CHALMERS
UNIVERSITY OF TECHNOLOGY

State of Charge Estimation for Battery Cells: A Sliding–Mode Approach

A comparative analysis of conventional and sliding–mode approaches to state of charge estimation in battery management system

Master's thesis in Systems, control and mechatronics

ANTON MARK
MARKUS NILSSON

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

State of Charge Estimation for Battery Cells: A Sliding–Mode Approach

A comparative analysis of conventional and sliding–mode approaches to state of charge estimation in battery management system

ANTON MARK
MARKUS NILSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2026

State of Charge Estimation for Battery Cells: A Sliding–Mode Approach
A comparative analysis of conventional and sliding–mode approaches to state of charge estimation in battery management system
ANTON MARK
MARKUS NILSSON

Supervisor: José Roberto Franco Jaramillo, Volvo Technology AB
Examiner: Torsten Wik, Department of Electrical Engineering

Master's Thesis 2026
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Volvo FH Electric next to charging station.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

State of Charge Estimation for Battery Cells: A Sliding–Mode Approach
A comparative analysis of conventional and sliding–mode approaches to state of charge estimation in battery management system
ANTON MARK
MARKUS NILSSON
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This thesis presents a comparative analysis of different state of charge (SOC) estimators in a battery management system. Different standard and sliding–mode algorithms to solve the robust SOC estimation problem are evaluated. An Equivalent Circuit Model (ECM) is presented for formulation purposes, with a parameterization based on a P2D model given by a high–fidelity simulator. The standard observers comprise the Luenberger and Extended Kalman Filter (EKF) algorithms, whereas the sliding–mode algorithms considered are first–order, second–order, and terminal–twisting algorithms. A tuning framework is proposed to find the optimal observer gains based on a global hyperparameter optimization algorithm. The results show, under an ideal scenario, that Kalman filtering provides a similar result to the best sliding–mode approaches. However, the sliding–mode algorithms present a higher robustness against nonlinear unmodeled dynamics, large parameter uncertainties, and measurement noise. Additionally, a computational effort analysis is performed, which demonstrates that the Luenberger observer is the least expensive algorithm followed by the sliding–mode approaches. Moreover, the results show that all the proposed algorithms are feasible; still, the best balance between robustness and computational effort is demonstrated by the first and second order sliding–mode observers followed by Luenberger observer. These statements are supported by the simulation results, confirming the feasibility of the proposed algorithms. To conclude, selecting an optimal state estimation technique requires a trade–off between accuracy, robustness, and computational efficiency.

Keywords: Battery Management System (BMS), State of Charge (SOC) estimation, Sliding Mode Observers (SMO), Extended Kalman Filter (EKF), Equivalent Circuit Model (ECM), PyBaMM, Robustness, Computational Complexity.

Acknowledgments

First and foremost, we would like to thank manager Olle Friberg and the EMOB BMS Control team for the opportunity to conduct this Master's thesis at Volvo GTT. We have received great advice, help, and a tremendous working environment. For this, we are truly grateful.

A special thank you goes out to our supervisor Roberto Franco. His tireless and invaluable assistance, expertise, and ideas have been crucial for the success of this thesis. He has been everything you could ask for in a supervisor – patient, encouraging, and most importantly very kind. Thank you for all the time you've invested in our work. We would also like to extend a thank you to Professor Torsten Wik, our examiner at Chalmers. He has analyzed our progress with a critical eye, thus improving the rigor of our work. Most importantly, he was non–hesitant and very kind to take on the role as our examiner. Thank you for this.

Anton Mark & Markus Nilsson, Gothenburg, June 2026

List of Acronyms

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

BEV	Battery Electric Vehicle
BMS	Battery Management System
BOL	Beginning–of–Life
CC	Coulomb Counting
CNTSMO	Continuous Non–Singular Terminal Sliding Mode Observer
CSTSMO	Continuous Singular Terminal Sliding Mode Observer
CTSMO	Continuous Twisting Sliding Mode Observer
DFN	Doyle–Fuller–Newman
ECM	Equivalent Circuit Model
EKF	Extended Kalman Filter
ESC	Enhanced Self–Correcting Circuit
EV	Electric Vehicle
FOSMO	First–Order Sliding Mode Observer
HPPC	Hybrid Pulse Power Characterization
ISS	Input–to–State Stable / Input–to–State Stability
KF	Kalman Filter
LFP	Lithium Iron Phosphate ($LiFePO_4$)
MAE	Maximum Absolute Error
OCV	Open Circuit Voltage
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PyBaMM	Python Battery Mathematical Modeling
RMSE	Root Mean Square Error
SMO	Sliding Mode Observer
SOC	State Of Charge
SOSMO	Second–Order Sliding Mode Observer
TPE	Tree–structured Parzen Estimator
TR	Thermal Runaway

Nomenclature

Below is the nomenclature that is used throughout this thesis.

Indices

i,j	Indices RC–parameters
k	Discrete–time step index
t	Index for time step

Distributions

$\mathcal{N}$	Normal distribution

Parameters

C_1, C_2	Polarization capacitances of the first and second RC branches
γ	Hysteresis change parameter
M	Maximum hysteresis voltage magnitude
Q	Nominal cell capacity
R_0	Ohmic internal resistance
R_1, R_2	Polarization resistances of the first and second RC branches
τ_1, τ_2	Time constants of the first and second RC branches
$\Delta t, T_s$	Discrete sampling time interval
η	Coulombic efficiency

Variables

$\mathbf{A}$	Linear system matrix
$\mathbf{B}$	Input mapping matrix
$\mathbf{B}_\nu$	Input vector for sliding mode state correction
$\mathbf{C}$	Output observation matrix
$\mathbf{D}$	Disturbance matrix
Δ	Bounded, unmodeled nonlinear system plant dynamics
e_y	Output voltage estimation error
e_z	State of Charge tracking error
$\mathbf{F}_k$	Discrete–time state transition Jacobian matrix
h	Unitless dynamic hysteresis state variable
$\mathbf{H}_k$	Discrete–time measurement Jacobian matrix
i	Applied load current signal
i_{R_1}, i_{R_2}	Currents passing through the resistor elements of the RC branches
$\mathbf{K}_k$	Kalman filter gain matrix
k_1, k_2, k_3, k_4	Tuning parameters for twisting sliding mode observers
$\mathbf{L}$	Linear observer feedback gain matrix
$\lambda_0, \lambda_1, \xi$	Tuning parameters for Exact Levant differentiator
ν, μ	Sliding mode internal output correction variables
$\mathcal{O}$	System observability matrix
$\mathbf{P}$	State estimation error covariance matrix
ϕ	Modeled nonlinear system dynamics
$\mathbf{Q}$	Process noise covariance matrix
R	Measurement noise covariance scalar
ρ, ρ_1, ρ_2	Switching feedback gains
s	Sliding manifold variable vector
σ_i	Standard deviation of the current sensor noise
σ_v	Standard deviation of the voltage sensor noise
v_1, v_2	Polarization voltages across the first and second RC branches
v_{oc}	Open–circuit voltage function
v_t, y	Terminal voltage output
$\mathbf{v}_k$	Zero–mean white Gaussian measurement noise vector
$\mathbf{w}, \mathbf{w}_1, \mathbf{w}_2$	Sliding mode internal switching injection variables

$\mathbf{w}_k$	Zero–mean white Gaussian process noise vector
$\mathbf{x}$	True system state vector
$\hat{\mathbf{x}}$	Observer estimated state vector
$\tilde{\mathbf{x}}$	State estimation error vector
z	True State of Charge of the battery cell
$\hat{z}_0$	Initial observer estimated State of Charge state
z_0	Initial true plant State of Charge state

Contents

List of Figures

List of Tables

1

Introduction

The world is transitioning away from fossil fuels [2] and the demand for battery electric vehicles (BEV) is increasing [3]. Although electric cars make up the largest contribution to this increase in demand, a similar trend can be seen in the electric truck segment. It is no surprise that company visions align with this trend, one example being Volvo Trucks aiming for a net–zero emission product range by 2040 [4].

It is well–known that the main source of energy for an electric vehicle (EV) is provided by the battery. Still, the battery demand comes with significant challenges regarding sustainability. The battery packs for EV applications are typically made of Lithium–ion (Li–ion) cells [5]. Li–ion batteries have become a popular choice for electric vehicles because of their advantageous properties, *i.e.*, low weight, high energy density, and the ability to be recharged for thousands of cycles. This drives an increase in the extraction of critical minerals such as lithium, cobalt, and nickel which are subject to high geographical concentration [6]. Improper administration can lead to ethical concerns due to unsafe working conditions and environmental degradation in mining regions. It is important to address the above–mentioned issues to ensure that the overall impact of battery technologies remains responsible and sustainable.

In addition, the batteries need to be safe during operation. To this aim, an accurate monitoring of internal states, such as temperature, voltage, and mechanical stress, of each individual cell is relevant to prevent events such as thermal runaway (TR) [7]. Still, it is worth highlighting that the longevity of a battery is fundamental to expand the applications and to increase the vehicle's performance; therefore, an accurate monitoring of internal states can help to extend the battery life expectancy [8].

1.1 Background

Batteries are the most expensive component in an electric truck. Thus, it is crucial to optimize their usage. Today, for a long–haul electric truck with a range of 500 km, the cost of the battery is almost half of the vehicle's purchase price [3]. As previously mentioned, accurate monitoring and control of the battery's internal states is essential for the safety, performance, and longevity of the battery pack. In this sense, the battery management system (BMS) plays an important role. The BMS

is responsible for an efficient, safe, and reliable operation of the plurality of battery cells onboard the vehicle [9].

One of the key features of the BMS is to estimate the state of charge (SOC). SOC represents the ratio between the remaining capacity in the cell and the nominal capacity. SOC cannot be measured directly from the cell. Thus, it is estimated based on onboard sensors. Li–ion batteries exhibit highly nonlinear and complex behavior, which makes accurate SOC estimation a challenging task. Moreover, as the cell ages, the chemistry involved provokes changes in the system parameters, and the capacity degrades in a nonlinear way.

There are different techniques used to estimate SOC. For instance, conventional estimation techniques such as the Extended Kalman Filter (EKF) or the Luenberger observer are widely used [10]. However, these conventional algorithms can be sensitive to unmodeled dynamics, parameter uncertainties, and external disturbances. To solve these problems, sliding–mode techniques is a possible solution.

1.2 Objective

The objective of this work has been to design, analyze, and validate different sliding–mode observer (SMO) techniques for SOC estimation in Li–ion battery cells. The proposed approach is a comparative analysis of different sliding–mode techniques against the conventional methods. To evaluate the performance of the presented algorithms, metrics for the following are proposed: robustness against parameter uncertainties, performance in the presence of nonlinear unmodeled dynamics, and tolerance against measurement noise. Furthermore, the study includes implementation aspects that need to be considered for BMS integration, *e.g.,* discretization, computational effort, and implementability.

1.3 Related work

SOC estimation for Li–ion battery cells is a well–established research topic, and several studies have addressed this matter. Different classes of SMOs have been implemented to solve the SOC estimation problem in Li–ion batteries, for instance, [11, p. 20], [12, p. 9], [13, p. 17], [14, p. 309], [15, p. 120]. Most of the research is focused on chattering attenuation and changes to the classic algorithms to improve the robustness against certain classes of disturbances. However, there is a lack of convergence proofs for these algorithms and the tuning is not properly explained.

In the context of implementability, most of the papers assume that the dynamic parameters are constants, *e.g.,* in [16] and [17]. However, it is well–known that the dynamic parameters of an electrical model are time–varying depending on the temperature, SOC, and C–rate. On the other hand, one of the most relevant factors is related to model (including disturbances) inaccuracies. Examples are external disturbances considered in the system, parameter uncertainties that appear together with the system states, and unmodeled nonlinear dynamics. All the previous ele-

ments appear in the system dynamics similarly to additive disturbances of different kinds. Most of the previously mentioned works only consider external disturbances. However, in the implementation, parameter uncertainties and nonlinear dynamics are often the most relevant inaccuracies in the system.

Besides the nonlinear time–varying parameters, another relevant factor is the observer tuning. Most of the papers do not present any tuning rule for the algorithms, which could lead to unfair comparisons. Moreover, it is worth to highlight that the gain tuning is fundamental for the chattering attenuation [18]. Therefore, to avoid the previous problem, a tuning framework needs to be introduced.

Finally, to evaluate the performance of the observers on different cell chemistries, most of the presented works considers a well–defined plant with known parameters [16], [17], [19], [20]. However, there is always a gap between implementation and an approximated model. In this sense, one of the best ways to evaluate the performance without cell testing is to use PyBaMM as baseline for simulating a state–of–the–art physics–based electrochemical model.

Based on the previous insights, the objective is to provide a comparison of different sliding–mode techniques following the algorithms with proper convergence proofs. Moreover, the cell parameterization with time–varying parameters depending on SOC and C–rates are obtained. Still, the algorithms are evaluated under different classes of parameter uncertainties, unknown nonlinear dynamics, and sensor noise. To obtain the best possible tuning, an optimization framework is built. Then, based on the baseline tuning, all the cases are analyzed. The simulation results show the feasibility of the proposed schemes using PyBaMM as the baseline.

1.4 Limitations

- Throughout the thesis development, there was no plan to apply the proposed algorithms to hardware; *i.e.*, there was no intention to use real batteries to validate the results. This restriction was due to the safety risks associated with experimental battery testing.
- The proposed SOC–estimation techniques were implemented and validated purely on cell level. Evaluating the performance across an entire battery pack is beyond the scope of this thesis.
- Furthermore, stochastic approaches or data–driven "black–box" methods, such as Neural Networks or Machine Learning algorithms, were excluded from this study. The thesis focuses exclusively on deterministic approaches from model–based control theory.

1.5 Specification of the issue being investigated

- How can an SMO be designed to estimate SOC for a Li–ion cell based on an equivalent circuit model (ECM)?

- How does the robustness of the proposed SMO compare to conventional state estimation methods, such as the Luenberger observer and the EKF, under parameter uncertainties and external disturbances?
- What are the computational and implementation considerations of integrating advanced and robust algorithms into a BMS?

2

Theory

This chapter establishes the theoretical foundation, structured as a progression from the core estimation problem to its algorithmic solutions. The primary target variable SOC, is defined in Section 2.2. SOC is an internal state that cannot be physically measured by sensors, thus requiring a state estimator. For this, a mathematical representation of the battery's dynamics is required, which is established in Section 2.3. Finally, the conventional observers and SM observers are introduced in Sections 2.5–2.6.

2.1 Preliminaries

This section introduces concepts that are fundamental to the theory related to the observers.

2.1.1 Observability

The following definition and theorem are valid for linear systems described by the state–space representation

$$\dot{x} = Ax + Bu,$$
$$y = Cx + Du,$$

where x is the state vector, y is the output vector, A is the state matrix, B is the input matrix, C is the output matrix, and D is the feedthrough matrix.

Definition 1 ([**21, p. 45**]). *The state $x^* \neq 0$ is said to be **unobservable** if, when $u(t) = 0, t \geq 0$ and $x(0) = x^*$, the output is $y(t) \equiv 0,\ t \geq 0$. The system is said to be **observable** if it lacks unobservable states.*

The following theorem can be used to assess observability:

Theorem 1 ([**21, p. 45**]). *The unobservable states constitute a linear subspace, viz, the null space of the matrix (the observability matrix)*

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

The system is thus observable if and only if $\mathcal{O}$ has full rank.

2.1.2 Detectability

The following definition and theorem are valid for linear systems.

Definition 2 ([**21, p. 45**]). *A system (A, B, C) is said to be detectable if there exists a matrix K, so that $A - KC$ is stable.*

2.1.3 Input–to–state stability

Definition 3 ([**22, p. 175**]). *The system $\dot{x} = f(t, x, u)$ where $f : [0, \infty) \times R^n \times R^m \to R^n$ is piecewise continuous in t is locally Lipschitz in x and u, the input $u(t)$ is a piecewise continuous, bounded function of t for all $t \geq 0$, is said to be input–to–state stable if there exist a class $\mathcal{KL}$ function β and a class $\mathcal{K}$ function γ such that for any initial state $x(t_0)$ and any bounded input $u(t)$, the solution $x(t)$ exists for all $t \geq t_0$ and satisfies*

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0) + \gamma\left(\sup_{t_0 \leq \tau \leq t} \|u(\tau)\|\right)$$

2.2 State of charge

SOC is defined to be 100% when the cell is fully charged and 0% when fully discharged. Thus, it follows that SOC $\in [0, 100]\%$ or SOC $\in [0, 1]$. From an electrochemical perspective, SOC is directly related to the average lithium–ion concentration within the negative–electrode (anode) solid particles [23]. However, measuring the lithium concentration is infeasible in real–world applications. Instead, SOC must be estimated based on external variables such as terminal voltage, current, and temperature. The SOC estimation is further complicated by the battery's nonlinear and time–varying characteristics. The simplest way to estimate SOC is by Coulomb counting (CC), which calculates the remaining capacity by integrating the load current over time, *i.e.*,

$$\mathrm{SOC}(t) = \mathrm{SOC}(t_0) - \frac{\eta}{Q}\int_{t_0}^{t} i(\tau)d\tau, \tag{2.1}$$

where $\mathrm{SOC}(t_0)$ is the initial state, Q is the nominal capacity, $i(\tau)$ is the current, and η is the coulombic efficiency. However, CC is an open–loop method. Thus, it suffers from cumulative errors due to sensor noise, it requires a precise initial value, and is susceptible to capacity errors. To achieve higher precision, which is required for most EV applications, a model–based observer is a feasible solution [10].

