. 1–8. ISBN: 978-1-4799-6075-0. DOI: 10.1109/IEVC.2014.7056126. (Visited on 06/06/2025) (cit. on pp. 9, 27, 28, 68).

- [17] *Estimate Parameters and States - MATLAB & Simulink*. URL: [https://it.mathworks.com/help/slido/estimate-parameters-and-states.html?s\\_tid=CRUX\\_topnav](https://it.mathworks.com/help/slido/estimate-parameters-and-states.html?s_tid=CRUX_topnav) (visited on 06/18/2025) (cit. on p. 16).
- [18] *lsqnonlin - Solve nonlinear least-squares (nonlinear data-fitting) problems - MATLAB*. URL: <https://it.mathworks.com/help/optim/ug/lsqnonlin.html> (visited on 06/18/2025) (cit. on p. 16).
- [19] Min Chen and G.A. Rincon-Mora. “Accurate electrical battery model capable of predicting runtime and I-V performance”. In: *IEEE Transactions on Energy Conversion* 21.2 (2006), pp. 504–511. DOI: 10.1109/TEC.2006.874229 (cit. on pp. 24, 25).
- [20] Girisha Joshi, Lakshmi Narayana Valluru, and Amol Prakash Khade. “Two RC model and parameter estimation of lithium-ion battery”. In: *Indonesian Journal of Electrical Engineering and Computer Science* 37.2 (Feb. 1, 2025), p. 730. ISSN: 2502-4760, 2502-4752. DOI: 10.11591/ijeecs.v37.i2.pp730-739. (Visited on 06/23/2025) (cit. on p. 24).
- [21] Nikolaos Wassiliadis, Jörn Adermann, Alexander Frericks, Mikhail Pak, Christoph Reiter, Boris Lohmann, and Markus Lienkamp. “Revisiting the dual extended Kalman filter for battery state-of-charge and state-of-health estimation: A use-case life cycle analysis”. In: *Journal of Energy Storage* 19 (Oct. 2018). ISSN: 2352152X. DOI: 10.1016/j.est.2018.07.006. (Visited on 06/06/2025) (cit. on pp. 24, 28, 68).
- [22] Yidan Xu, Minghui Hu, Chunyun Fu, Kaibin Cao, Zhong Su, and Zhong Yang. “State of Charge Estimation for Lithium-Ion Batteries Based on Temperature-Dependent Second-Order RC Model”. In: *Electronics* 8.9 (Sept. 10, 2019), p. 1012. ISSN: 2079-9292. DOI: 10.3390/electronics8091012. (Visited on 06/06/2025) (cit. on pp. 24, 25).
- [23] Gregory L. Plett. *Battery management systems: Equivalent-Circuit Methods*. Artech House Power engineering series. Boston: Artech house, 2016. ISBN: 978-1-63081-027-6 (cit. on pp. 27, 28, 38).
- [24] M. Bercibar, I. Gandiaga, I. Villarreal, N. Omar, J. Van Mierlo, and P. Van Den Bossche. “Critical review of state of health estimation methods of Li-ion batteries for real applications”. In: *Renewable and Sustainable Energy Reviews* 56 (Apr. 2016). ISSN: 13640321. DOI: 10.1016/j.rser.2015.11.042. (Visited on 06/30/2025) (cit. on p. 27).
- [25] René Schneider and Christos Georgakis. “How To NOT Make the Extended Kalman Filter Fail”. In: *Industrial & Engineering Chemistry Research* 52.9 (Mar. 6, 2013), pp. 3354–3362. ISSN: 0888-5885, 1520-5045. DOI: 10.1021/ie300415d. (Visited on 07/19/2025) (cit. on p. 38).



- [26] Benedikt Rzepka, Simon Bischof, and Thomas Blank. “Implementing an Extended Kalman Filter for SoC Estimation of a Li-Ion Battery with Hysteresis: A Step-by-Step Guide”. In: *Energies* 14.13 (June 22, 2021), p. 3733. ISSN: 1996-1073. DOI: 10.3390/en14133733. (Visited on 07/21/2025) (cit. on p. 39).
- [27] *Generate Code and Executables for Individual Subsystems - MATLAB & Simulink*. URL: <https://it.mathworks.com/help/rtw/ug/generate-code-and-executables-for-an-individual-subsystem.html> (visited on 09/07/2025) (cit. on p. 57).
- [28] *What Is an S-Function? - MATLAB & Simulink*. URL: <https://it.mathworks.com/help/simulink/sfg/what-is-an-s-function.html> (visited on 10/03/2025) (cit. on p. 59).
- [29] Gasga Leo. *Lithium-Ion Battery 18650 3D Model - 3DCADBrower*. 2019. URL: <https://www.3dcadbrower.com/3d-model/lithium-ion-battery-18650> (visited on 06/06/2025) (cit. on p. 69).