2.3 Cell modeling

For model–based approaches, one of the most relevant requirements is the emulated dynamics. The complex nature of the batteries makes ECM the preferred approach

over electrochemical models. ECM is based on electrical components to approximate the real cell dynamics. The ECM is the most common approach to model batteries in BMS applications [24, p. 29] and can be formulated as a system of ordinary differential equations (ODE).

2.3.1 Equivalent–circuit model

The ECM is composed of different states, for instance, SOC, polarization voltages, and hysteresis to produce the signal output, *i.e.*, the voltage measured at the terminals of the cell. A model capable of producing a feasible output with a proper accuracy is the enhanced self–correcting model (ESC) proposed in [24, p. 44]. In this derivation, the current is defined as positive for discharge and negative for charge.

An ECM with two resistor–capacitor (RC) pairs is shown in Figure 2.1. Adding RC circuits improves accuracy [25, p. 33] but also increases complexity.

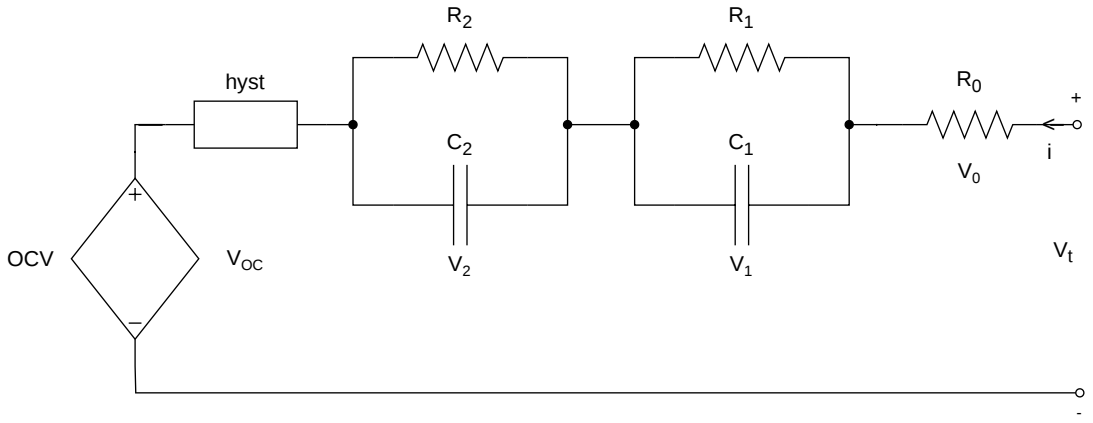


Figure 2.1: A second order ECM circuit.

The continuous–time equations for the circuit in Figure 2.1 are derived using Kirchhoff's circuit laws. The voltage across the RC–circuits are denoted as v_1 and v_2, respectively. The hysteresis component is denoted h, the open-circuit voltage (OCV) is denoted v_{OC}, and the voltage across the ohmic resistance is denoted v_0. The terminal voltage is then defined as follows

$$v_t(t) = v_{oc}(z(t)) - Mh(t) - v_1(t) - v_2(t) - v_0(t), \tag{2.2}$$

where M is the hysteresis voltage magnitude and $z(t)$ is the SOC. For simplicity, using current as state is preferred over voltage for empirical parameter identification [24, p. 35]. Hence, Equation (2.2) can be reformulated as

$$v_t(t) = v_{oc}(z(t)) - Mh(t) - R_1 i_1(t) - R_2 i_2(t) - R_0 i(t).$$

The input current $i(t)$ is equal to the sum of the currents passing through the resistance and the capacitor of an RC–circuit. This means that the current through an RC–circuit can be written

$$i(t) = i_{R_j}(t) + C\dot{v}_{C_j}(t), \qquad j = 1, 2,$$

where the voltage over the capacitor is

$$v_{C_j} = R_j i_j,$$

meaning that the change in current i_{R_j} can be defined by

$$\frac{di_{R_j}(t)}{dt} = -\frac{1}{R_j C_j} i_{R_j}(t) + \frac{1}{R_j C_j} i(t).$$

The change in SOC can be derived by differentiation of Equation (2.1), *i.e.*,

$$\frac{dz(t)}{dt} = \frac{-\eta i(t)}{Q}.$$

The change in hysteresis can be modeled as

$$\frac{dh(t)}{dt} = -\left|\frac{\eta(t) i(t) \gamma}{Q}\right| h(t) + \left|\frac{\eta(t) i(t) \gamma}{Q}\right| M(z, \dot{z}),$$

where γ is the hysteresis change rate [24, p. 43]. The hysteresis voltage magnitude can be modeled by

$$M(z, \dot{z}) = -M \operatorname{sgn}(i(t)).$$

For parameter identification purposes it is useful to have a unitless hysteresis state and instead multiplying it by M in the output equation to get the hysteresis voltage [24, p. 43]. Then, the change in hysteresis is modeled by

$$\frac{dh(t)}{dt} = -\left|\frac{\eta(t) i(t) \gamma}{Q}\right| h(t) + \left|\frac{\eta(t) i(t) \gamma}{Q}\right| \operatorname{sgn}(i(t)),$$

and the hysteresis voltage becomes

$$\text{Hysteresis voltage} = Mh(t).$$

2.3.1.1 State–space model

To implement the ECM in a digital BMS, the continuous–time ODEs are either transformed into a discrete–time state space model or a continuous–time state space model that is discretized using *e.g.*, the explicit Euler method. For a second–order ESC model, the state vector keeps track of the SOC (z), the currents passing through the two resistor–capacitor branches (i_{R1} and i_{R2}), and the dynamic hysteresis state (h). The continuous state–space model can then be formulated as

$$\underbrace{\frac{d}{dt}\begin{bmatrix} z \\ i_{R_1} \\ i_{R_2} \\ h \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{\tau_1} & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_2} & 0 \\ 0 & 0 & 0 & -\gamma\left|\frac{\eta}{Q} i\right| \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} z \\ i_{R_1} \\ i_{R_2} \\ h \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \frac{\eta}{Q} \\ \frac{1}{\tau_1} \\ \frac{1}{\tau_2} \\ \gamma\left|\frac{\eta}{Q}\right| \end{bmatrix}}_{\mathbf{B}} i(t), \tag{2.3}$$

$$v(t) = v_{OC} + Mh + R_0 i + R_1 i_1 + R_2 i_2, \tag{2.4}$$

where $\tau_1 = R_1 C_1$ and $\tau_2 = R_2 C_2$. The discrete state space model is then formulated as

$$\underbrace{\begin{bmatrix} z_{k+1} \\ i_{R_1,k+1} \\ i_{R_2,k+1} \\ h_{k+1} \end{bmatrix}}_{\mathbf{x_{k+1}}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \exp\left(\frac{-\Delta t}{\tau_1}\right) & 0 & 0 \\ 0 & 0 & \exp\left(\frac{-\Delta t}{\tau_2}\right) & 0 \\ 0 & 0 & 0 & \exp\left(-\left|\frac{\eta_k i_k \gamma \Delta t}{Q}\right|\right) \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} z_k \\ i_{R_1,k} \\ i_{R_2,k} \\ h_k \end{bmatrix}}_{\mathbf{x_k}} + \underbrace{\begin{bmatrix} \frac{\eta_k \Delta t}{Q} i_k \\ \left(1 - \exp\left(\frac{-\Delta t}{\tau_1}\right)\right) \cdot i_k \\ \left(1 - \exp\left(\frac{-\Delta t}{\tau_2}\right)\right) \cdot i_k \\ \left(1 - \exp\left(-\left|\frac{\eta_k i_k \gamma \Delta t}{Q}\right|\right)\right) \cdot \operatorname{sgn}(i_k) \end{bmatrix}}_{\mathbf{B}}, \quad (2.5)$$

see for example [24, p. 44] and [25, p. 32]. The output equation

$$v_k = v_{OC,k} + M h_k + R_0 i_k + R_1 i_{1,k} + R_2 i_{2,k}, \quad (2.6)$$

differs from the one proposed by Plett [24] as it does not contain a term for instantaneous hysteresis voltage. Note also that the current direction is reversed in the Model Formulations (2.3)–(2.6). Charge current is defined as positive, *i.e.*, $i > 0$, and discharge current is defined as negative, *i.e.*, $i < 0$.

2.3.1.2 Cell characterization

The most critical factor for the ECM accuracy is the parameterization. Therefore, it is important to have extensive cell tests with advanced protocols to derive a good set of parameters. To capture the static and dynamic behavior, two different experimental protocols are required [24, p. 45]. The OCV defines the static relationship between the cell's voltage and its SOC at equilibrium. To isolate the true OCV from the dynamic effects, the cell is subjected to a slow continuous discharge and charge cycles, typically at a $C/30$ rate [24, pp. 45–46] or lower. The final OCV curve is approximated by taking the average of voltages for the discharge and charge curves, which reduces the effect of remaining ohmic drops and hysteresis effects. Moreover, it is possible to obtain the static parameters M, Q, and η by various post–processing procedures.

To identify the dynamic parameters of the ECM ($R_0, R_1, C_1, R_2, C_2, \gamma$), a dynamic test is required. A test protocol includes a current profile that excites the internal dynamics over the cell's entire operating range. This is usually achieved by using standardized, highly dynamic load profiles or pulse tests, such as High Power Pulse Characterization (HPPC) [24, p. 53].

2.3.2 Electrochemical model

Electrochemical battery models provide an advanced description of a cell's internal physical and chemical processes, offering a more accurate and informative model of the cell than ECMs [26]. One of the most established ones is the Doyle–Fuller–Newman (DFN) model, commonly referred to as the pseudo–two–dimensional (P2D) model [27]. Based on the fundamental physics of the battery's materials and liquid electrolyte, the DFN model tracks how the system operates on both a large and small scale. It calculates the flow of mass and electricity across the whole cell, as well as how lithium diffuses into the individual particles of the active electrode

materials [28]. Because the DFN model relies on these fundamental physics, similar to the real cell, it can capture the nonlinear dynamics and internal states of the battery under different operating conditions. However, solving these complex coupled partial differential equations (PDEs) is highly computationally demanding, making the full DFN model generally unsuitable for real–time onboard execution in an EV [29]. Instead of using the model in the BMS, it can be used as the simulated "true battery" due to its high fidelity. The DFN model is implemented in Python Battery Mathematical Modeling (PyBaMM), an open–source framework specifically designed to discretize and solve these underlying PDEs efficiently [30].

2.4 Model–based estimation

By combining the input and output (current and voltage) from the cell model, SOC can be estimated using a model–based observer. Unlike CC, a model–based observer is based on a closed–loop feedback mechanism to correct the estimation error caused by drift and initialization errors [25, pp. 77–79].

As illustrated in Figure 2.2, the system input is supplied into both the physical battery (the true system) and the mathematical model. The observer then compares the *measured output* (y) from the sensors against the *predicted output* ($\hat{y}$) generated by the model. The discrepancy between these two values is defined as the estimation error $e = y - \hat{y}$, which is used as a correction signal to improve the estimated state $\hat{\mathbf{x}}$ [25, pp. 77–79].

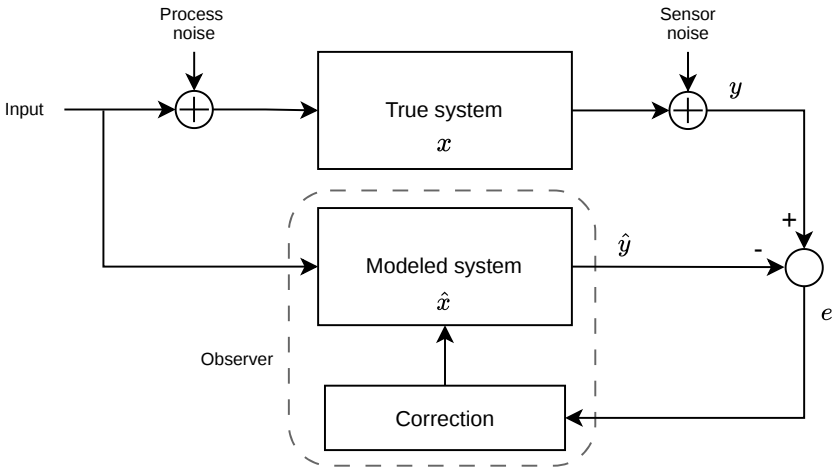


Figure 2.2: Schematic diagram of a model–based estimation system.

2.5 Conventional methods

Model–based state estimation traditionally relies on filtering architectures designed to handle linear dynamics. This section introduces two examples, the Luenberger observer, which represents the simplest linear feedback observer, and the EKF. These

conventional filtering models will serve as the primary benchmarks to evaluate the SOC estimation of the proposed sliding–mode techniques.

2.5.1 Luenberger observer

One of the most fundamental approaches to state estimation for linear systems is the Luenberger observer, which utilizes a closed–loop feedback mechanism from the measured output [31, p. 207]. The model's predicted output ($\hat{\mathbf{y}}$) is compared to the actual measured output ($\mathbf{y}$). The difference is multiplied by an observer gain matrix $\mathbf{L}$ to provide a proportional feedback term.

Consider a linear system

$$\dot{\mathbf{x}} = \mathbf{A}x + \mathbf{B}\mathbf{u}, \tag{2.7a}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x}. \tag{2.7b}$$

Then, a Luenberger observer for (2.7) is defined as

$$\begin{aligned} \frac{d\hat{\mathbf{x}}}{dt} &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \hat{\mathbf{y}}), \\ \hat{\mathbf{y}} &= \mathbf{C}\hat{\mathbf{x}}. \end{aligned}$$

If the state estimation error is defined as $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$, then the error dynamics is given by

$$\frac{d\tilde{\mathbf{x}}}{dt} = (\mathbf{A} - \mathbf{L}\mathbf{C})\tilde{\mathbf{x}}.$$

By appropriately choosing the observer gain $\mathbf{L}$, the eigenvalues of the matrix ($\mathbf{A} - \mathbf{L}\mathbf{C}$) can be assigned to have negative real parts. Consequently, the observer error dynamics are stable, ensuring asymptotic convergence to zero of the state estimation error [31, p. 207].

2.5.2 Kalman filter

A linear discrete–time, time–invariant system, with zero mean white Gaussian noise in the input and output can be written as

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \\ \mathbf{y}_k &= \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \end{aligned} \tag{2.9}$$

An optimal observer design for (2.9) is given by the Kalman Filter (KF) [21, pp.123-131]. The first step for the KF is the prediction step given as

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{A}_k\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k, \\ \mathbf{P}_{k|k-1} &= \mathbf{A}_k\mathbf{P}_{k-1|k-1}\mathbf{A}_k^\top + \mathbf{Q}_k, \end{aligned} \tag{2.10}$$

and an update step

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{C}_k^\top \left(\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^\top + \mathbf{R}\right)^{-1}, \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{x}}_{k|k-1}), \\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\mathbf{P}_{k|k-1}. \end{aligned}$$

The KF guarantees asymptotic convergence to zero of the state estimation error. However, this requires a precise knowledge of the system dynamics.

2.5.3 Extended Kalman filter

The EKF is essentially a regular KF applied on a nonlinear system that is linearized around the current state estimate in each time step [21, p. 380]. The structure remains the same, but the model formulation changes to a nonlinear version, *i.e.*,

$$\begin{aligned}\mathbf{x}_k &= f(\hat{\mathbf{x}}_{\mathbf{k-1}}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}, \quad \mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}) \\ \mathbf{y}_k &= h(\mathbf{x}_k) + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k).\end{aligned}$$

The prediction step changes to

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}), \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_{k-1},\end{aligned}$$

and the update step becomes

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \left(\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k \right)^{-1}, \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) \right), \\ \mathbf{P}_{k|k} &= \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_{k|k-1},\end{aligned}$$

where

$$\begin{aligned}\mathbf{F}_k &= \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}}, \\ \mathbf{H}_k &= \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}}.\end{aligned}$$

Although the EKF is a non–linear filtering method, it is only applicable to systems with mild non–linearities with Gaussian process noise in the input and output. Still, its advantage lies in its good performance despite being a relatively simple filtering method [32, p. 116].

2.6 Sliding–mode

Sliding–mode, originally developed by Utkin [33], is a nonlinear control technique that utilizes a discontinuous, high–frequency switching signal to drive the system's state trajectories onto a predefined sliding surface. The dynamics of the system are divided into two distinct stages: the reaching phase, where the system trajectories are forced to the surface, and the sliding phase, where the system is constrained to the surface and slides towards the origin [11, p. vii]. The primary advantage is that, once the system is in the sliding phase, its dynamic behavior is robust to some class of model uncertainties and external disturbances [11, p. 6].

The sliding–mode concept can be applied to both control and state estimation problems, where the latter is named as sliding–mode observer (SMO). Instead of relying solely on linear feedback like a standard Luenberger observer, an SMO injects a discontinuous switching term based on the estimation error [11, pp. 105–106]. The SMOs that follow are based on a linear state–space formulation, which will be further discussed in Chapter 3.

2.6.1 First–Order SMO

The simplest example of an SMO is the first–order SMO (FOSMO) with the following observer dynamics,

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \mathbf{L}(e_y) + \rho\,\mathrm{sgn}(e_y), \tag{2.12a}$$

$$\hat{y} = \mathbf{C}\hat{\mathbf{x}}, \tag{2.12b}$$

where the output estimation error is defined as $e_y = \hat{y} - y$. In this observer, the linear gain matrix $\mathbf{L}$ is responsible for stabilizing the known part of the system, functioning like a standard Luenberger observer. However, it is the nonlinear switching term multiplied by ρ that gives the observer its robustness properties. By designing the gain (ρ) to be strictly greater than the maximum bound of any modeling uncertainties and external disturbances, the observer mathematically forces the output estimation error to zero in finite time. Once the output error is driven to zero ($e_y = 0$), the observer states converge asymptotically to the true system states [11, p. 105-107].

The theoretical advantages make the FOSMO suitable for practical applications characterized by nonlinearities and unmodeled dynamics. On the other hand, FOSMOs suffer from a phenomenon called chattering due to the discontinuous nature of the correction term $\mathrm{sgn}(e_y)$ [11, pp. 106–115]. The chattering is an undesired effect and might be problematic depending on the length of the discretization time–step, tuning, and model formulation.

2.6.2 Second–Order SMO

High–frequency discontinuous switching signal is problematic from the user perspective if a continuous action is required. To this aim, the second–order SMO (SOSMO) can be a solution [11, p. 34]. The SOSMO is also called super–twisting SMO, and provides finite–time convergence of the output estimation error [11, p. 253]. Moreover, it solves the chattering problem by adding the high–frequency signal as an integrated value in the nonlinear output injection. However, this adds a new assumption to the disturbances: the disturbances must be continuous, and their derivative

should exist. The SOSMO has the following form [12, p. 9]:

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \mathbf{w}, \tag{2.13a}$$

$$\mathbf{w} = \mathbf{w}_1 + \mathbf{w}_2, \tag{2.13b}$$

$$\mathbf{w_1} = \rho_1 \left|e_y\right|^{\frac{1}{2}} \operatorname{sgn}(e_y), \tag{2.13c}$$

$$\dot{\mathbf{w}}_2 = \rho_2 \operatorname{sgn}(e_y). \tag{2.13d}$$

The derivative of the switching injection term $\mathbf{w}_2$ is integrated, and the discontinuity is removed. Therefore, the injection term is smooth. Note that the SOSMO has two tuning parameters (ρ_1 and ρ_2) acting on each correction term ($\mathbf{w}_1$ and $\mathbf{w}_2$) respectively.

2.6.3 Twisting SMOs

The continuous singular/non–singular terminal SMO (CSTSMO/CNTSMO) and the continuous twisting SMO (CTSMO) are based on the error dynamics and its derivative. By explicitly including the error derivative in the sliding variable, the twisting algorithms can ensure finite–time convergence to zero for both the sliding variable and its derivative, *i.e.,* $e_y = \dot{e}_y = 0$, in the absence of noise.

Therefore, as the derivative of the error is known, the algorithms are capable of attenuating a wider class of inaccuracies that can affect the error dynamics.

Introducing the error derivative $\dot{e}_y$ can lead to several issues if a nominal numerical method is used for its calculation. A nominal numerical method can amplify the signal noise from the sensor causing issues in the estimation algorithm. To solve the previous problem, filtering differentiators provide an exact robust derivative in finite time despite noise in the measurement signals [34, pp. 1–3].

2.6.3.1 Continuous singular terminal SMO

The CSTSMO applied to an LTI system has the following form [13, p. 15]

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \mathbf{B}_\nu \nu, \tag{2.14a}$$

$$s = \dot{e}_y + k_1 \lfloor e_y \rceil^{2/3}, \tag{2.14b}$$

$$\nu = \mu + k_2 \lfloor s \rceil^{1/2}, \tag{2.14c}$$

$$\dot{\mu} = k_3 \lfloor s \rceil^{0}, \tag{2.14d}$$

where $\mathbf{B}_\nu$ is a column vector to decide which states are corrected, $\lfloor s \rceil^\alpha = |s|^\alpha \operatorname{sgn}(s)$, and $k_1, k_2, k_3 > 0$ are tuning parameters. The singularity appears in the power term of the sliding surface. Note that the sliding surface and its derivative must be driven to zero, thus, the derivative of the power term $\lceil e_y \rfloor^{2/3}$ is given by $k e_y^{-1/3}$ where k is a constant value. Then, as the estimation error approaches zero, the derivative of the surface creates a singularity. For implementation purposes, there exist several ways to protect the behavior of these dynamics, so that it is implementable.

2.6.3.2 Continuous non–singular terminal SMO

Similar to CSTSMO, the CNTSMO applied to an LTI system is given by [14, p.309]

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \mathbf{B}_\nu \nu, \tag{2.15a}$$
$$s = \dot{e}_y + k_1 \lfloor e_y \rceil^{3/2}, \tag{2.15b}$$
$$\nu = \mu + k_2 \lfloor s \rceil^{1/3}, \tag{2.15c}$$
$$\dot{\mu} = k_3 \lfloor s \rceil^{0}. \tag{2.15d}$$

2.6.3.3 Continuous twisting SMO

The CTSMO applied to an LTI system has the following form [15, p.120]

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \mathbf{B}_\nu \nu, \tag{2.16a}$$
$$\nu = \mu + k_1 \lfloor e_y \rceil^{1/3} + k_2 \lfloor \dot{e}_y \rceil^{1/2}, \tag{2.16b}$$
$$\dot{\mu} = k_3 \lfloor e_y \rceil^{0} + k_4 \lfloor \dot{e}_y \rceil^{0}. \tag{2.16c}$$

2.6.4 Lyapunov stability

Control theory relies heavily on Lyapunov stability, and sliding–mode theory is no exception. Lyapunov function techniques are employed to prove finite–time convergence of the sliding variables.

The choice of Lyapunov function depends on the observer, ranging from a simple quadratic function for FOSMOs to homogeneous or nonlinear functions for SOSMOs and CTSMOs, respectively.

The Lyapunov stability proof for first–order sliding–mode controllers can be found in [11, pp. 4–5], and the proof is analogous to that of SMOs. The stability proof for the SOSMO can be found in [12, pp. 9–12], for the CSTSMO in [13, pp. 19–25], for the CNTSMO in [14, pp. 310–311], and for the CTSMO in [15, pp. 120–124].

If the system satisfies the conditions given in the previously mentioned papers, the stated convergence proofs are applicable, and the stability properties are inherited. Note that the solutions of the differential equations with discontinuous right–hand sides are understood in the Filippov sense [35].

2.6.5 Detectability

If the system is input–to–state stable (ISS) according to the definition in Section 2.1, detectability is ensured which is the minimum requirement for the proposed observers. Moreover, an observable (see Section 2.1.1) system is always detectable.

3

Methods

This chapter outlines the methodology used to design, optimize, and evaluate the proposed estimation algorithms. The mathematical derivation of the continuous ECM state–space formulation is presented in Section 3.1. The high–fidelity DFN model simulated via PyBaMM was used as the baseline plant to parametrize the ECM shown in Section 3.1.2, and the implementation of the state observers is shown in Section 3.2. Also, an automated hyperparameter tuning for objective comparison of the observers is given in Section 3.3. Finally, the validation framework and test procedure are presented in Section 3.4.

3.1 Cell modeling

The cell modeling process can be divided into three different stages. First, the state–space model is derived according to the proposed ECM.

3.1.1 State space modeling

A state–space model for the conventional methods was presented in Equation (2.5). However, for the sliding–mode techniques, it is necessary to introduce a new formulation based on the same cell equations. The main difference lies in the state formulation. The model presented in Equation (2.5) depends on the current, and the output is a combination of the states. In the formulation used for the sliding–mode techniques, the terminal voltage needs to be included as part of the state to estimate. Thus, the output will be re–defined based on the new state. The continuous–time current state–space model can be seen in Equations (2.3) to (2.4) and the discrete version can be seen in Equations (2.5) to (2.6). The voltage model is derived as follows:

Let us assume that the OCV can be modeled as

$$v_{oc} = az + f(z), \tag{3.1}$$

where $f(z)$ is the non–linear relationship between OCV and SOC. Then, the change

in terminal voltage v_t can be written as follows

$$\begin{aligned}\dot{v}_t &= \dot{v}_{oc} + \dot{v}_1 + \dot{v}_2 \\ &= \frac{\partial v_{oc}}{\partial z}\dot{z} - \frac{1}{\tau_1}v_1 + \frac{R_1}{\tau_1}i - \frac{1}{\tau_2}v_2 + \frac{R_2}{\tau_2}i + \Delta \\ &= \frac{\partial v_{oc}}{\partial z}\dot{z} - \frac{1}{\tau_1}(v_t - v_{oc} - v_2 - iR_0) \\ &\quad - \frac{1}{\tau_2}(v_t - v_{oc} - v_1 + iR_0) + \left(\frac{R_1}{\tau_1} + \frac{R_2}{\tau_2}\right)i + \Delta,\end{aligned}$$

where Δ is unmodeled nonlinear dynamics that appear as a consequence of the assumption in Equation (3.1). Contrary to the model where currents are used as states, the hysteresis also appear as an external disturbance. SOC can be reformulated as follows

$$\begin{aligned}\dot{z} &= \frac{\eta}{Q}i, \\ &= \frac{\eta}{QR_0}\left(v_t - v_{oc} - v_1 - v_2\right).\end{aligned}$$

The continuous–time voltage formulation becomes

$$\underbrace{\frac{d}{dt}\begin{bmatrix} v_t \\ z \\ v_1 \\ v_2 \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} -(\frac{1}{\tau_1}+\frac{1}{\tau_2}) & 0 & \frac{1}{\tau_2} & \frac{1}{\tau_1} \\ \frac{\eta}{QR_0} & 0 & -\frac{\eta}{QR_0} & -\frac{\eta}{QR_0} \\ 0 & 0 & -\frac{1}{\tau_1} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_2} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} v_t \\ z \\ v_1 \\ v_2 \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \frac{R_1}{\tau_1}+\frac{R_2}{\tau_2}+\frac{R_0}{\tau_1}+\frac{R_0}{\tau_2} \\ 0 \\ \frac{R_1}{\tau_1} \\ \frac{R_2}{\tau_2} \end{bmatrix}}_{\mathbf{B}} i(t) + \underbrace{\begin{bmatrix} \left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)v_{oc} + \frac{dv_{oc}(z)}{dz}\frac{\eta}{Q}i \\ -\frac{\eta}{QR_0}v_{oc} \\ 0 \\ 0 \end{bmatrix}}_{\phi} + \underbrace{\begin{bmatrix} \Delta \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{D}}, \quad (3.2)$$

$$y = v_t.$$

As can be noted in Equation (3.2), the second column of the matrix $\mathbf{A}$ consists entirely of zeros. This indicates that the SOC gives no linear feedback to any of the other states. Consequently, the observability matrix of the linear parts of the system $\mathcal{O}$ is rank deficient ($\text{rank}(\mathcal{O}) < 4$). The linear part of this formulation being unobservable implies a problem in satisfying the conditions given in different linear observer methods, such as pole placement.

To resolve this limitation without altering the underlying physical equations of the cell, algebraic restructuring is used. This is purely a theoretical manipulation used to obtain observability for the linear part of the system. A linear factor of 10 is introduced to populate the column of zeros where $10z$ is first subtracted in the $\mathbf{Ax}$ operation and then added in ϕ for the $\dot{v}_t$ and $\dot{z}$ state equations, respectively. Because these two modifications perfectly cancel each other out, the underlying system remains unchanged while forcing the linear system matrix to become fully observable. Moreover, the systems satisfy the conditions mentioned in Section 2.6.5. Thus, the systems as a whole are detectable. The modified, observable state–space configuration is formulated as:

$$\underbrace{\frac{d}{dt}\begin{bmatrix} v_t \\ z \\ v_1 \\ v_2 \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} -(\frac{1}{\tau_1}+\frac{1}{\tau_2}) & -10 & \frac{1}{\tau_2} & \frac{1}{\tau_1} \\ \frac{\eta}{QR_0} & -10 & -\frac{\eta}{QR_0} & -\frac{\eta}{QR_0} \\ 0 & 0 & -\frac{1}{\tau_1} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_2} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} v_t \\ z \\ v_1 \\ v_2 \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \frac{R_1}{\tau_1}+\frac{R_2}{\tau_2}+\frac{R_0}{\tau_1}+\frac{R_0}{\tau_2} \\ 0 \\ \frac{R_1}{\tau_1} \\ \frac{R_2}{\tau_2} \end{bmatrix}}_{\mathbf{B}} i(t) + \underbrace{\begin{bmatrix} \left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)v_{oc} + \frac{dv_{oc}(z)}{dz}\frac{\eta}{Q}i + 10z \\ -\frac{\eta}{QR_0}v_{oc} + 10z \\ 0 \\ 0 \end{bmatrix}}_{\phi} + \underbrace{\begin{bmatrix} \Delta \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{D}}, \quad (3.3)$$

$$y = v_t.$$

which can be discretized and simulated using explicit Euler integration. The matrix ϕ represents the modeled nonlinear dynamics.

As it is previously stated, the purpose of the model formulation given in Equation (3.3) was to propagate the nonlinear terminal voltage injection throughout all the states. By using the previous formulation, with the terminal voltage correction, the system is capable of correcting the SOC.

3.1.2 Parametrization

Due to a lack of resources, the parameterization for the ECM is based on a high–fidelity battery simulator, PyBaMM. PyBaMM is then represented as the real battery system.

A virtual testing environment using PyBaMM is set up using the standard settings. DFN is used as a model. The electrochemical parameters for the DFN model are the ones proposed in "Prada2013", which characterizes a commercial 2.3 Ah Lithium Iron Phosphate (LFP, $LiFePO_4$) cylindrical cell [1]. A summary of the key cell specifications utilized in the simulation is shown in Table 3.1.

Table 3.1: Key electrochemical parameters for the Prada2013 $LiFePO_4$ cell set [1].

Parameter Description	**Value**	**Unit**
Nominal cell capacity	2.3	Ah
Upper voltage cut–off	3.6	V
Lower voltage cut–off	2.0	V
Negative electrode material	Graphite	-
Positive electrode material	LFP	-
Negative electrode thickness	3.4×10^{-5}	m
Positive electrode thickness	8.0×10^{-5}	m
Separator thickness	2.5×10^{-5}	m
Initial concentration in electrolyte	1200	$\text{mol} \cdot \text{m}^{-3}$
Reference temperature	298	K

To extract the necessary data for ECM parameterization, two simulated experiments were conducted by defining custom input current profiles based on Volvo protocols [36].

First, to establish the static relationship between the OCV and SOC, an OCV test was simulated. The resulting OCV–SOC curve, illustrating the characteristic wide voltage plateau of the LFP chemistry, is presented in Figure 3.1. The static relationship is utilized as a 1D look–up table within the observer implementations.

The DFN used here has identical OCV–SOC curves for charge and discharge, meaning hysteresis is not being considered. However, the OCV–test protocol provided by Volvo interpolates the OCV samples for the charge and discharge OCV curve. Hence, there is a discrepancy between the curves which is captured by the maximum hysteresis voltage magnitude parameter. This effect is explicitly modeled in the current as a state, while it appears as an external disturbance in the voltage formulation.

A Hybrid Pulse Power Characterization (HPPC) profile was applied to capture the dynamic behavior of the cell, together with a fitting framework given by Volvo (the details are presented in the following section). All PyBaMM simulations were performed assuming a constant temperature of 25°C.

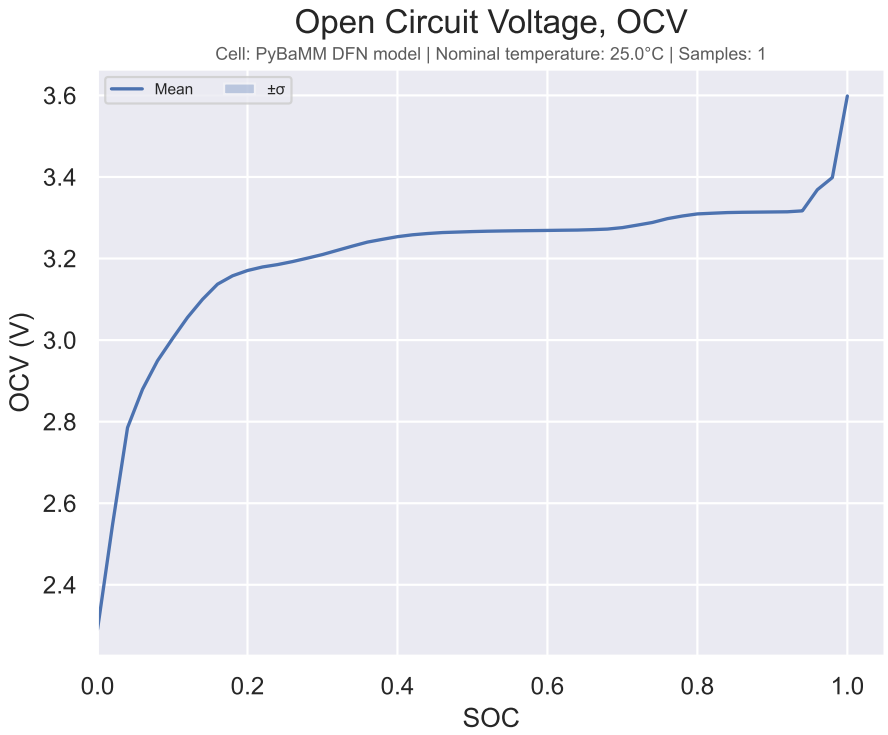


Figure 3.1: Static OCV–SOC relationship extracted from the PyBaMM simulated OCV test for the Prada2013 LFP cell.

For evaluation purposes, a drive cycle profile is simulated on the PyBaMM battery. The drive cycle is derived from a real–world drive, capturing realistic usage of the cell.

3.1.3 Equivalent circuit model and system identification

The data obtained from the PyBaMM OCV and HPPC test simulations is used as input to a characterization framework, developed by the Volvo BMS control team, to derive the parameters for the ECM (the overall procedure is illustrated in Figure 3.2). The current–based state–space model (2.5) is used for the EKF, and the voltage–

based model (3.3) for the SMOs. The Luenberger observer was implemented for both formulations. The framework utilizes a numerical optimization approach to extract the complete set of parameters required by the model. Specifically, it identifies the OCV (v_{oc}) and its local derivative with respect to SOC ($\frac{\partial v_{oc}}{\partial z}$), the instantaneous ohmic resistance (R_0), the polarization resistances and their corresponding time constants (R_1, τ_1, R_2, τ_2), the hysteresis magnitude and rate constants (M, γ), as well as the nominal cell capacity (Q) and Coulombic efficiency (η).

While parameters such as Q, η, and γ are treated as static constants, the majority of the extracted parameters are mapped as nonlinear functions of the cell's operating conditions. For instance, v_{oc}, R_0, and M are highly dependent on the SOC, whereas the transient RC parameters (R_1, τ_1, R_2, τ_2) exhibit dependencies on both the SOC and the applied current (i).

This comprehensive extraction enables a rigorous evaluation of the ECM in comparison to the high–fidelity PyBaMM battery cell model. The comparison is based on simulated dynamic current profiles previously mentioned. The current profiles are fed to both the ECM and the PyBaMM simulation, returning the respective terminal voltage responses which are compared to validate the ECM parameterization. The Root Mean Square Error (RMSE) is used as the primary metric to evaluate the performance of the ECM, and the threshold for acceptance is strictly set to 10 mV.

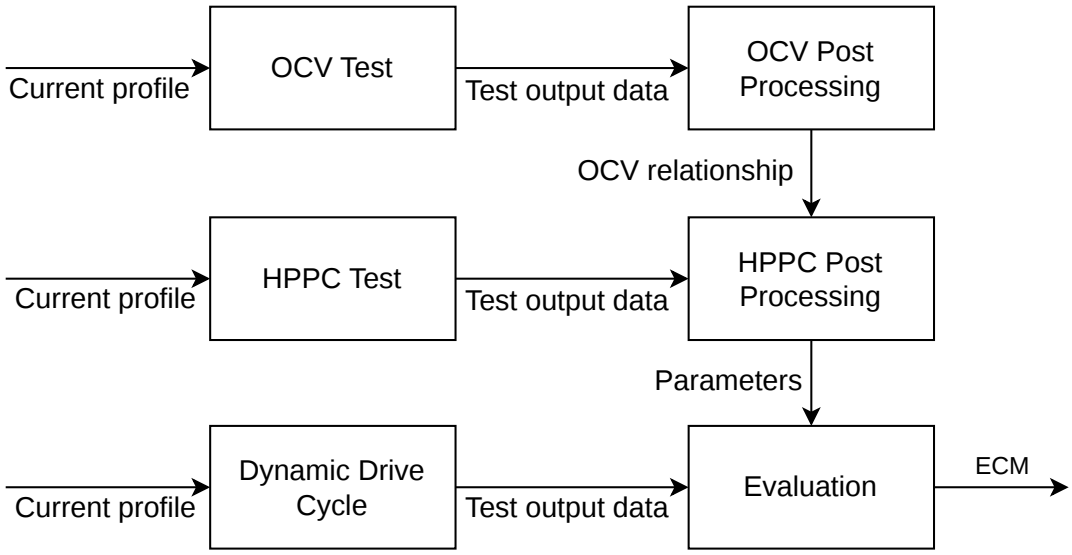


Figure 3.2: Block diagram of the ECM parameterization procedure. Simulated current profiles are applied to the PyBaMM plant to generate characterization data, which are utilized in a parameterization framework to obtain the ECM.

3.2 State estimation observers

To ensure a consistent and fair comparison, all evaluated observers share a standardized implementation framework. The battery dynamics are nonlinear, *i.e.,* the internal system matrices of each observer must be dynamically updated at every

discrete time step. At each sample step k, a set of ECM parameters (obtained from Section 3.1.2) is extracted via look–up tables based on the SOC estimate from the previous time step ($\hat{z}_{k-1}$) and the measured input current. The parameter set updates the standard ECM parameters (R_0, R_1, τ_1, R_2, and τ_2), hysteresis parameters (M and γ), the OCV (v_{oc}), and its local derivative ($\frac{\partial v_{oc}}{\partial z}$).

Furthermore, while the EKF is formulated in discrete time, the deterministic observers (Luenberger and all sliding–mode variants) are mathematically derived in continuous time. Therefore, During runtime, the explicit Euler method is applied over the sampling interval $\Delta t = 0.1$ s to propagate the states to the next iteration as $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \dot{\hat{\mathbf{x}}}_{k-1}\Delta t$.

3.2.1 Extended Kalman filter

The EKF is an industry–standard algorithm for real–time SOC estimation in Battery Management Systems [37], and it will serve as the primary conventional benchmark for evaluating the sliding–mode algorithms. For the EKF, the current–based state–space formulation defined in Equations (2.5) and (2.6) is utilized.

Using the shared dynamic parameter extraction framework, the discrete–time system matrices are evaluated analytically at each time step. Conveniently, the dynamically evaluated state transition matrix $\mathbf{A}_k$ and measurement matrix $\mathbf{C}_k$ act directly as the discrete Jacobians $\mathbf{F}_k$ and $\mathbf{H}_k$ required by the standard EKF Equations (2.11).

The execution of the EKF follows a recursive two–step process at each sampling instant:

1. **Prediction Step:** The state–space model propagates the *a priori* state estimate forward, and the error covariance matrix is projected using the state transition matrix and the process noise covariance matrix $\mathbf{Q}$. Based on these predicted states, the estimated terminal voltage is calculated according to Equation (2.6).

2. **Update Step:** The Kalman gain $\mathbf{K}_k$ is computed using the predicted error covariance and the measurement noise covariance scalar R. The innovation $(y - \hat{y})$ is multiplied by the Kalman gain to correct the estimates, yielding the *a posteriori* state vector and the updated covariance matrix $\mathbf{P}_{k|k}$.

Note that the observer needs to be initialized based on SOC. Moreover, $\mathbf{P_0}$ must be initialized as well. The parameters that are used as the tunable gains are $\mathbf{Q}$ and R. Note also, that when applied to a linear model, the EKF works as a regular KF.

3.2.2 Luenberger observer

The Luenberger observer is also a conventional observer that will be used as a deterministic benchmark. Unlike the EKF, which computes a new optimal gain each time step, the Luenberger observer utilizes a constant, time–invariant feedback gain matrix $\mathbf{L}$.

The observer is implemented using the voltage–based state space formulation (see Equation 3.3). Utilizing the dynamically updated system matrices, the estimated continuous state derivatives of the observer are governed by the standard Luenberger dynamics, driven by the residual (e_y) between the measured plant voltage (y) and the predicted terminal voltage ($\hat{y}$):

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \phi + \mathbf{L}(e_y).$$

The continuous dynamics are then propagated to the next iteration using explicit Euler integration.

3.2.3 SMO design

The SMOs are implemented based on the continuous time voltage state space–model given in Equation (3.3). Note that the Lyapunov analysis is developed in continuous time, thus, the mathematical formulation is in continuous time. For implementation purposes, the sliding–mode algorithms also use explicit Euler integration.

3.2.4 First–Order SMO

The FOSMO is the simplest of the SMOs and consists of the algorithm (2.12). The stability analysis guarantees the convergence of the state estimation error with or without the Luenberger term. Therefore, the algorithm takes a slightly different shape, where the modeled nonlinear dynamics term ϕ is included, identically to the Luenberger observer, *i.e.*,

$$\begin{aligned} \frac{d\hat{\mathbf{x}}}{dt} &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \phi + \rho\,\mathrm{sgn}(e_y), \\ \hat{y} &= \mathbf{C}\hat{\mathbf{x}}. \end{aligned}$$

The tuning parameter ρ is constant between iterations and is tuned based on the upper bound of the external disturbances.

3.2.5 Second–Order SMO

Implementation–wise the SOSMO is no different than the FOSMO, except that the algorithm is different as seen in Equation (2.13). The state equation still incorporates the ϕ term to account for the modeled nonlinear dynamics. The tuning parameters (ρ_1 and ρ_2) also remain constant between iterations, where ρ_2 should be tuned as the upper bound of the external disturbance, and ρ_1 to modify the force of the correction action.

3.2.6 Twisting versions of SMO

The twisting versions require an exact differentiator for obtaining the derivative of the estimated terminal voltage ($\dot{\hat{v}}_t$). Therefore, the following exact robust Levant differentiator is proposed [34, pp. 34-35]:

$$\begin{aligned} \dot{\mathbf{z}}_0 &= -\lambda_0 \xi^{\frac{1}{2}} \left|(\mathbf{z}_0 - \mathbf{x}_1)\right|^{\frac{1}{2}} \mathrm{sgn}(\mathbf{z}_0 - \mathbf{x_1}) + \mathbf{z}_1, \\ \dot{\mathbf{z}}_1 &= -\lambda_1 \xi\, \mathrm{sgn}(\mathbf{z}_1 - \dot{\mathbf{z}}_0). \end{aligned}$$

The derivative of the estimated terminal voltage is then used to compute the error derivative $\dot{e}_y = \dot{v}_t - \dot{\hat{v}}_t$ required in the algorithms. The exact differentiator contains tuning parameters ξ, λ_0, λ_1, and $\mathbf{z}$, not to be confused with SOC, is a vector containing the estimate of the terminal voltage ($\hat{v}_t$) and the derivative of the estimate ($\dot{\hat{v}}_t$). As suggested by Levant [34, p. 35], the parameter values are chosen as $\lambda_0 = 1.5$ and $\lambda_1 = 1.1$. The algorithms of the twisting SMOs can be seen in Equations (2.14), (2.15), and (2.16) respectively, and just like the lower–order observers, their state update equations are extended with the nonlinear dynamics vector ϕ.

3.3 Observer tuning

To obtain a fair comparison of the observers, an automated hyperparameter optimization framework is proposed. Each observer has multiple hyperparameters that have a high impact on the outcome of the state estimation. To avoid human bias, an optimization framework is used to systematically identify the optimal tuning parameters for each individual observer.

The optimization is formulated as a minimization problem, where the objective cost function is defined as the RMSE of the SOC estimation over a dynamic drive cycle.

The following requirements are given for the optimization:

- Nominal parameters are used for the observers.
- White Gaussian noise is injected into the voltage and current signals.
- The observers are initialized with an initial SOC offset of 3%.

If tuned with perfect initial conditions, the optimization tends to minimize error by selecting near–zero feedback gains, effectively reducing the observers to open–loop integrators that ignore sensor feedback. By introducing an initial state mismatch, the optimizer is forced to find a robust balance between fast transient convergence (to correct the offset) and steady–state noise attenuation.

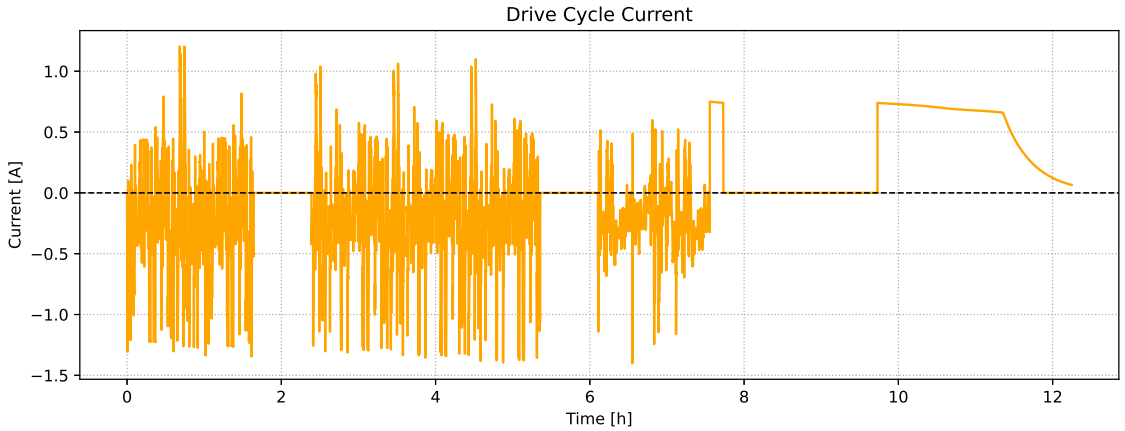
The tuning process was executed utilizing Optuna [38], a hyperparameter optimization framework. The framework uses a Bayesian optimization method called Tree–structured Parzen Estimator (TPE). To manage the high computational load of simulating hundreds of trials for each observer, the Optuna optimization was executed in parallel across multiple CPU cores.

3.4 Validation framework

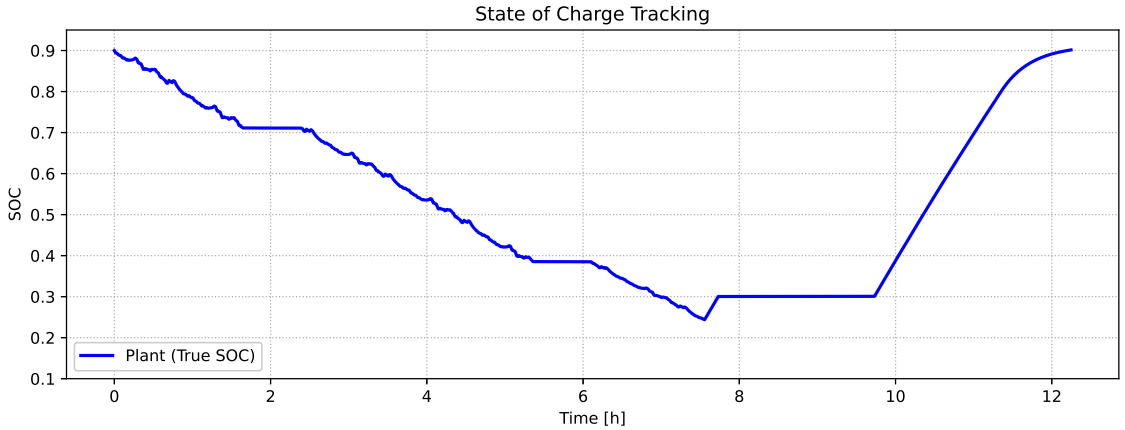
To evaluate the performance of the state estimation algorithms, a simulation framework was developed in Python. As previously outlined, all observers were validated against a realistic drive cycle simulated by the high–fidelity PyBaMM DFN model. As illustrated in Figure 3.3, this profile spans just over 12 hours and subjects the cell to highly transient loads, including aggressive discharge phases corresponding

to vehicle acceleration, and high–current charge pulses representing regenerative braking.

During the simulation, the DFN model processes the drive cycle current to compute both the cell's true terminal voltage and SOC is calculated via exact CC. To emulate a physical environment subject to real–world sensor inaccuracies, white Gaussian noise is injected into both the current and voltage measurements. The baseline configurations and noise characteristics utilized within this simulation framework are summarized in Table 3.2. A schematic diagram of the validation framework is shown in Figure 3.4.



(a) Dynamic input current profile.



(b) True State of Charge (SOC) of the simulated plant.

Figure 3.3: The validation drive cycle utilized for observer evaluation. The noise–free current profile (a) and the cell's true SOC (b). The SOC signal is computed using CC without the artificial sensor noise used during observer testing.

Table 3.2: Simulation and configuration parameters used in the validation framework.

Parameter Description	Value	Unit
Sampling time (T_s)	0.1	s
Initial plant true SOC (z_0)	90	%
Initial observer estimated SOC ($\hat{z}_0$)	87	%
Current sensor noise (σ_i)	0.01	A
Voltage sensor noise (σ_v)	0.002	V

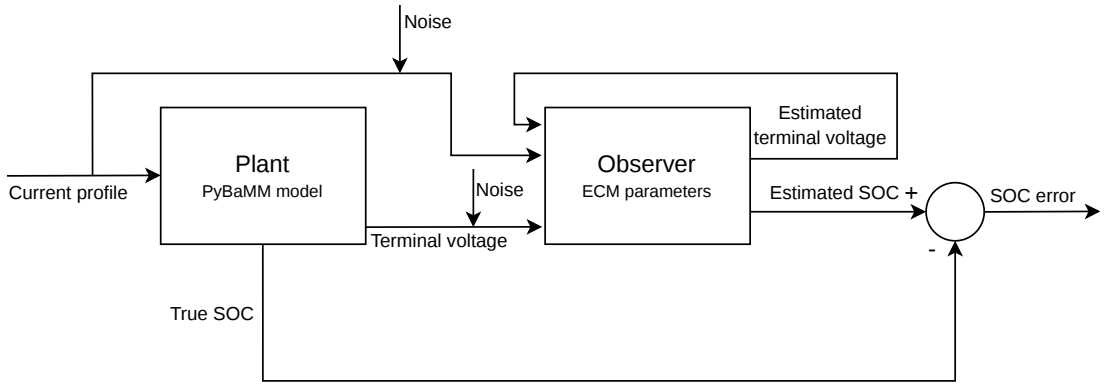


Figure 3.4: Validation framework scheme for the observers.

A core objective of this validation was to assess how well the different observers handle parameter variations caused by, for instance, cell aging, temperature shifts, and parameter uncertainties. In other words, how the observers perform from a robustness perspective. This was simulated by scaling the internal parameters in the ECM observer formulation, according to the configurations seen in Table 3.3. By scaling the internal parameters, the parameter uncertainty appears in the system. Three different uncertainty cases were simulated. Firstly, uncertainties were applied to the Ohmic resistance and capacity (R_0, Q). Secondly, uncertainties were applied to the fast dynamics (τ_1). Finally, uncertainties were applied to all parameters, *i.e.*, Ohmic resistance, capacity, fast and slow dynamics ($R_0, Q, R_1, \tau_1, R_2, \tau_2$).

In Case 1, the scaling on the Ohmic resistance and the capacity ranged from 10% to 30% for the former and 5% to 20% for the latter, as can be seen for configurations 1–4 in Table 3.3. It represents an aged battery cell where the capacity has deteriorated, and the resistance has increased. Thus, it represents a scenario where the plant emulates a battery cell at the beginning of life (BOL) while the observer formulation emulates an aged battery cell. Indeed, these are worst–case scenarios and purposely aim at assessing robustness.

Case 5 represents a realistic behavior in the system, *i.e.,* a small uncertainty in the ohmic resistance and capacity measurement.

Case 6 represents a case where there is a small uncertainty in the fast dynamics, whereas Cases 7 and 8 represent a small and large uncertainty in all the parameters, respectively.

Table 3.3: Simulation configurations for different uncertainties. Note that the numbers are scaling factors, not the actual value for each parameter.

Case	R_0	Q	R_1	τ_1	R_2	τ_2
1	1.1	0.8	1.0	1.0	1.0	1.0
2	1.1	0.95	1.0	1.0	1.0	1.0
3	1.3	0.8	1.0	1.0	1.0	1.0
4	1.04	0.975	1.0	1.0	1.0	1.0
5	1.3	0.95	1.0	1.0	1.0	1.0
6	1.0	1.0	1.0	1.2	1.0	1.0
7	1.05	0.95	0.95	1.05	1.05	0.95
8	1.2	0.8	0.8	1.2	1.2	0.8

3.4.1 Evaluation criteria

To provide an objective analysis among the developed observers, statistical error metrics are calculated over the drive cycle. The true SOC provided by the PyBaMM plant at a discrete time step k is denoted as z_k, and the observer's estimated SOC is denoted as $\hat{z}_k$. The SOC estimation error at any given time step is thus defined as $e_{z,k} = \hat{z}_k - z_k$.

is used as the primary evaluation metric, where N represents the total number of sampled data points in the simulation. Because the RMSE squares the error terms before averaging, it penalizes larger deviations, making it a good metric for the overall stability and accuracy of the state estimation throughout the drive cycle.

Furthermore, the Maximum Absolute Error (MAE) of the SOC error is recorded to identify the single worst–case estimation deviation. It is calculated as:

$$\text{MAE}_{SOC} = \max_{k \in \{1,\ldots,N\}} |e_{z,k}|.$$

This can be used to ensure that the estimation error never deviates more than a chosen threshold.

Besides evaluation based on statistical error metrics, computational complexity was also taken into consideration. However, without deploying the algorithms on actual

hardware, there was limited room for testing, *e.g.,* the algorithms do not run in real–time in the simulation, which causes the computer to run the whole sequence as fast as possible rather than processing data when available. Moreover, the resource usage is affected by other tasks running on the computer.

Consequently, the average CPU time for the math operations was used as an evaluation metric. Runtime measurements were obtained using Python's `time.perf_counter()` around the loop running the math operations in the algorithms. The validation drive cycle was executed over 10 independent iterations for each algorithm to ensure statistical reliability. From this dataset, the mean execution time per discrete step was calculated to evaluate the average computational load. While the maximum execution time remains a vital metric for real–time deadline compliance in embedded environments, isolating the true worst–case execution spikes requires an isolated real–time operating system environment. Due to the limited development resources and hardware constraints, a dedicated worst–case test was not carried out.

4

Results and discussion

This chapter aims to present the results obtained when characterizing the ECM, optimizing the observers, and running the simulation framework as presented in Chapter 3.

4.1 Characterization and parametrization

Before evaluating the state estimation observers, the parametrized ECM must be validated to ensure it accurately replicates the PyBaMM plant. The parameters were extracted using the characterization tests outlined in the previous chapter. The resulting parameters are provided in Appendix A.

To verify the accuracy of the parametrized ECM[1], the model was subjected to a validation drive cycle profile. Figure 4.1 displays the predicted terminal voltage of the ECM overlaid on the true terminal voltage generated by the PyBaMM model. Visually, the ECM demonstrates strong tracking during the different segments (Dynamic, Rest, Charge). The overall RMSE between the ECM and the PyBaMM terminal voltage over the entire drive cycle is 3.0 mV, and this, together with the remaining error metrics, can be seen in Table 4.1. The error is below the 10 mV threshold. Thus, the ECM is considered an accurate representation of the PyBaMM model and can be used in the observer evaluations.

4.2 Hyperparameter optimization

As mentioned in the Methods chapter, hyperparameter optimization was performed using the Optuna framework with the TPE algorithm. To ensure convergence to a global minimum, the optimization was allowed to run for 500 trials per observer, and a clear convergence was observed for all observers. The final optimized hyperparameter sets for each observer are presented in Table 4.2. The given parameters are used for all subsequent simulations to ensure an objective comparison.

[1]The ECM formulation used in this comparison is the one based on currents as states. Note that the voltage formulation is based on the current formulation, thus, the accuracy is similar. However, to prove the robustness of the proposed algorithms, the voltage formulation does not include model hysteresis.

Table 4.1: Error metrics calculated based on comparison of model voltage to the drive cycle voltage response.

Classification	RMSE (mV)
Overall	3.0
Charge	4.4
Discharge	4.2
Dynamic	2.9
Rest	1.3

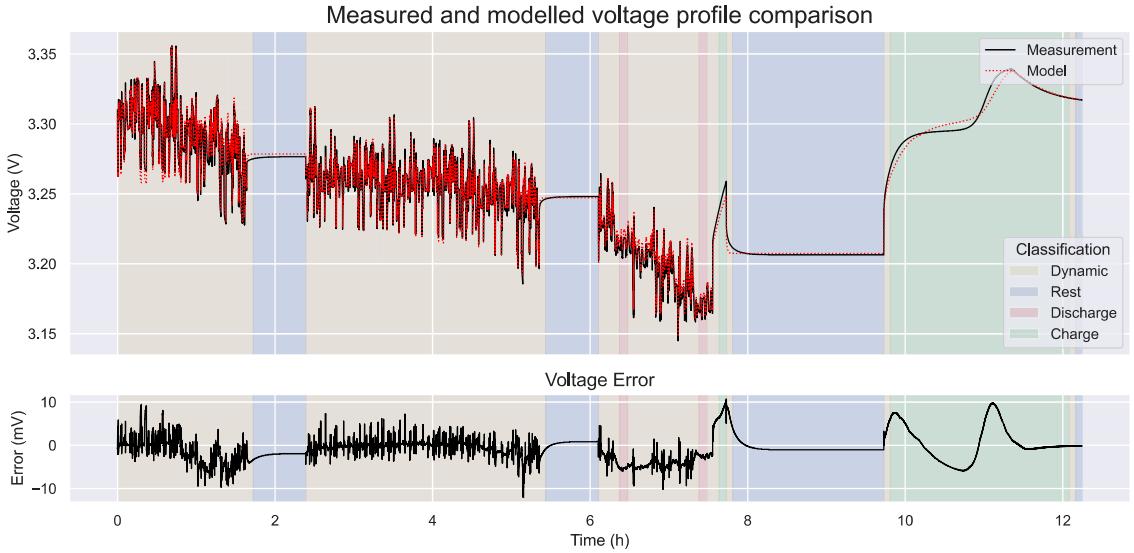


Figure 4.1: Voltage profile from PyBaMM (measurement) and as generated by the ECM, with corresponding errors. This voltage profile is generated by the ECM based on the current state–space formulation.

An interesting observation from the optimized hyperparameters in Table 4.2 is the gain distribution across the state vector. For the Luenberger observer and the lower–order SMOs, the optimization algorithm consistently assigned a substantially larger feedback gain to the terminal voltage state compared to the SOC state. Since voltage and SOC are linked through the system matrix $\mathbf{A}$, correcting the voltage naturally pulls the SOC to the right value over time. If the optimizer had applied a high gain directly to the SOC state, it would have just pumped raw sensor noise straight into the estimate, ruining the RMSE.

Table 4.2: Optimized hyperparameter sets for the evaluated state estimation algorithms.

Observer	Optimized Tuning Parameters
Extended Kalman Filter (EKF)	$\mathbf{Q} = \text{diag}(2.99 \times 10^{-12},\ 5.08 \times 10^{-3},\ 1.04 \times 10^{-5},\ 4.76 \times 10^{-12})$ $R = 1.32 \times 10^{-6}$
Luenberger Observer	$\mathbf{L} = \begin{bmatrix} 2.99 & 1.23 \times 10^{-13} & 4.26 \times 10^{-8} & 3.13 \times 10^{-11} \end{bmatrix}^\top$
First–Order SMO (FOSMO)	$\boldsymbol{\rho} = \begin{bmatrix} 1.91 \times 10^{-1} & 2.86 \times 10^{-3} & 3.56 \times 10^{-5} & 1.14 \times 10^{-14} \end{bmatrix}^\top$
Second–Order SMO SOSMO	$\boldsymbol{\rho}_1 = \begin{bmatrix} 3.58 & 2.87 \times 10^{-14} & 1.28 \times 10^{-11} & 3.71 \times 10^{-15} \end{bmatrix}^\top$ $\boldsymbol{\rho}_2 = \begin{bmatrix} 2.94 \times 10^{-3} & 1.28 \times 10^{-5} & 3.30 \times 10^{-5} & 7.80 \times 10^{-5} \end{bmatrix}^\top$
Singular Terminal SMO (CSTSMO)	$k_1 = 237.0, \quad k_2 = 0.525, \quad k_3 = 2.25 \times 10^{-4}$ $\xi = 3.16$
Continuous Non–Singular Terminal SMO (CNTSMO)	$k_1 = 22.6, \quad k_2 = 0.774, \quad k_3 = 1.45 \times 10^{-5}$ $\xi = 0.398$
Continuous Twisting SMO (CTSMO)	$k_1 = 2.89, \quad k_2 = 9.90 \times 10^{-6}, \quad k_3 = 3.81 \times 10^{-2}, \quad k_4 = 2.14 \times 10^{-5}$ $\xi = 1.44 \times 10^{-7}$

4.3 Baseline state estimation performance

All the observers are populated with nominal parameters extracted from the PyBaMM characterization tests. The observers are initialized with a 3% offset ($\hat{z}_0 = 87\%$, whereas the true plant SOC is $z_{true} = 90\%$). It is realistic to have an initial SOC error due to the latest known SOC point being obtained when the batteries were under load. Therefore, an initial SOC error will occur as the voltage increases slightly when the batteries rest. Three percent is a rather large error and can be considered a worst–case scenario.

The absolute SOC tracking over the duration of the dynamic drive cycle is presented in Figure 4.2. As depicted, all evaluated observers successfully compensate for the initial offset, with their estimated trajectories converging toward the true plant SOC. To better visualize the estimation accuracy and the transient behavior, the SOC estimation error ($e_z = \hat{z} - z_{true}$) is plotted in Figure 4.3.

When looking at the initial reaching phase in Figure 4.3, it is clear that the EKF presents a higher overshoot compared to the other observers. However, the SMOs have a higher error overshoot at 55% SOC which is likely due to the SMOs and Luenberger observers not having knowledge of the hysteresis effect. Around 50–60% SOC, the hysteresis effect is stronger, which leads to a higher error. This effect is not affecting the EKF, which perfectly maps the hysteresis effect.

Since there is no artificial parameter degradation, the observers achieved a SOC RMSE below 1 % as shown in Table 4.3. Notably, the FOSMO yielded the lowest overall error (0.8063%), marginally outperforming the EKF benchmark (0.8304%).

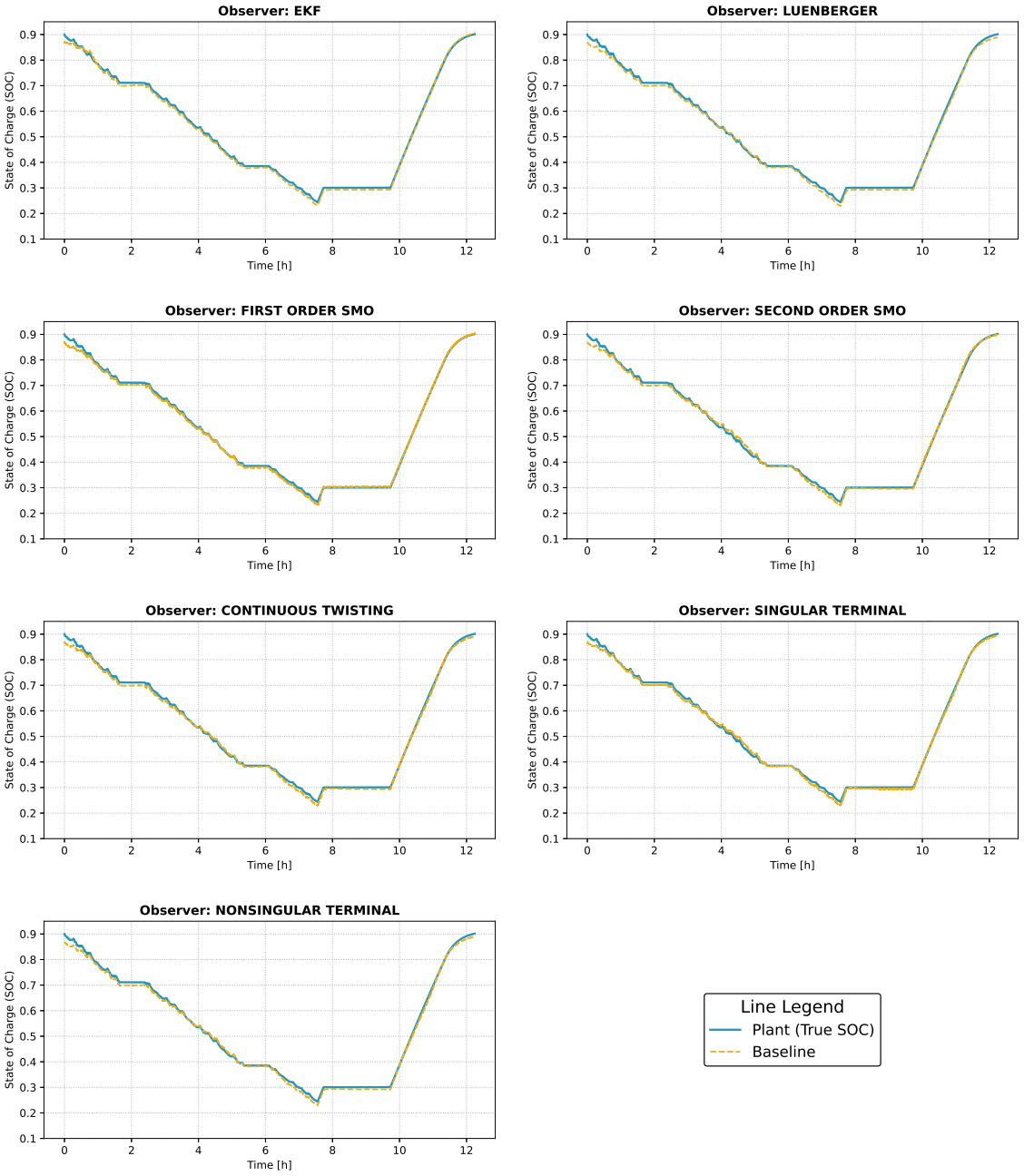


Figure 4.2: SOC estimation without parameter uncertainties compared to the plant. The observers are tuned for optimal performance with respect to RMSE in the baseline setting.

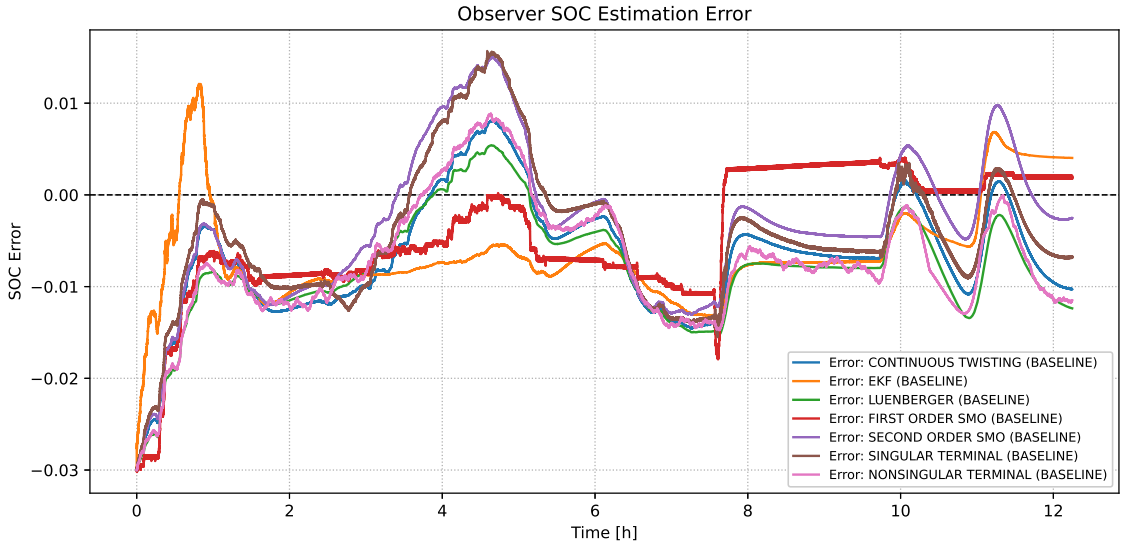


Figure 4.3: SOC estimation error comparison between all observers without parameter uncertainties.

Table 4.3: Baseline State of Charge (SOC) estimation performance. The observers are initialized with a 3% offset ($\hat{z}_0 = 0.87, z_{true} = 0.90$) and evaluated under nominal parameter conditions with sensor noise. The observers are listed in ascending order with respect to RMSE.

Observer	**SOC RMSE**
First–Order SMO (FOSMO)	0.8063
Extended Kalman Filter (EKF)	0.8304
Second–Order SMO (SOSMO)	0.8756
Continuous Singular Terminal SMO (CSTSMO)	0.8840
Continuous Twisting SMO (CTSMO)	0.9218
Continuous Non–Singular Terminal SMO (CNTSMO)	0.9842
Luenberger Observer	0.9951

4.4 Robustness to parameter uncertainties

Looking back at Section 1.5, one of the main issues to investigate was the robustness of the proposed SMOs against conventional state estimation techniques, under parameter uncertainties and external disturbances. With this in mind, the results will be divided into two sections. Firstly, observer performance when uncertainties are applied to the Ohmic resistance and the capacity. Secondly, when uncertainties are applied to the fast dynamics, followed by the case where the uncertainties are applied to all the parameters. In all simulations, the observers were tuned for the ideal case where the parameters of the ECM are known. The SOC tracking plots for all observers, for the respective uncertainty case, can be seen in Appendix B.

4.4.1 Uncertainties in Ohmic resistance and capacity

The numerical results are presented in Figure 4.4. Note that the conventional filtering techniques have a lower performance in comparison to the SMOs. Clearly, the proposed SMOs are robust against the parameter uncertainties. Similar results can be observed when looking at the MAE (See Figure 4.5).

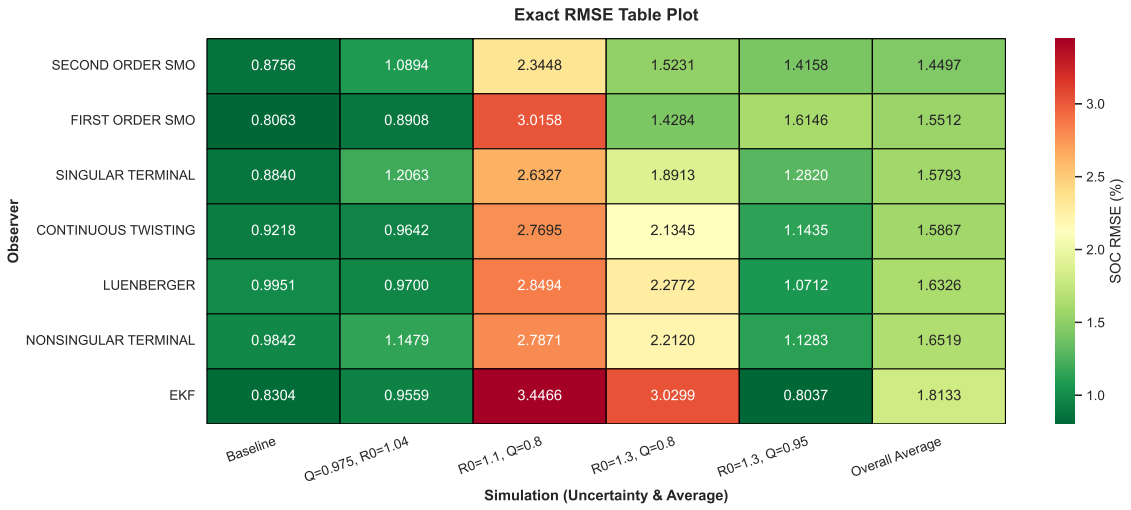


Figure 4.4: Heatmap visualizing RMSE performance for each observer in different simulation scenarios. Each column on the x–axis represent a scenario where different ECM parameter uncertainties are applied. Specifically, uncertainties in Ohmic resistance and capacity. The simulations were initiated with a 3% initial SOC error.

In the MAE context, the Luenberger observer performs slightly better than the FOSMO and the CNTSMO. It is worth mentioning that the SOC error initiated with a 3% error, thus the MAE is at its lowest 3%. In reality, the simulation scenarios can be considered unrealistic due to the large uncertainties. A battery with 80% capacity is at the end of life, which makes it extreme to consider it an uncertainty. Still, not all applications implement online parameter updates which can make uncertainty about capacity and ohmic resistance a reasonable real–world scenario in a worst–case context. Then, the ability for an observer to deal with the

worst–case scenario can be considered an important performance metric. Therefore, SMOs stand out as a robust alternative compared to EKF.

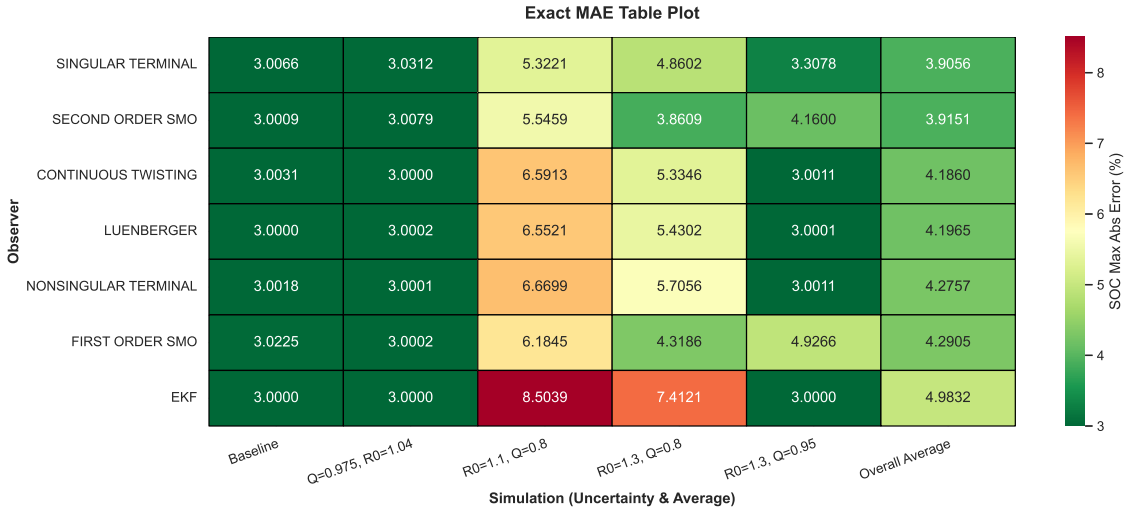


Figure 4.5: Heatmap visualizing MAE performance for each observer in different simulation scenarios. Each column on the x–axis represent a scenario where different ECM parameter uncertainties are applied. More specifically, uncertainties in Ohmic resistance and capacity. Moreover, the simulations were initiated with a 3% initial SOC error.

The Luenberger performs surprisingly well and is on par with some of the SMOs even in the worst–case scenarios. Still, it can be seen that the smaller the uncertainties the smaller the difference in RMSE between all observers.

Moreover, the EKF Kalman gain is recomputed in every timestep while the SMO gains are static. Therefore, when the model uncertainties are in the lower regions, the EKF may have an advantage. However, the covariance matrices are still static meaning the initial tuning is crucial. If it is tuned to trust the model, this will naturally result in worse performance when the model is incorrect due to large changes in the internal parameters of the battery cell. This is where the robustness properties of the SMOs can be advantageous, due to the Lyapunov stability mentioned in Section 2.6.4.

Furthermore, it can be seen that the observer performance deteriorates more when R_0 is perturbed less relative to Q. This is a reasonable result, due to the physical meaning of perturbing the resistance and the capacity. It also aligns with the proposed estimation strategies. As can be seen, uncertainty about capacity has the largest influence on the performance. A smaller battery drains and charges faster than a large battery when subjected to the same input current stimuli. When the model has an incorrect capacity, it is predicting that the SOC is higher than in reality during charge, and lower than in reality during discharge. When it has an incorrect resistance, it is predicting that the terminal voltage is higher than in reality during charge, and lower than in reality during discharge. As the terminal voltage is being measured, the output estimation error is going to indicate that the terminal voltage

is too high or too low depending on charge and discharge. Therefore, the observer will correct and lower or raise the terminal voltage which in turn propagates to the SOC. A larger perturbation in the resistance will thus result in a larger correction from the observers, which turns out to be beneficial when uncertainties are applied on the capacity. Thus, the results are better in the case where R_0 is perturbed by 30% than when it is perturbed by 10%. For the FOSMO however, it can be seen that this effect is realized differently. This is because the FOSMO only uses the sign of the error, and does not consider the magnitude. The effect that it gets from R_0 perturbation is that the sign will be more consistent, *i.e*, affecting the estimation in the same direction for a longer duration.

4.4.2 Uncertainties in dynamics

The RMSE results are shown in Figure 4.6. It is important to highlight that, in the case where all the uncertainties are applied, the RMSE is lower than in some of the cases of the previous section. Still, this could be due to the measurement random noise or a parameter compensation. However, the parameter analysis is not part of the current scope. In conclusion, SMOs are the robust alternative. EKF obtains the best performance when uncertainties are applied only to the fast dynamics. Note that the fast dynamics are only affecting the system for a very short duration in each pulse–like stimuli of the drive cycle. Therefore, the average effect over the entire drive cycle is negligible in comparison to the cases where the capacity and Ohmic resistance are altered.

The results for the MAE are shown in Figure 4.7. It is worth highlighting that most of the SMOs are below the 3% expected boundary. In the presented case, the EKF has the worst performance.

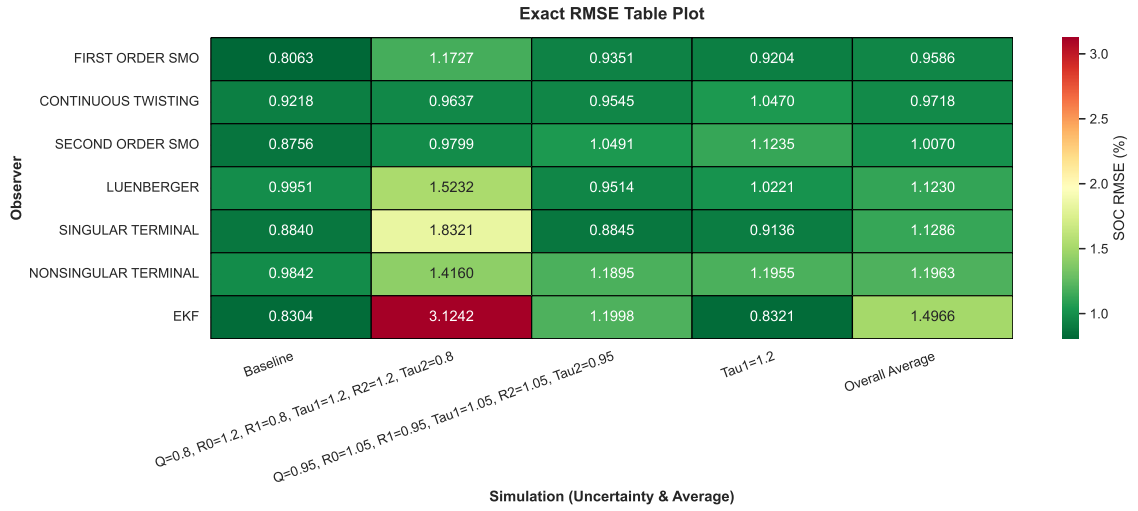


Figure 4.6: Heatmap visualizing RMSE performance for each observer in different simulation scenarios. Each column on the x–axis represent a scenario where different ECM parameter uncertainties are applied. Specifically, uncertainties in ohmic resistance, capacity, fast and/or slow dynamics.

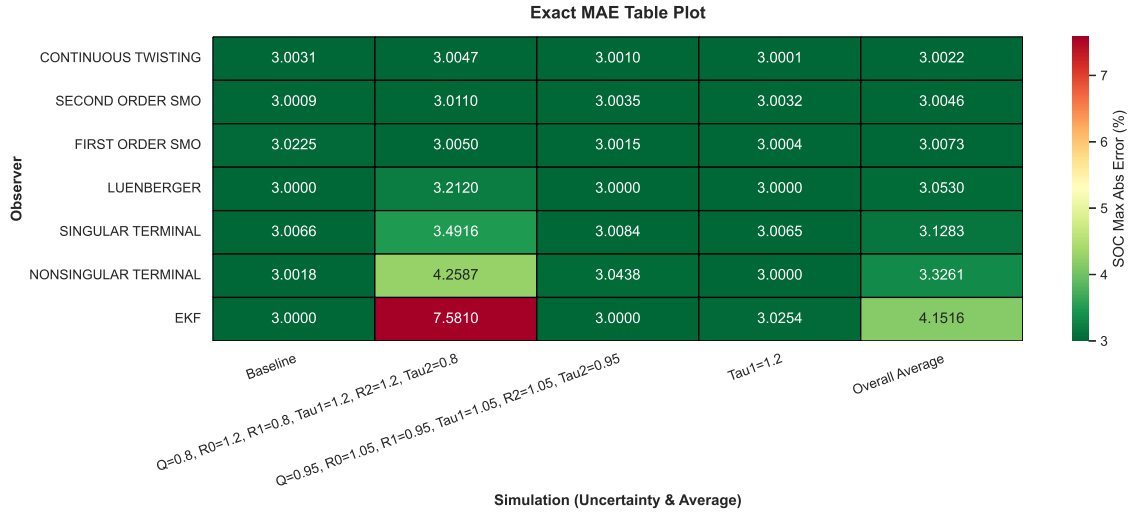


Figure 4.7: Heatmap visualizing MAE performance for each observer in different simulation scenarios. Each column on the x–axis represent a scenario where different ECM parameter uncertainties are applied. Specifically, uncertainties in ohmic resistance, capacity, fast and/or slow dynamics.

4.5 Chattering analysis

As mentioned in Section 2.6.1 FOSMOs suffer from chattering. As this effect is amplified by the tuning parameter ρ, the tuning has a big influence on whether chattering is visible in the estimates or not. As can be seen in Figure 4.8 the optimal tuning resulted in a gain where chattering occurs in the SOC estimates. Still, the magnitude of the chattering is negligible and thus does not deteriorate the performance of the observer.

When comparing to the results obtained in a recent study [18], it is evident that the proposed tuning strategy in this work significantly reduces chattering for the FOSMO. It can be seen that the CSTSMO exhibits a similar tendency. However, this is not chattering but oscillations that occur due to singularities in the sliding surface (as discussed in Section 2.6.3.1).

Nevertheless, based on the tuning, the chattering could decrease, but the external disturbances will impact the state estimation performance and vice versa. However, the chattering is only problematic for actuators, which is not the case in state estimation algorithms.

4.6 Computational complexity

The computational complexity of the evaluated state estimation algorithms is summarized in Table 4.4. It is important to note that the benchmarks are executed on a standard PC operating system rather than a dedicated BMS microcontroller, the absolute execution times in microseconds are not directly representative of embedded hardware performance. To provide a realistic comparison of algorithmic

efficiency, the execution times have been normalized against the fastest algorithm (the Luenberger observer) to establish a relative execution multiplier.

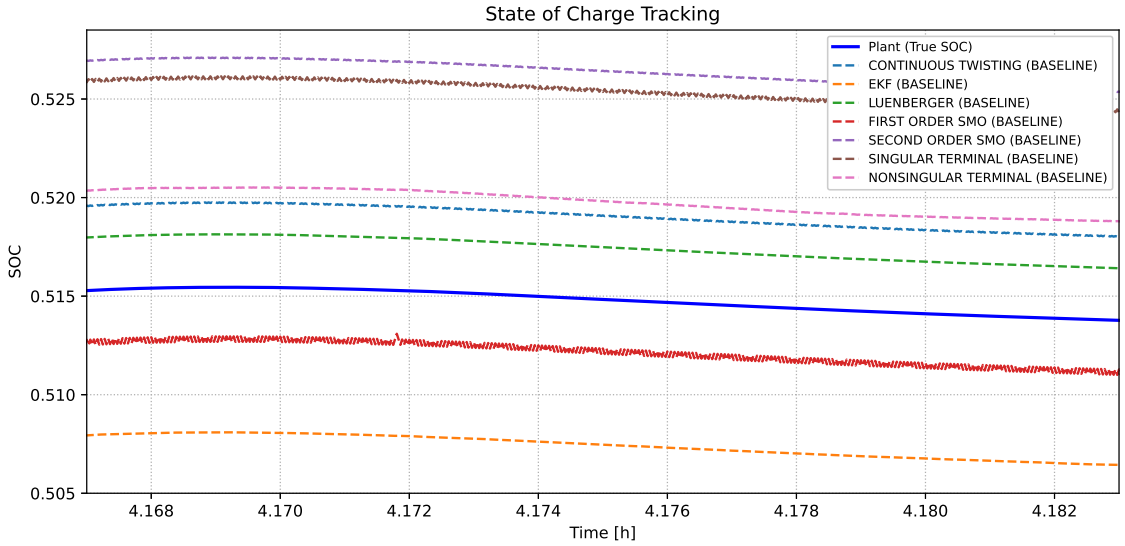


Figure 4.8: Comparison of chattering in SOC estimates. The FOSMO exhibits chattering, and the CSTSMO oscillates but does not present chattering. The remaining observers provide smooth estimates.

As evident in the results, the EKF imposes the highest average computational load, executing 3.41× slower than the baseline.

Table 4.4: Computational complexity benchmark across 10 sequential iterations. Values represent the isolated mathematical execution time per discrete step. The normalized multiplier provides a relative comparison between the observers (baseline = Luenberger).

Observer	**Mean Time (μs)**	**Normalized Time**
Luenberger	8.90	1.00×
FOSMO	9.61	1.08×
SOSMO	12.74	1.43×
CSTSMO	20.90	2.35×
CNTSMO	24.76	2.78×
CTSMO	25.19	2.83×
EKF	30.31	3.41×

EKF algorithm requires computationally demanding matrix–matrix multiplications and inversions at every discrete time step, primarily to recursively update the error covariance matrix ($\mathbf{P}_k$). In contrast, the baseline Luenberger observer and the FOSMO are efficient. By relying on a constant feedback gain matrix and, in the case of FOSMO, a computationally lightweight signum function, the execution is effectively reduced to standard matrix–vector multiplications. Consequently, the FOSMO requires approximately 68% less processing time than the EKF.

As the math of the sliding mode variants increases, the computational cost rises accordingly. While the SOSMO remains efficient, the inclusion of exact differentiators in the CSTSMO, CNTSMO, and CTSMO algorithms introduces substantial overhead. This increase is primarily driven by the fractional power calculations (*e.g.,* $x^{1/2}$, $x^{2/3}$), which are inherently more demanding for basic processors to evaluate than standard linear arithmetic.

From an implementation perspective, the reduction in computational complexity offered by the lower–order SMOs provides advantages for real–world BMS applications. Extra CPU–time can be strategically reallocated to execute other resource–intensive algorithms, such as other BMS state estimations, or utilized to increase the observer's sampling frequency to improve the SOC estimation accuracy.

5

Conclusion and future work

5.1 Conclusion

A comparative study between conventional observers and sliding–mode observers are given. A high–fidelity battery simulator is used as the plant for simulation purposes. A parameterization scheme is built to obtain the parameters of an ECM that is used as a mathematical formulation for the observers. An optimization framework is presented to obtain the best hyperparameters for the observers.

Following the stated objectives, the SM and conventional observers are designed, analyzed, and validated for a Li–ion battery. The most relevant metric is the robustness under parameter uncertainties, nonlinear unmodeled dynamics, external disturbances, and measurement noise. The SMOs are compared against the conventional state estimation methodologies, such as Luenberger and EKF.

The results indicate that, for the baseline, EKF has one of the best performances among the observers. However, in the presence of parameter uncertainties, nonlinear unmodeled dynamics, external disturbances, and measurement noise, the results suggest that some of the SMOs are better than nominal observers.

Besides the robustness, the computational complexity is evaluated. The results highlights that the Luenberger observer and SMOs have the lowest computational effort and complexity compared against the EKF. However, it is worth noting that twisting versions introduce computational effort due to their reliance on exact differentiators and fractional power calculations.

Ultimately, selecting an optimal state estimation technique requires a trade–off between accuracy, robustness, and computational efficiency. While the EKF provides similar state estimation to FOSMO under nominal conditions, its computational complexity and vulnerability to large parameter uncertainties limit the effectiveness when large uncertainties are prevalent. Contrary to this, SMOs are substantially more robust to large uncertainties and also more computationally efficient in the case of lower order SMOs. The Luenberger is not the most robust nor accurate observer, but the least expensive which makes it a candidate to consider when choosing state estimation strategy.

5.2 Suggested future work

Future improvements of this research involve evaluating the observers under more realistic uncertainty scenarios. This could be achieved by gathering empirical data from physical battery cells to accurately identify which parameter deviations are most prevalent during real–world operation and to quantify their typical magnitudes.

Furthermore, to make the observers more robust and ready for real–world applications, future work should incorporate temperature dependence. Currently, the parameters used in the ECM and the plant used during validation assume a constant operating temperature.

Another area for future improvement is to transition from static to adaptive tuning gains. In this study, the hyperparameters for all observers were optimized for no uncertainty and held constant during the entire simulation. To improve performance and robustness, future work should explore algorithms that dynamically change the values of the gains with respect to time and external disturbances.

Finally, this study focused on an LFP cell, which is challenging because of its flat OCV curve, future work should extend the analysis to different cell chemistries. Applying the developed simulation framework to cells with steeper, more forgiving OCV gradients would provide a more comprehensive understanding of the performance and general applicability of SMOs.

Bibliography

[1] E. Prada, D. Di Domenico, Y. Creff, J. Bernard, V. Sauvant-Moynot, and F. Huet, "A simplified electrochemical and thermal aging model of lifepo4-graphite li-ion batteries: Power and capacity fade simulations," *Journal of The Electrochemical Society*, vol. 160, p. A616, feb 2013.

[2] European Commission, Directorate-General for Communication, "UN Climate Change Conference: World agrees to transition away from fossil fuels and reduce global emissions by 43% by 2030," 2023. Accessed: 2026-01-21.

[3] International Energy Agency (IEA), "Global ev outlook 2025," 2025. Accessed: 2026-01-21; Licence: CC BY 4.0.

[4] Volvo Trucks, "Driving progress in electromobility." Accessed: 2026-01-21.

[5] Volvo, "Battery Electric Vehicles." Accessed: 2026-01-21.

[6] International Energy Agency (IEA), "The role of critical minerals in clean energy transitions," 2025. Licence: CC BY 4.0.

[7] F. Sciatti, P. Tamburrano, E. Distaso, R. Amirante, A. V. Radogna, A. Morciano, and G. Grassi, "A comprehensive overview of lithium-ion batteries for electric vehicles: Materials, performance, safety, recycling, and emerging technologies," *Journal of Energy Storage*, vol. 144, p. 119694, 2026.

[8] MathWorks, "What Is a Battery Management System (BMS)?." Accessed: 2026-01-22.

[9] A. Kurkin, A. Chivenkov, D. Aleshin, I. Trofimov, A. Shalukho, and D. Vilkov, "Battery management system for electric vehicles: Comprehensive review of circuitry configuration and algorithms," *World Electric Vehicle Journal*, vol. 16, no. 8, 2025.

[10] M. Hannan, M. Lipu, A. Hussain, and A. Mohamed, "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations," *Renewable and Sustainable Energy Reviews*, vol. 78, pp. 834–854, 2017.

[11] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, *Sliding Mode Control and Observation*. Control Engineering, Springer New York, 2013.

[12] Y. Huangfu, J. Xu, D. Zhao, Y. Liu, and F. Gao, "A novel battery state of charge estimation method based on a super-twisting sliding mode observer," *Energies*, vol. 11, p. 1211, 05 2018.

[13] L. Fridman, J. A. Moreno, B. Bandyopadhyay, S. Kamal, and A. Chalanga, *Continuous Nested Algorithms : The Fifth Generation of Sliding Mode Controllers*, pp. 5–35. Cham: Springer International Publishing, 2015.

[14] S. Kamal, J. A. Moreno, A. Chalanga, B. Bandyopadhyay, and L. M. Fridman, "Continuous terminal sliding-mode controller," *Automatica*, vol. 69, pp. 308–314, 2016.

[15] V. Torres-González, T. Sanchez, L. M. Fridman, and J. A. Moreno, "Design of continuous twisting algorithm," *Automatica*, vol. 80, pp. 119–126, 2017.

[16] S. Zhang, X. Wang, C. Li, and D. Xiao, "Robust state of charge estimation for battery with self-adaptive super twisting sliding mode observer," in *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1–6, 2023.

[17] I.-S. Kim, "The novel state of charge estimation method for lithium battery using sliding mode observer," *Journal of Power Sources*, vol. 163, no. 1, pp. 584–590, 2006. Special issue including selected papers presented at the Second International Conference on Polymer Batteries and Fuel Cells together with regular papers.

[18] V. Behnamgol, M. Asadi, M. Mohamed, S. Aphale, and M. Faraji-Niri, "Comprehensive review of lithium-ion battery state of charge estimation by sliding mode observers," *Energies*, vol. 17, p. 5754, 11 2024.

[19] Y. Feng, C. Xue, Q.-L. Han, H. Fengling, and J. Du, "Robust estimation for state-of-charge and state-of-health of lithium-ion batteries using integral-type terminal sliding-mode observers," *IEEE Transactions on Industrial Electronics*, vol. PP, pp. 1–1, 05 2019.

[20] O. Rezaei, H. A. Moghaddam, and B. Papari, "A fast sliding-mode-based estimation of state-of-charge for lithium-ion batteries for electric vehicle applications," *Journal of Energy Storage*, vol. 45, p. 103484, 2022.

[21] T. Glad and L. Ljung, *Control Theory: Multivariable and Nonlinear Methods*. 11 New Fetter Lane, London EC4P 4EE: Taylor & Francis, 2000.

[22] H. K. Khalil, *Nonlinear Systems, Third Edition*. Upper Saddle River, NJ 07458: Prentice Hall, 2002.

[23] E. Petri, T. Reynaudo, R. Postoyan, D. Astolfi, D. Nešić, and S. Raël, "State estimation of an electrochemical lithium-ion battery model: improved observer performance by hybrid redesign*," in *2023 European Control Conference (ECC)*, pp. 1–6, 2023.

[24] G. L. Plett, *Battery Management Systems, Volume 1: Battery Modeling*. 685 Canton Street, Norwood, Ma 02062: Artech House, 2016.

[25] G. L. Plett, *Battery Management Systems, Volume 2: Equivalent Circuit Methods*. 685 Canton Street, Norwood, Ma 02062: Artech House, 2015.

[26] N. A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, and A. Kojic, "Algorithms for advanced battery-management systems," *IEEE Control Systems Magazine*, vol. 30, no. 3, pp. 49–68, 2010.

[27] M. Doyle, T. F. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *Journal of The Electrochemical Society*, vol. 140, p. 1526, jun 1993.

[28] V. Ramadesigan, P. W. C. Northrop, S. De, S. Santhanagopalan, R. D. Braatz, and V. R. Subramanian, "Modeling and simulation of lithium-ion batteries from a systems engineering perspective," *Journal of The Electrochemical Society*, vol. 159, p. R31, jan 2012.

[29] S. Santhanagopalan, Q. Guo, P. Ramadass, and R. White, "Review of models for predicting the cycling performance of lithium ion batteries," *Journal of Power Sources*, vol. 156, pp. 620–628, 06 2006.

[30] V. Sulzer, S. G. Marquis, R. Timms, M. Robinson, and S. J. Chapman, "Python battery mathematical modelling (PyBaMM)," *Journal of Open Research Software*, vol. 9, no. 1, p. 14, 2021.

[31] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ: Princeton University Press, 2008. Version v2.11b.

[32] S. Säarkkä and L. Svensson, *Bayesian Filtering and Smoothing*. Cambridge University Press, second ed., 2023.

[33] V. I. Utkin, *Sliding Modes in Control and Optimization*. Berlin, Heidelberg: Springer-Verlag, 1992.

[34] A. Levant and M. Livne, "Robust exact filtering differentiators," *European Journal of Control*, vol. 55, pp. 33–44, 2020. Finite-time estimation, diagnosis and synchronization of uncertain systems.

[35] A. F. Filippov, "Differential equations with discontinuous righthand sides," in *Mathematics and Its Applications*, 1988.

[36] Volvo GTT, "Cell data," 2025. Accessed: 2026-02-20.

[37] A. Mohan, P. V. Manitha, and U. Subramaniam, "Comparative analysis of state of charge estimation methods for li-ion batteries using kalman filter variants and machine learning techniques for electric vehicle applications," *IEEE Access*, vol. 13, pp. 177382–177404, 2025.

[38] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

A

ECM parameters

This appendix shows the dynamic parameters of the ECM used in the observers. All variables are derived from the offline parameter identification procedure described in Section 3.1.2. While t (R_0) is mapped as a nonlinear function of SOC, the transient polarization parameters (R_1, R_2, τ_1, and τ_2) capture dual dependencies on both the instantaneous SOC and the applied load current. These multi–dimensional relationships are implemented as look–up tables in the Framework. The parameters and their dependence are illustrated in Figures A.1–A.5.

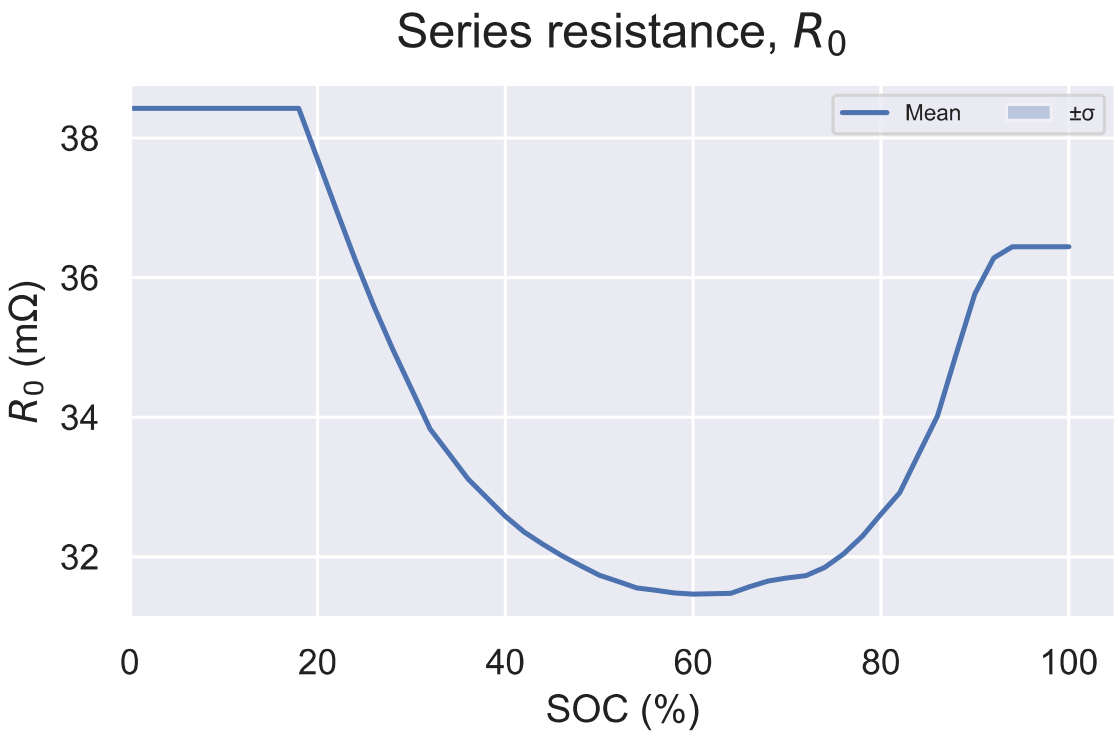


Figure A.1: Dynamic ECM parameter R_0. Dependent on SOC.

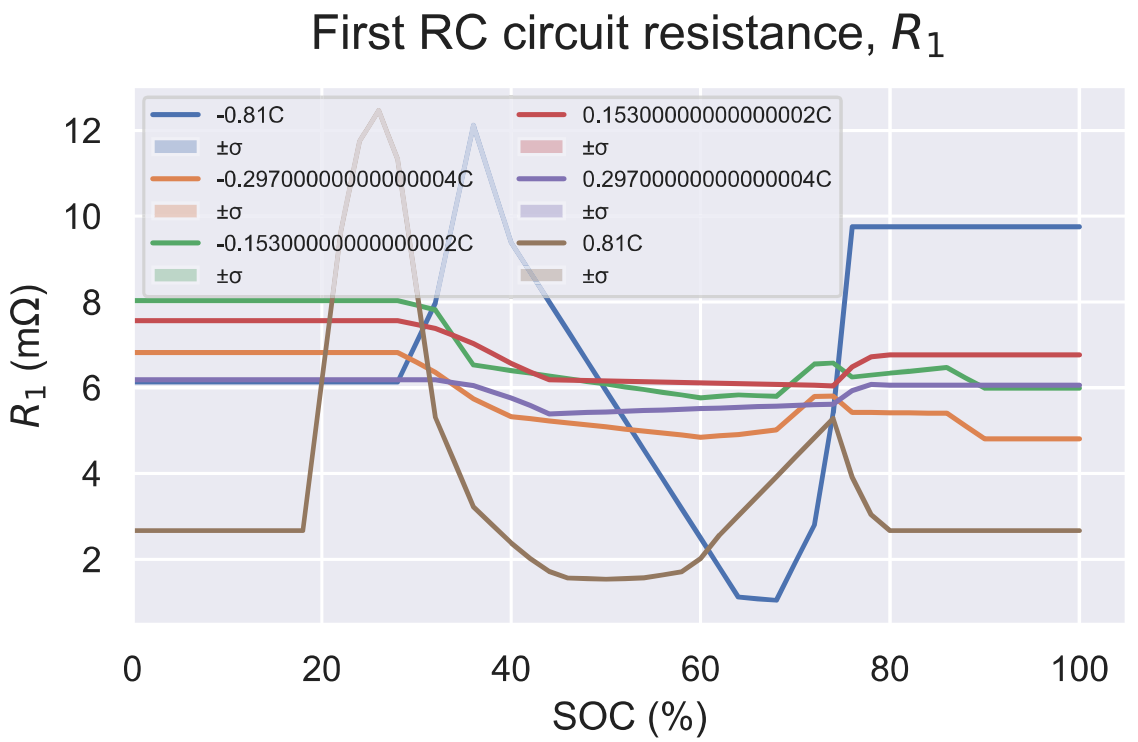


Figure A.2: Dynamic ECM parameter R_1. Dependent on SOC and current. The different colors in the graph indicate the C–rate.

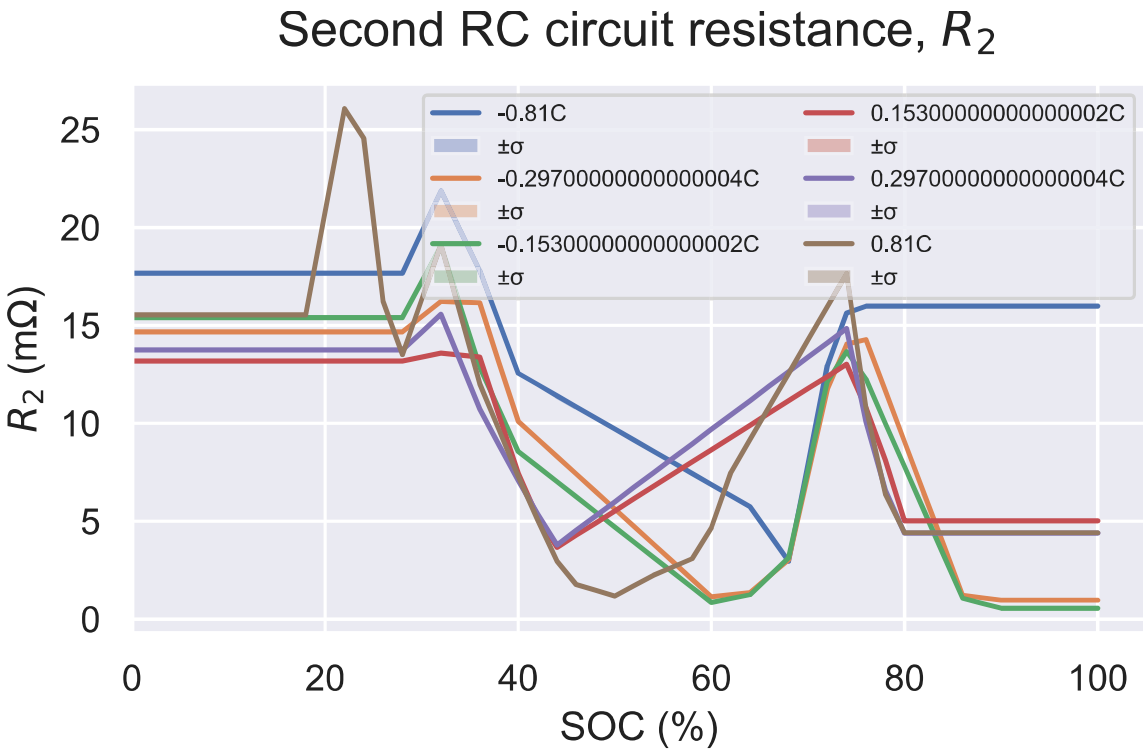


Figure A.3: Dynamic ECM parameter R_2. Dependent on SOC and current. The different colors in the graph indicate the C–rate.

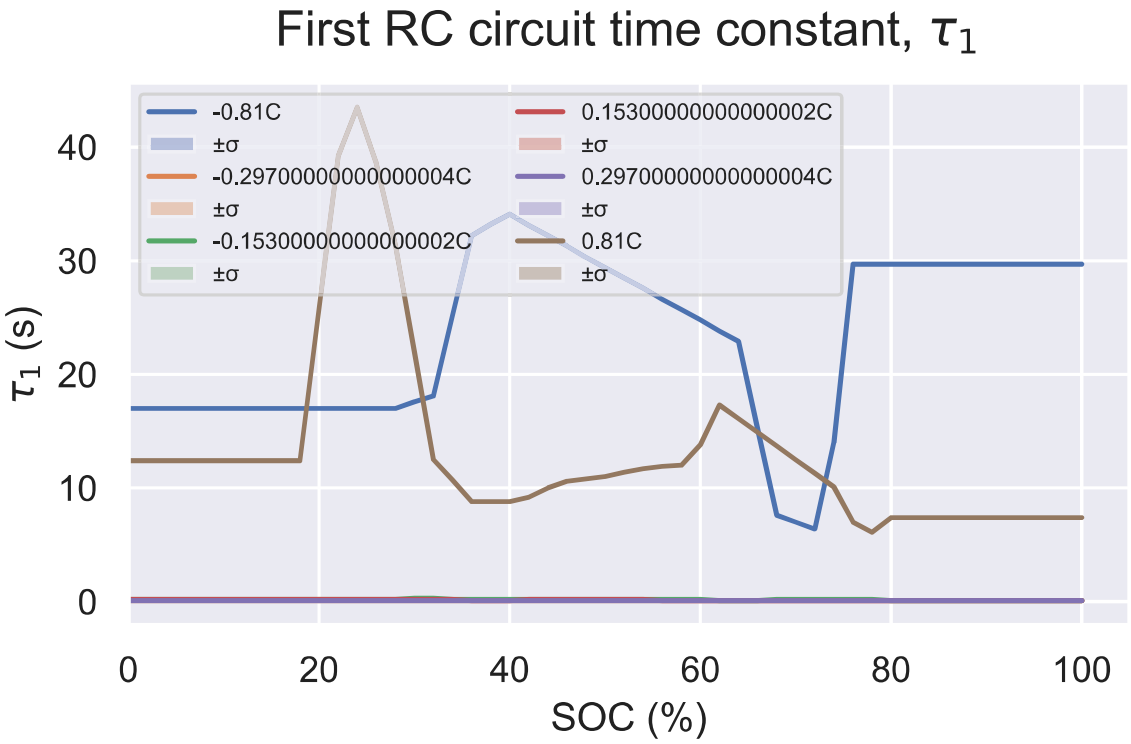


Figure A.4: Dynamic ECM parameter τ_1. Dependent on SOC and current. The different colors in the graph indicate the C–rate.

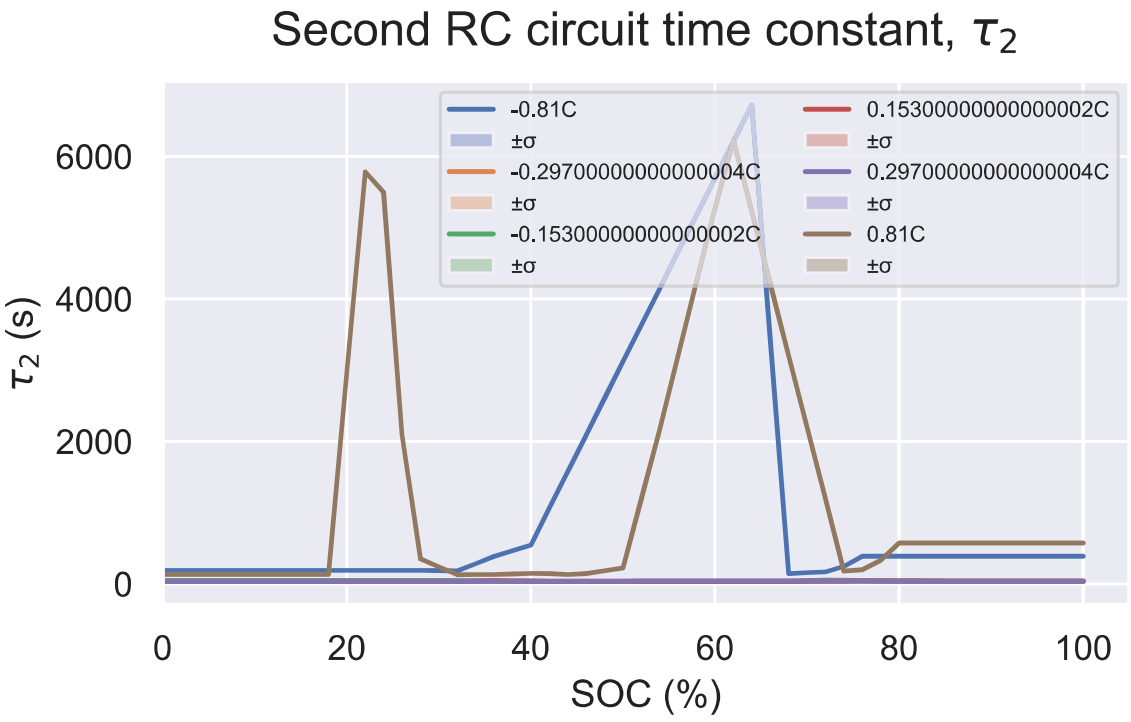


Figure A.5: Dynamic ECM parameter τ_2. Dependent on SOC and current. The different colors in the graph indicate the C–rate.

B

Observer SOC tracking with introduced uncertainty

This appendix shows the SOC tracking for the different observers with introduced parameter uncertainty. The absolute state tracking performance under severe variations in the ohmic internal resistance and nominal capacity is illustrated in Figure B.1. Furthermore, Figure B.2 displays the corresponding estimation trajectories when uncertainties are applied directly to the dynamics.

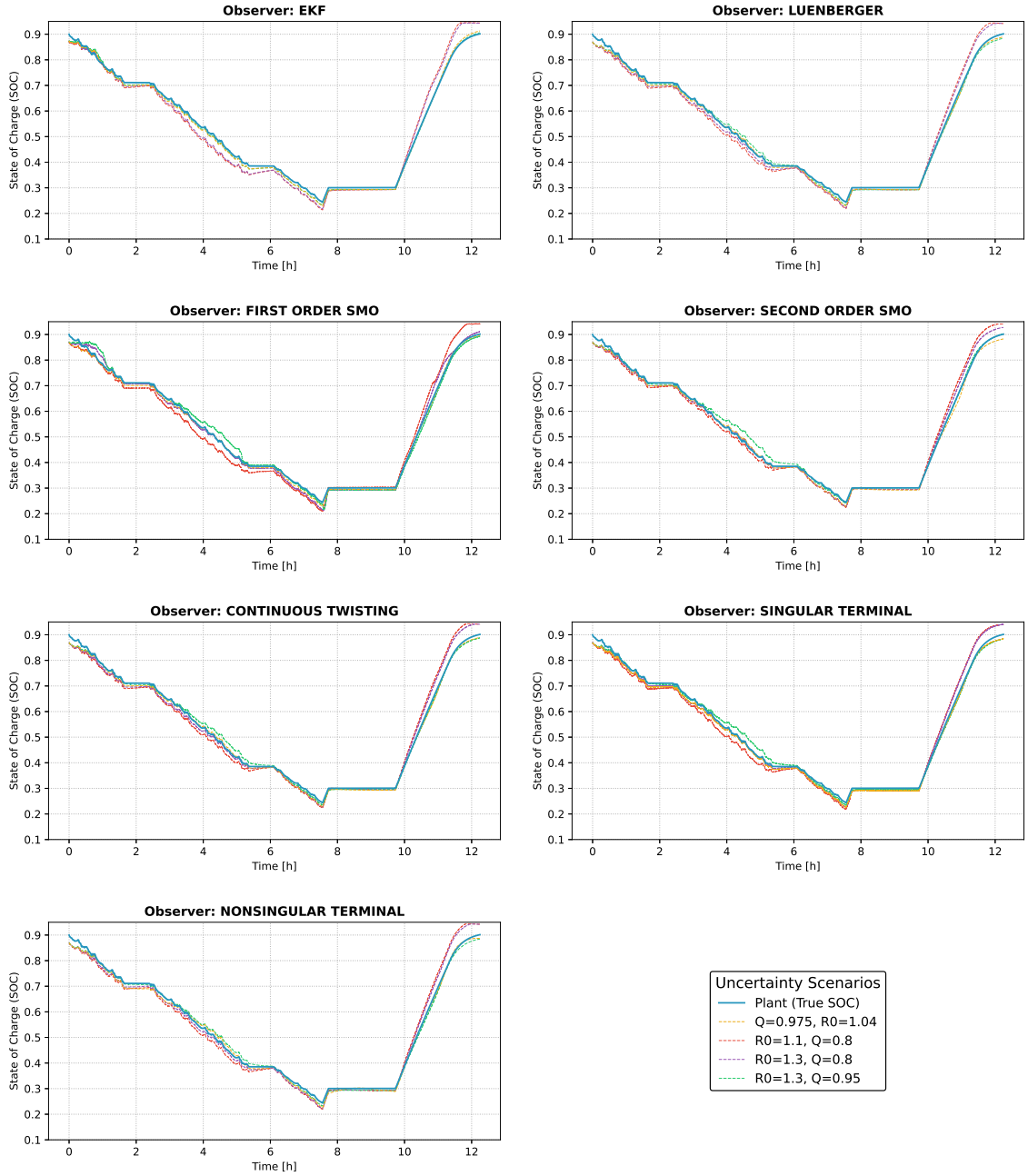


Figure B.1: Observer performance with introduced uncertainties in Ohmic resistance and capacity case.

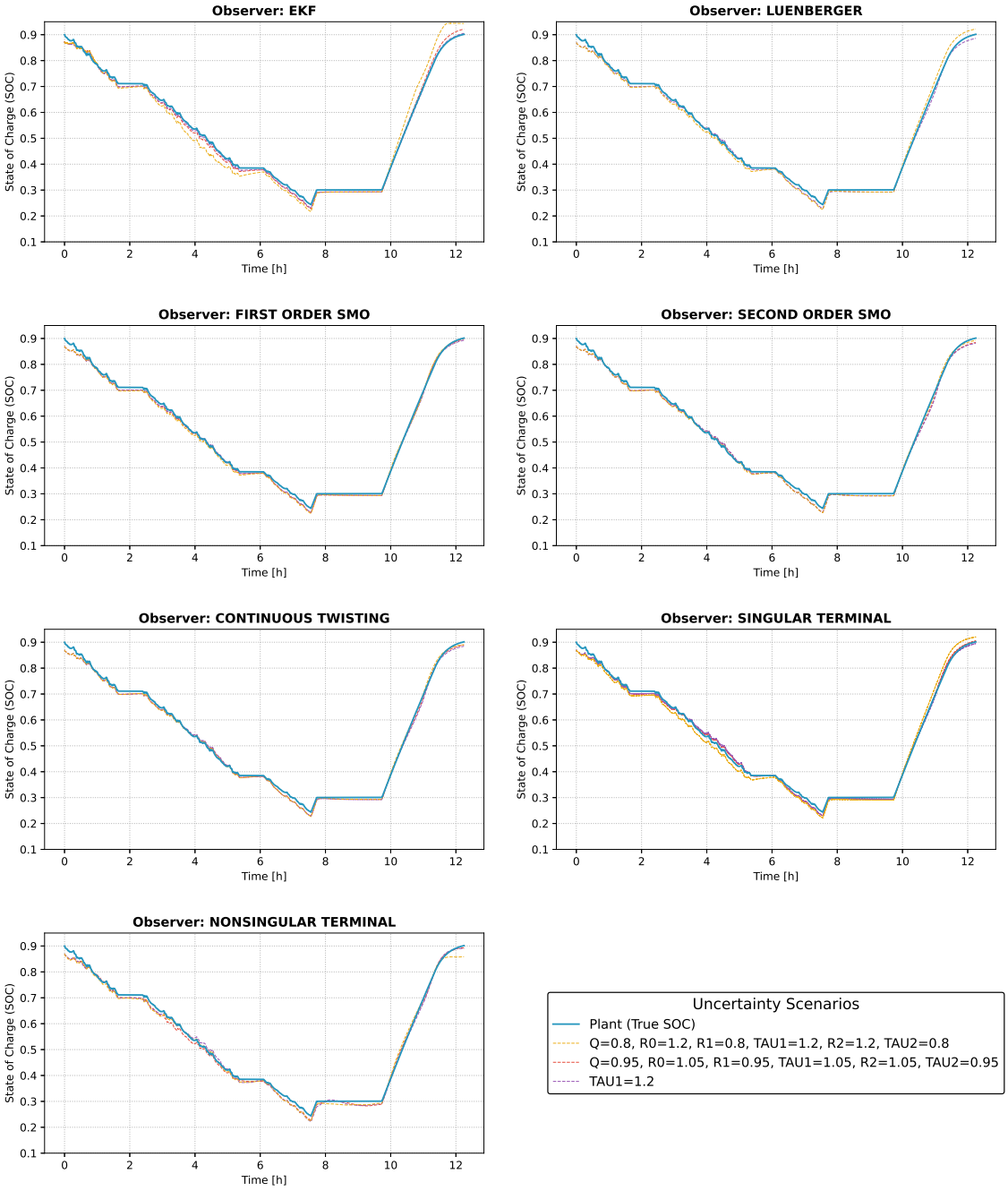


Figure B.2: Observer performance with introduced uncertainties in the dynamics case.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY