CELL-INTEGRATED SENSING FUNCTIONALITIES FOR SMART BATTERY SYSTEMS WITH IMPROVED PERFORMANCE AND SAFETY

GA 957273

D4.4 – ADVANCED MODULE-LEVEL STATE ESTIMATORS BASED ON LEVEL-1 SENSORS

LC-BAT-13-2020 - Sensing functionalities for smart battery cell chemistries

Deliverable No.	D4.4	
Related WP	4	
Deliverable Title	Advanced module-level state estimators based on level-1 sensors	
Deliverable Date	04-12-2023	
Deliverable Type	Report	
Dissemination level	Public (PU)	
Written By	Josu Olmos (IKE) Taranjitsingh Singh (FM) Henk Jan Bergveld (TUE-NXP-NL)	30-10-2023 30-10-2023 30-11-2023
Checked by	Joris De Hoog (FM)	30-10-2023
Reviewed by	Martin Wenger (FHG) Joris De Hoog (FM)	30-11-2023 30-11-2023
Approved by	Iñigo Gandiaga (IKE)	04-12-2023
Status	Final	04-12-2023

Summary

This deliverable report describes the development of baseline models and algorithms for SOC, SOH and SOS algorithms, as well as SOE and SOP baseline algorithms that are closely related to the baseline SOC algorithm. These baseline algorithms only use voltage, current and temperature measurements, and the applied models are therefore fitted only on these available measurement variables. This deliverable report also validates the performance of these baseline state estimation algorithms on measurement data obtained from baseline cells, which are regular 5Ah pouch cells of which voltage current and external temperature measurements are available in three different datasets. These data sets include a short data set to parameterize a linear parameter-varying model to enable SOC and SOH indication, a safety-test related dataset (including cells being brought into thermal runaway to validate safety-related state estimation algorithms) and a longer data set in which cells were cycled in blocks of 50 cycles, in between which check-up cycles were used to characterize the cells, e.g. to obtain cell capacity values. This latter dataset was used to validate the performance of the baseline SOH algorithm.

The aim of the SENSIBAT project is to improve these baseline algorithms by applying additional two-dimensional temperature and mechanical pressure measurements that would be obtained from a so-called level-1 sensor integrated into the cell. However, since cells with this level-1 sensor were not available for experiments at the time of writing this deliverable report, an alternative was used in the form of the mentioned longer dataset, in which besides normal voltage, current and temperature measurements, also an externally measured one-dimensional mechanical pressure measurement was available. This report describes in general how the baseline SOC, SOH and SOS algorithms should be changed to take the eventual two-dimensional temperature and mechanical pressure data from the real level-1 sensors into account. However, since only validation data with externally measured one-dimensional pressure data was available, validation results are only shown for SOH and SOS estimation. In both cases, it is shown that having additional measurements available helps to increase the accuracy of the algorithms.

The present document exhibits a slight deviation from the objectives outlined in the Grant Agreement. Due to the absence of a 6-cell module with the SENSIBAT L1 sensor, necessary for the validation tasks, the algorithms presented herein are constrained to utilizing pressure measurements for battery state optimization at the cell level. However, based in previous experiences the main findings established at the cell remain valid when extrapolated to the module.

Table of Contents

Table of Figures

Abbreviations

Abbreviation	
SOC	*State of Charge*
SOH	*State of Health*
SOE	*State of Energy*
SOP	*State of Power*
SOS	*State of Safety*
RC	*Resistance-Capacitance*
ECM	*Equivalent-Circuit Model*
OCV	*Open-Circuit Voltage*
L1	*Level 1*
L2	*Level 2*
LPV	*Linear Parameter Varying*
EKF	*Extended Kalman Filter*
EMF	*Electro Motive Force*
dEMF	*dEMF/dSOC, the jacobian of EMF(SOC) function*
RMSE	*Root Mean Square Error*
LPV-IO	*Linear Parameter Varying - Input Output*
WLS	*Weighted Least Squares*
BOL	*Beginning Of Life*
EOL	*End Of Life*
ECT	*Electrochemical-thermal model*
FEM	*Finite-Element Method*
SEI	*Solid Electrolyte Interphase*

1 Introduction

In the domain of high-performance battery management, accurately assessing a battery's health and safety is crucial. Due to their intricate nature, modern batteries are complex, and it's not possible to measure these conditions or states directly. Therefore, it is necessary to develop smart algorithms to help understand how a battery is working by estimating these states based on the basic voltage, current or temperature measurements.

When it comes to assessing the overall battery state accurately, there are three key pivotal states: the State of Charge (SOC), the State of Health (SOH) and the State of Safety (SOS). Obtaining direct measurements of these states is not possible, but their significance remains important: SOC defines how much charge is left in the battery, SOH defines its overall health status, and SOS serves as a safeguard against unsafe operating conditions. See SENSIBAT deliverable reports D1.1 (section 4.6) [1] and D1.2 [2] (terms and definitions at the start of the report) for formal definitions of all considered states in the project.

The assessment of battery health or SOH, in particular, is quite challenging. SOH is typically broken down into two components: the first one is about how much capacity the battery has lost ($SOHq = Q_{actual}/Q_n$), where Q_n represents the nominal capacity, and the other relates to how much power the battery can still deliver based on the increase in resistance ($SOHr = R_{actual}/R_n$), where R_n represents the nominal resistance. Somehow, the first component can be also linked to the State of Energy (SOE), while the second component is related to the State of Power (SOP). These two components provide a good picture of how well the battery can still perform.

The goal of this deliverable D4.4 is to improve the performance of typical Battery Management Systems (BMS) through the utilization of advanced state estimation algorithms. Improved SOC and SOH algorithms are developed and validated, which make use of data from the battery built-in novel sensor technology developed in SENSIBAT project (Level-1, or L1 sensor technology, including 2-dimensional (2D) temperature and mechanical pressure inside the pouch cell). At the same time, in the journey towards an increased battery safety, this deliverable also addresses the battery safety concept through the development of an SOS algorithm. The objective of this algorithm is not only to understand what's happening inside the battery, but also prevent dangerous situations before they occur. Similar to SOC and SOH algorithms, in the case of SOS an improved version of the algorithm with the aggregation of the data provided by the L1 sensor is also proposed and validated.

The document is structured as follows. First, in section 2 the datasets used for the development and validation of the algorithms are presented. Due to a set of specific circumstances of SENSIBAT project, not all the expected datasets were available at the time the algorithms were developed. Consequently, different datasets were used for baseline and L1 algorithms, and for the different algorithms. With the aim of demonstrating the value of the built-in novel sensor technology of the SENSIBAT project, the development of the state algorithms is broken down into two steps. On the one hand, baseline algorithms are developed in section 3, which rely on state-of-the-art measurement technologies, basically voltage, current and temperature. On the other hand, in a second step, and in order to validate the benefit of SENSIBAT L1-sensor technology, improved state algorithms are presented in section 4, which rely on typical voltage and current measurements, but also on the internal 2D mechanical pressure and temperature measurements provided by the L1 sensors. The results of these improved state algorithms are compared against the baseline algorithms, in order to properly evaluate the value of the mentioned sensors. Finally, conclusions are drawn in section 5.

2 Datasets for development and validation of algorithms

As mentioned in the introduction, due to specific circumstances in the context of the SENSIBAT project, different datasets were used during the development and validation of the different algorithms. Specifically, three distinct datasets were used:

- Dataset 1: The first dataset was provided by Eindhoven University of Technology (TU/e) and consists of a set of experiments held to the 5 Ah baseline SENSIBAT NMC cell (so without the added L1 sensors). Measurements were done as required in D1.2 [2], but also a Linear Parameter-Varying (LPV) model was constructed based on these measurements according to the methodology presented in [3]. This model has been used in the development and validation of the baseline SOC, SOP and SOE algorithms.
- Dataset 2: The second dataset consists of the safety tests carried out during the SENSIBAT project, which were performed by IKERLAN. The main findings of these tests are presented in Deliverable 5.1 [4]. The safety tests consisted of three routines (nail penetration test, heating test and overcharge test) and were performed to the three cells that were used in SENSIBAT project (baseline cell, L1 cell and L2 cell). This dataset has been used in the validation of the baseline SOS algorithm. As will be explained in Section 4.3, it was not possible to use this dataset to validate the L1-SOS algorithm.
- Dataset 3: The final dataset consists of the ageing tests carried out during the SENSIBAT project, which were performed by IKERLAN. The test routines were specified in Deliverable 1.2 [2], and the results of these tests are presented in Deliverable 5.1 [4]. These tests were performed to two types of cells developed in SENSIBAT project, both of 5Ah: baseline cell and L1 cell. The objective was to evaluate the impact of L1 sensor technology on the useful life of the cells. The tests consisted of a sequence of performance and cycling tests, from beginning of life until the SOH dropped around approximately 75%. Due to the unavailability of L1 sensor read-out circuits when executing the tests to the L1 cell, the temperature and the pressure were measured from external sensors, with a single measurement point per cell. In order to differentiate between the tests held to the baseline and to the L1 cell, two sub-datasets are defined:
    - Dataset 3.1: Ageing tests to baseline cell. This dataset has been used for the validation of the baseline SOH algorithm.
    - Dataset 3.2: Ageing tests to L1 cell. This dataset has been used for the validation of the baseline SOH algorithm, but also for the development and validation of L1-SOC, L1-SOH and L1-SOS algorithms.

3 Baseline algorithms

In order to develop novel state algorithm based on L1 sensor technology, the first step consists of developing the baseline algorithms that will provide a benchmark performance. In this section, baseline SOC, SOH and SOS algorithms are developed and validated. SOE and SOP algorithms based on the output of SOC are also proposed and tested.

3.1 SOC

There are several algorithms available for estimation of the SOC. These algorithms can be divided into following methods:

1. OCV look-up tables
2. Coulomb-counting method
3. Observer-based method
4. Data-driven method
5. Filter-based method

Although the **OCV look-up tables method** and **Coulomb-counting method** offer computational simplicity, they have low accuracies due to using very simple battery models such as a linear capacitor for the Coulomb-counting method and have poor robustness to sensor errors. The **Observer-based method** on the other hand offers robustness to sensor errors, however, it is sensitive to model inaccuracies and complexity of the model chosen for obtaining observers. **Data-driven methods** based on machine-learning concepts occupy the vast on-going research on SOC estimation. Although they show much promise, it is extremely difficult to find the right trade-off between avoiding underfitting, avoiding overfitting and computational complexity [5].

The estimation algorithm must be able to adapt to changing cell characteristics as the cell ages and must be able to provide accurate estimates over the lifetime of the pack [6]. **Filter-based methods**, apart from yielding robustness to model inaccuracies due to their ability to correct and update estimation, are preferred methods and hence the proposed method in this report. The first step for this method is to obtain a model.

3.1.1 Model for SOC estimation

The 1RC Equivalent-Circuit Model (ECM) captures the dynamics of battery quite well. Usually, the 1RC ECM model consists of single ohmic series resistance and one RC time constant and these parameters are not fixed, rather parameter-varying, with scheduling parameters as SOC and possibly temperature.

The overpotential model applied in this report is shown below:

$$\begin{aligned} \boldsymbol{o_{k+1} = \theta^1 o_k + \theta^2 u_k} \\ \boldsymbol{ov_k = o_k + \theta^3 u_k} \end{aligned} \qquad (3\text{-}1)$$

where o_{k+1} is the dynamic state for overpotential and ov_k is the total overpotential, u_k is the input load current, and θ^1, θ^2 and θ^3represent $A(SoC,T)$, $B(SoC,T)$ and $D(SoC,T)$ in a common state-space model of the battery such as the one shown below in (3-2), which are the Linear Parameter-Varying (LPV) matrices/functions with dependency on SOC and temperature T. The terminal voltage is the output of the 1RC ECM model and is denoted by y_k. The full expression is shown in the equation below:

$$\boldsymbol{y_k = ov_k + EMF(SOC,T)} \qquad (3\text{-}2)$$

as in [3]. In order to obtain the function $EMF(SOC, T)$ and the parameters θ_1, θ_2 and θ_3, we follow these steps: a) Conduct experiments where we record Voltage, Current and Temperature. b) Calculate SOC using a high-accuracy current sensor and apply coulomb-counting. c) Determine overpotential by subtracting the EMF derived from the measured data, from the terminal voltage. d) Use the data to fit the parameters θ_1, θ_2 and θ_3. The resulting model is a Linear Parameter-Varying – Input-Output (LPV-IO) model and its done straightforward by throwing data to the LPVCore toolbox [7]. Polynomial basis functions are used to fit the dependency of parameters θ_1, θ_2 and θ_3 on SOC and T and the model parameters have been identified by TU/e using the methodology described in [3]. This has also been validated on Dataset 1.

3.1.2 SOC Estimation Algorithm

Until now we may have everything needed to start with our estimation of the SOC, i.e., the 1RC ECM model including SOC and T dependency of its parameters from TU/e. The model also includes the EMF function as B-splines, and the gradient dEMF which is used for the estimation. Figure 1 represents the EMF and dEMF functions with dependencies on SOC and T.

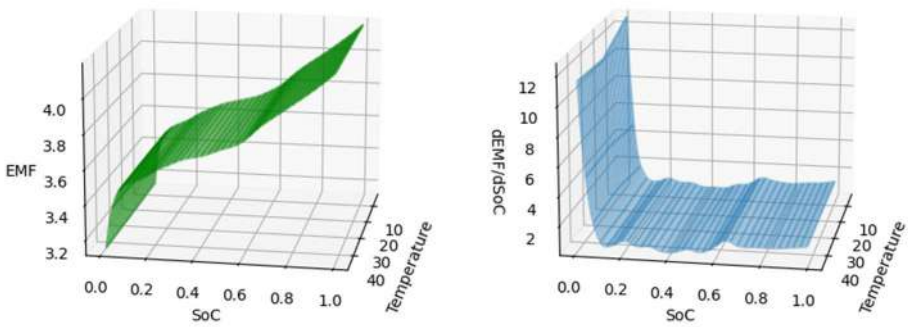

Figure 1 EMF and dEMF functions yielded with dependencies on SOC and Temperature

The next natural step is to obtain SOC estimation using an Extended Kalman Filter (EKF). The ordinary KF is not feasible because of observability issues [8]. The EKF is a widely known algorithm, however the process noise covariance matric Q_k is often constant. In our estimation algorithm we use an adaptive Q_k every iteration of estimation to cope up with the modelling error. The first step is to get the complete 1RC ECM state-space model for the EKF that includes SOC and the overpotential o as states, which is shown below:

$$\begin{bmatrix} SOC_{k+1} \\ o_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \theta^1 \end{bmatrix} \begin{bmatrix} SOC_k \\ o_k \end{bmatrix} + \begin{bmatrix} \frac{\Delta t_{SOC}}{Q_{cap_{nom}}} \\ \theta^2 \end{bmatrix} u_k \qquad (3\text{-}3)$$
$$y_k = EMF(SOC_k, T_k) + o_k + \theta^3 u_k$$

where Δt_{SoC} and $Q_{cap_{nom}}$ represents the sampling time and nominal Capacity, respectively; and θ^1, θ^2 and θ^3 represent $A(SOC_k, T_k)$, $B(SOC_k, T_k)$ and $D(SOC_k, T_k)$, respectively. This modified EKF algorithm that takes inputs as current, voltage, and previous states (SOC and overpotential state) yields the estimated states i.e., SOC and overpotential every Δt_{SoC} seconds. The algorithm is represented visually in Figure 2 and shown in detail in Annex A – EKF Algorithm.

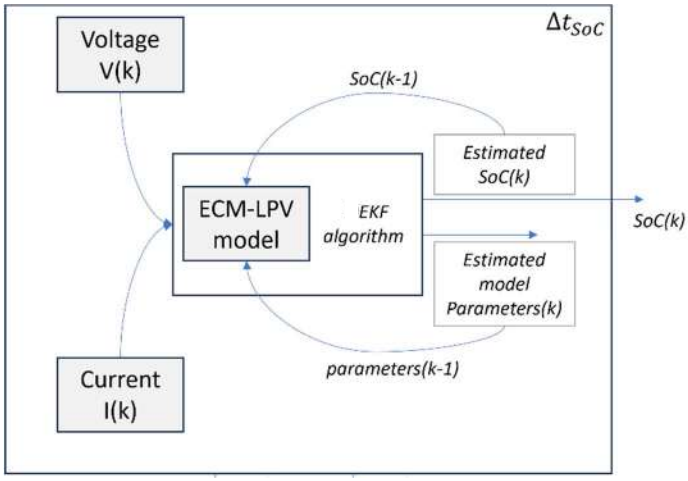

Figure 2 Visual representation of the EKF algorithm for baseline SOC estimation

3.1.3 Results and Conclusions

Figure 4 until Figure 6 show empirical results for estimation of SOC and overpotential states. Dataset 1 (data provided by TU/e) has been used on the validation of baseline SOC algorithm. Validation on various profiles from Dataset 1 yielded Root-Mean-Square Error (RMSE) for SOC estimation approximately between **0.5% to 1%,** see Figure 3**.** Another quantitative analysis on these validation data depicted in Figure 5 shows that mostly all the estimation are within 2σ i.e., 95% confidence ellipsoid. Another important validation of the proposed algorithm is performed to understand the sensitivity of the algorithm to different initial conditions. Figure 6 shows the sensitivity of the algorithm to the initial SOC condition of the estimator for a Dataset 1, where real initial SOC was 99%. Figure 6 Sensitivity to initial conditions illustrates that even when starting with an incorrect initial condition of 60%, the RMSE remains below 4%. This unexpected relatively large increase in RMSE with a large deviation from the real value of the initial condition, i.e., 60% instead of the correct 99%, can be attributed to the fact that the RMSE is calculated over a single discharge cycle. In simpler terms, when the SOC estimation algorithm is applied to extensive datasets involving both discharging ang charging of a battery over time, and the initial condition has a significant error (due to errors in measurement), the estimation will eventually approach the actual SOC, leading to a lower RMSE.

These validations have been done on Dataset 1, where there is no evolution of SOH, i.e., $Q_{ca\ \ nom}$ is constant in (3-11). However, the situation where the capacity degradation occurs, i.e., Q_{cap} degrades over time, requires that Q_{cap} needs to be updated. We will later focus on SOH estimation in the subsection 3.4 that will be responsible for the update of Q_{cap} and the results obtained in this section will also act as validation of the proposed SOC algorithm for long datasets where the battery actually ages over time.

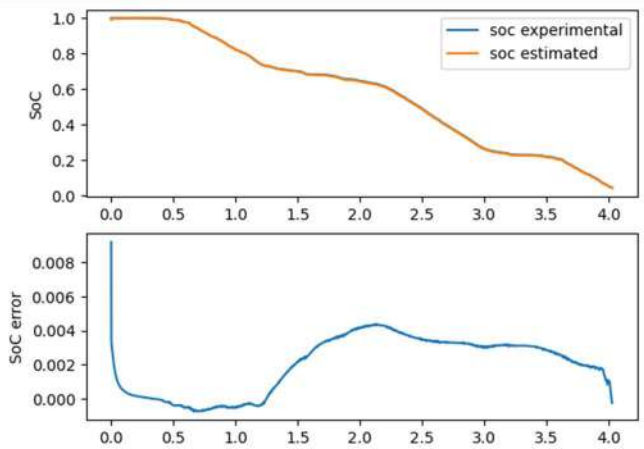

Figure 3 SOC state estimation accuracy performance

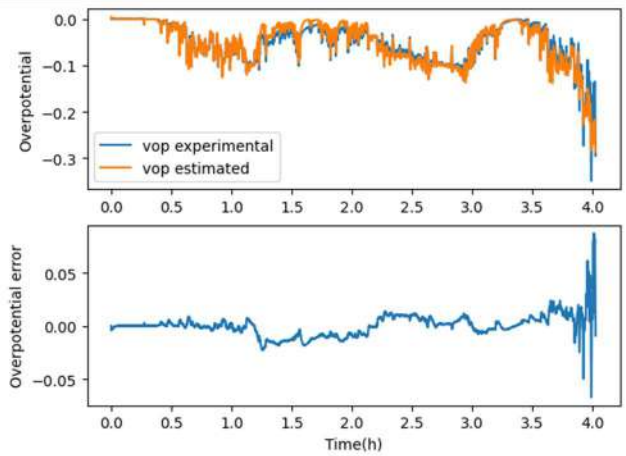

Figure 4 Overpotential state estimation accuracy performance

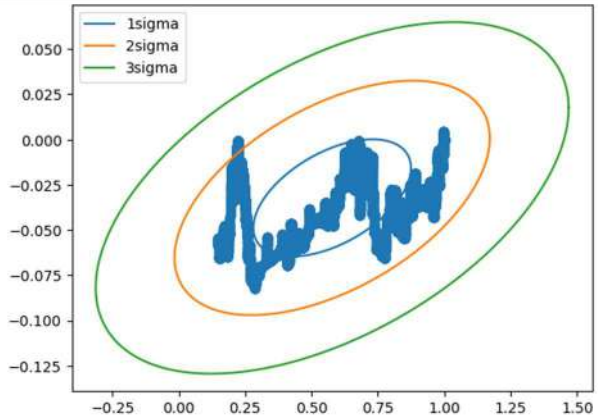

Figure 5 Confidence ellipsoid on a validation data

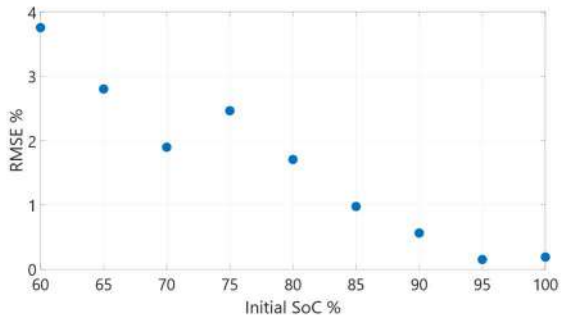

Figure 6 Sensitivity to initial conditions (real initial condition 99%)

3.2 SOE

SOE indicates the remaining energy stored in the battery and it can be defined as integration of power obtained with terminal voltage and current [9]. Figure 7 shows the estimated SOE with the same validation tests used for the SOC algorithm.

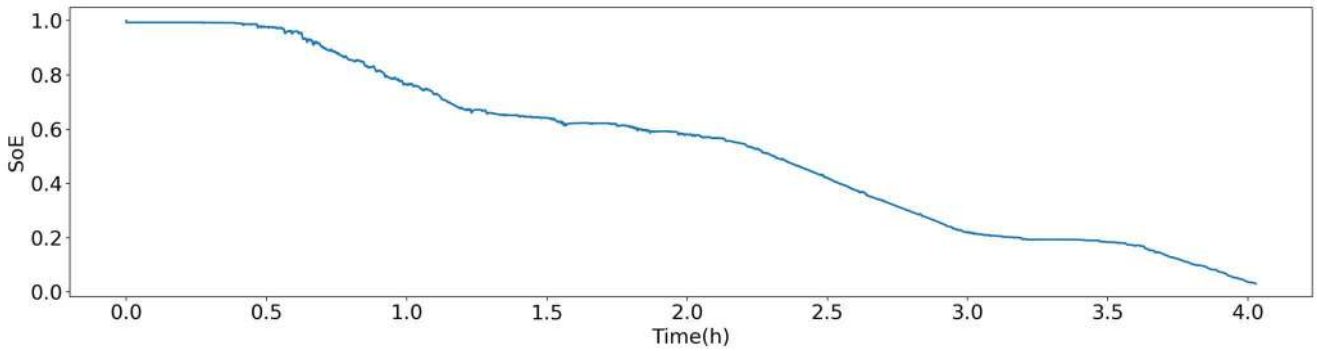

Figure 7 SOE state estimation result on validation dataset.

3.3 SOP

The SOP algorithm is dependent on the SOC estimation and this is merely a function determined by computing the immediate power drawn from the battery. This involves multiplying the current flowing through the battery with the voltage at that specific moment. Subsequently, this calculated power is assessed to ensure it falls within

predefined constraints, like SOC limits, voltage and current limits. This algorithm is currently only considered for discharging but can easily be adapted for charging as well. One can find these algorithms from the research studies such as described in [10] . Figure 8 shows the SOP estimation with the same validation test used for the SOC and SOE algorithms.

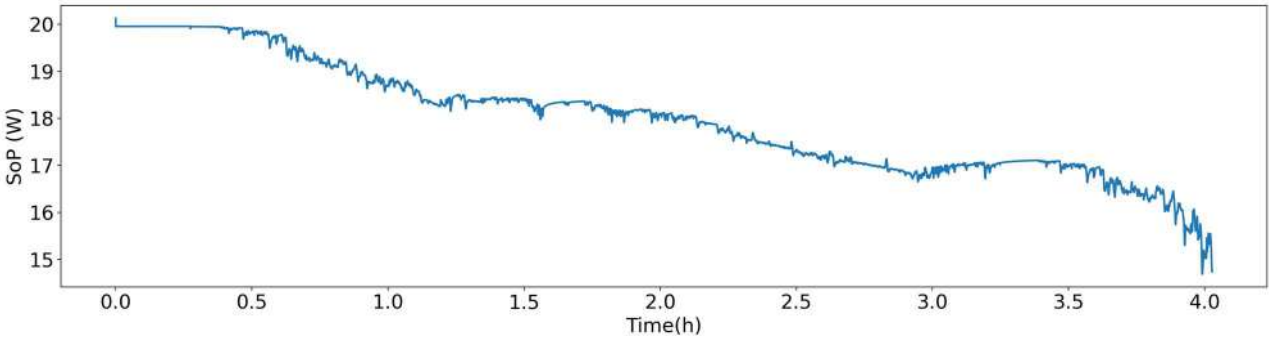

Figure 8 SOP state estimation

Note: The real-time deployable-ready python modules for SOC, SoE, and SoP algorithms are available in a private github repository and can be made available at disposal on request to taranjitsingh.singh@flandersmake.be.

3.4 SOH

3.4.1 SOH Estimation algorithm

SOH estimation plays a crucial role in assessing battery state. Inaccurate SOH estimations yields incorrect SOC and SOE estimations since these rely on an accurate knowledge of the actual value of Q_{cap}. SOH related to capacity loss can be expressed as below:

$$SoH = \frac{Q_{cap}}{Q_{BOL}} \times 100\% \qquad (3\text{-}4)$$

where, Q_{cap} is the cell's capacity at any time instant while, Q_{BOL} represents the cell's capacity at the beginning of the life. The existing SOH estimation algorithms are divided into two methods:

1. Direct measurements
2. Indirect measurements

Direct measurements, such as **capacity measurement tests**, **internal resistance measurements** and **impedance measurements tests** offer accuracy. However, if the checkups are performed in real-time, the current with which capacity is determined by discharging from completely full to completely empty should be carefully chosen. When these currents are chosen too high, using e.g., 3C current profiles, it may contribute in accelerated aging and fast degradation of the battery, as was found when characterizing the chosen cell using the measurement procedure defined in deliverable report D1.2 [2]. On the other hand, sensor-based direct measurements are prone to sensor-based measurement errors and may lead to fluctuations in SOH estimation. Direct measurements such as impedance measurements will be touched upon in Level-2 (L2) novel sensor technology in deliverable D4.5. For baseline estimation methods of SOH, we rather focus on indirect measurements such as **the model-based** and **data-driven method** [5]**.** These methods are more practical than direct measurements, especially for capacity, since a direct capacity measurement in real-time would require the battery to be fully charged and then discharged to get an accurate actual capacity estimate. The proposed algorithm in this deliverable is a hybrid method because it combines the *model-driven filter-based SOC estimation and data-driven SOH estimation*.

In the subsection 3.1, we presented the SOC estimation algorithm and validated it on a dataset (Dataset 1) to analyse the reliability of the algorithm. However, as the battery ages, there is a slow decrease of SOH due to the slow capacity degradation, i.e., time-varying Q_{cap} will decrease in (3-3), and thus it is required to have validation of the proposed algorithm on a longer dataset.

Besides a slow decrease in Q_{cap}, parameters $\theta_1 - \theta_3$ will also change slowly over time related to the increase in impedance that an aging battery experiences on top of the capacity decrease. In that case, the model needs to be updated to accommodate model parameters by taking the dynamics $\theta_{k+1} = \theta_k$ into account, assuming that the model parameters vary slowly over time. This modifies our EKF algorithm mentioned in subsection 3.1 since we no longer estimate the states but also the parameters. This accounts for extending the algorithm in Annex A – EKF Algorithm to call it as Dual-EKF (DEKF) with forgetting factor [8] and this extension is shown in Annex B – DEKF Algorithm.

$$\begin{bmatrix} \boldsymbol{SoC}_{k+1} \\ \boldsymbol{o}_{k+1} \\ \boldsymbol{\theta}^{m}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\theta}^{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{SoC}_{k} \\ \boldsymbol{o}_{k} \\ \boldsymbol{\theta}^{m}_{k} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t_{SOC}}{Q_{cap}} \\ \boldsymbol{\theta}^{2} \\ \mathbf{0} \end{bmatrix} \boldsymbol{u}_k \qquad (3\text{-}5)$$

$$\boldsymbol{y}_k = \boldsymbol{EMF}(\boldsymbol{SoC}_k, \boldsymbol{T}_k) + \boldsymbol{o}_k + \boldsymbol{\theta}^3 \boldsymbol{u}_k$$

The SOH estimation algorithm is represented visually by Figure 9. The steps for the algorithm are briefly mentioned below:

1. SOC estimation, along with estimating parameters;
2. Stacking historical SOC and past current data to obtain a comprehensive dataset;
3. Estimate $\hat{Q}_{cap}$ by finding the optimal estimate by solving a numerical optimization problem on the comprehensive dataset of step 2;
4. Iteratively estimate $\hat{Q}_{cap}$ and eventually estimate SOH using (3-4) until EOL.

The steps in detail for this algorithm are elaborated below:

1. Estimate SOC using DEKF, $\forall m = 2,3$ at Δt_{SOC} (so θ_1 remains fixed and θ_2 and θ_3 are estimated at each time step Δt_{SOC})

Estimate SOC using only current, voltage and temperature measurements. Since we briefly introduced earlier that this estimation algorithm is a DEKF, we also focus on estimating model parameters. In our case we only estimate θ_2 and θ_3 of the model in real-time. The reason behind this assumption is three-fold: 1) The maximum capacity Q_{cap} is not influenced by resistance/overpotential, it links to the maximum amount of Li^+ ions that can be stored in the electrodes. While the battery ages, this maximum capacity Q_{cap} goes down while at the same time, the resistance increases 2) apart from the $EMF(SoC_k, T_k)$ function, we retain the LPV dependency on the temperature T and SOC in θ_1, so we don't need to estimate it as it is given by a function dependent on temperature and SOC 3) we estimate the evolution of θ_2 and θ_3 to overcome modelling mismatch by accommodating for the propagation of modelling errors in the model. Ideally, the most efficient way of treating this evolution of the parameters over time due to aging would be to retain the dependency of θ_1, θ_2 and θ_3 on the scheduling parameters (T and SOC) and estimate the coefficients of the parameter-varying functions. However, this would add extra computational complexity in our proposed SOC estimation algorithm. A nice balance can be thought of, for e.g., adding a gain and an offset on these parameters as function of the T and SOC scheduling parameters, while still retaining the parameter dependency via the LPV functions, but this can be considered as a future research and consideration.

2. Estimate Capacity Q_{cap} at Δt_{SOH}

This method is based on the moving past M SOC and current measurement data points between t_1 and t_2. In order to estimate capacity Q_{cap} using past data, we utilize one of the simplest of regression techniques, i.e., Weighted Least Squares (WLS). The first step is to understand the relationship between SOC and capacity which is shown in the following Coulomb-counting method equation:

$$SoC(t_2) - SoC(t_1) = \frac{1}{Q_{cap}} \int_{t_1}^{t_2} \frac{u(t)}{3600} dt \qquad (3\text{-}6)$$

Consider, $x_i = \int_{t_1}^{t_2} \frac{u(t)}{3600} dt$ and $y_i = SoC(t_2) - SoC(t_1)$, we can transform the equation as follows:

$$y_i = \beta x_i \qquad (3\text{-}7)$$

where, $\beta = \frac{1}{Q_{cap}}$. This means that for a fixed value of (dis)charged coulombs (x_i), its equivalent ΔSoC (y_i) is considered to form the data. By accumulating past M data points from time t_1 to t_2 such that $M = \frac{t_2 - t_1}{\Delta t_{diff}} + 1, | \Delta t_{diff} \geq \Delta t_{SoC}$, the above equation can be expressed in vector form as:

$$\begin{bmatrix} y_{t_1} = y_{t_2 - M\Delta t_{diff}} \\ \vdots \\ y_{t_2 - 2.\Delta t_{diff}} \\ y_{t_2 - \Delta t_{diff}} \\ y_{t_2} \end{bmatrix} = \begin{bmatrix} x_{t_1} = x_{t_2 - M\Delta t_{diff}} \\ \vdots \\ x_{t_2 - 2.\Delta t_{diff}} \\ x_{t_2 - \Delta t_{diff}} \\ x_{t_2} \end{bmatrix} \beta \qquad (3\text{-}8)$$

And in the matrix form it is expressed as $Y = X.\beta$, and the estimated parameter $\hat{\beta}$ can be obtained by minimizing the weighted sum of squares, such that the function is denoted as follows:

$$\chi_{WLS}\left(\widehat{Q}_{Cap_k}\right) = \sum_{i=1}^{M} \| W\,(Y - X\,\widehat{\beta}_k) \|_2 \qquad (3\text{-}9)$$

3. Estimate $\widehat{Q}_{Cap_k}$ at every time instant with sampling time of $\Delta t_{SoH} | \Delta t_{SoH} \geq \Delta t_{SoC}$, using an optimization algorithm in (3-10) and interpolate it with zero-order hold towards SOC estimation if $\Delta t_{SoH} > \Delta t_{SoC}$. In the case of $\Delta t_{SOH} = \Delta t_{SOC}$, we can still obtain one point initially and use (3-7), until we collect M data points and M data points shift forward in time as new data is added.

$$\begin{aligned} &\arg\min_{\widehat{Q}_{Cap_k}} \chi_{WLS}\left(\widehat{Q}_{Cap_k}\right) \\ &\text{subject to: } Q_{min} \leq \widehat{Q}_{Cap_k} \leq Q_{max} \end{aligned} \qquad (3\text{-}10)$$

Note: In the optimization problem formulation (3-10), we have added a constraint on the estimated capacity $\widehat{Q}_{Cap_k}$ such that it is estimated within a chosen minimum and maximum capacity, to avoid any numerical issues that may arise from solving the optimization problem.

4. Yield $\widehat{Q}_{Cap_k}$ either from the estimation optimization algorithm or from zero-order hold and eventually obtain estimated SOH and update it in the model (3-5) and go back to the step 1 to estimate SOC using DEKF until the battery's first EOL. This algorithm is visually represented as in Figure 9.

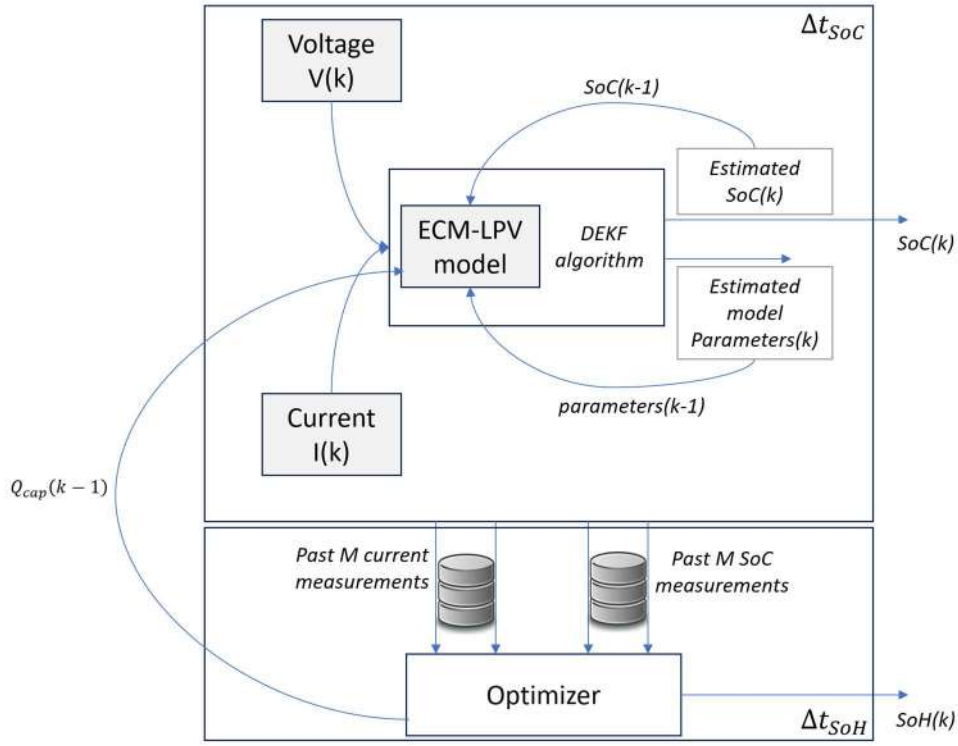

Figure 9 Visual representation for baseline SOH estimation method

So, in simpler terms, initially, we start with a known value, Q_{BOL}. Then, for the next Δt_{diff} instances, we gather SOC data at Δt_{diff} second intervals. Using these sets of Δt_{diff} SOC data points, we calculate the first updated value. This process is repeated, continually extending the vectors until they contain $2\Delta t_{diff}$ entries. At this point, we perform the next Q_{cap} estimation based on fitting a total of $2\Delta t_{diff}$ data points. We repeat this cycle until we've collected M data points in our vectors. After reaching this milestone, the vectors will always maintain a length of M entries, but they will shift forward in time as new data is added.

3.4.2 Results and Conclusions

As mentioned in Section 2, the baseline SOH algorithm is validated with Dataset 3.1 and Dataset 3.2, which correspond to the ageing tests performed to the 5Ah cell used in the SENSIBAT project. As defined in the procedures of Deliverable 1.2 [2], these tests consist of a series of cycling and performance tests. The cycling phases imply that the cells are charged-discharged in consecutive 50 cycles. In between these series of 50 charge/discharge cycles, the performance tests are executed, which determine the actual capacity and the internal resistance value of the cells.

The difference between Dataset 3.1 and Dataset 3.2 is that the first data set only has baseline measurements (V, I, T), which implies cells without added sensors were measured, while the second data set also includes measurements obtained with external pressure sensors. This second Dataset 3.2 was measured with external sensors since L1 read-out circuits were not yet available. This would eventually aid us to extend the above-mentioned algorithms to include real L1 sensor measurements and compare it to the baseline estimation methods that do not use additional sensor inputs.

For SOH estimation, we first consider the SOC estimation window of 85% to 15% during every available discharge cycle with a random noise of 0.2% added to the SOC starting point of the window and the end of the window to avoid the same SOC error accumulation. As for the results shown below, we chose $\Delta t_{diff} = 10s, M = 10^3$ and $\Delta t_{SoH} = 10s$. Eventually, in the next subsection, we do a sensitivity analysis on these hyperparameters

to see their impact on accuracy. We obtain the estimation of Q_{cap} and this estimation is used as the input to the next SOC estimation step. In our case, $\Delta t_{SoH} > \Delta t_{SoC} \mid \Delta t_{SoC} = 1$, this means that for the next steps of SOC estimation where we do not perform Capacity estimation, we perform zero-order hold, or in other words we assume the last estimated capacity as the input to the SOC algorithm until the next capacity value $\widehat{\boldsymbol{Q}}_{\boldsymbol{Cap}}$ is estimated 10 seconds later.

In essence, the process begins with an initial known value, $\widehat{\boldsymbol{Q}}_{\boldsymbol{Cap_1}} = Q_{BOL}$, and for each subsequent time interval of 10 seconds, SOC data is gathered. Utilizing sets of 10 SoC data points, we calculate an updated $\widehat{\boldsymbol{Q}}_{\boldsymbol{Cap_2}}$ value. This iterative process continues, expanding the vectors until they reach a length of 20 entries, and until these 20 seconds, the value of $\widehat{\boldsymbol{Q}}_{\boldsymbol{Cap_2}}$ is held as constant. At this stage, a new $\widehat{\boldsymbol{Q}}_{\boldsymbol{Cap_3}}$ estimation is performed based on fitting a total of 20 data points. The cycle repeats until 1000 data points are collected, and from that point onward, the vectors consistently maintain a length of 1000 entries, shifting forward in time with each new data addition.

Note: In order to validate the quality of SOH estimation, we need to validate on the Capacity checkup points that are available to us from the characterization cycles performed after each group of 50 charge-discharge cycles. We assume that in between these checkup points the true capacity follows a linear trend/linear interpolation. This means that whatever our algorithm yields, the true comparison should be done only on these checkup points, since we do not know the real development of capacity loss between check-up points, which may not be linear.

On validation of the above-mentioned algorithm on the Dataset 3.1, we obtain ***2.3% RMSE on SOH estimation*** while achieving 3.45% RMSE on SOC estimation. Figure 10 shows the comparison between the Capacity checkup points and the estimated capacity for each iteration of checkups. On considering Dataset 3.2 with constant temperature, we obtain ***0.9% RMSE on SOH estimation*** while achieving 3.33% RMSE on SOC estimation. Earlier in the subsection 3.1 we saw the SOC estimation yields between 0.5% to 1%, and in the long datasets we observe a higher RMSE. The explanation behind this increase is that we are comparing the estimated SOC throughout with the Coulomb-counting method. The Coulomb-counting method straight away does not account for errors in current measurements, while DEKF algorithm for SOC corrects for these errors. It can be visualized from Figure 12 that the Coulomb-counting method sometimes yields SOC below 0% and above 100%. So, this RMSE for SOC estimation of the used references is way lower for the validation tests done in subsection 3.1.3 for the short data set, which yields an RMSE of SoC estimation versus the coulomb-counting reference of 0.5 to 1%. This is due to the fact the Coulomb-counting method is performed on the noisy current measurements. This comparison should be ideally done on error-free perfect current measurements. The solution to avoid this issue is to post-process data by trimming the faulty measurements and jumps. For instance, in comparing SOC estimation with constrained coulomb-counting method (that is forcing the SOC to be between 0% and 100%), the RMSE drops from 3.45% to 2.42% for the dataset 3.1. However, the intention of these algorithms is to run it online, such that they can account for these errors and yield SOH estimations. On including the dynamic 1D temperature measurements in the algorithm for the second dataset 3.2, ***the RMSE for SOH estimation was reduced to 0.83%.*** These results are also represented in Table 1.

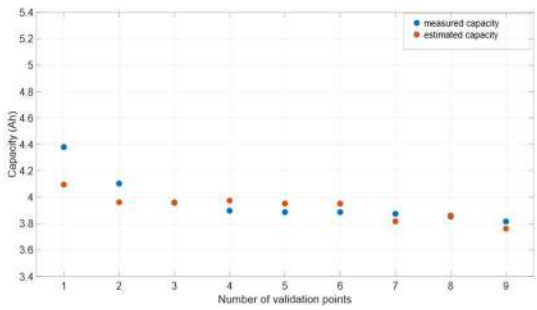

Figure 10 Validation of Capacity Q_{cap} on the Dataset 3.1.

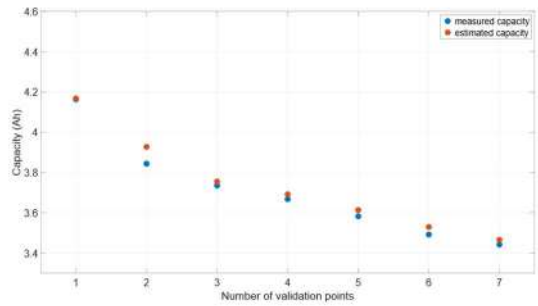

Figure 11 Validation of Capacity Q_{cap} on the Dataset 3.2 with Temperature measurements included (last row in Table 1).

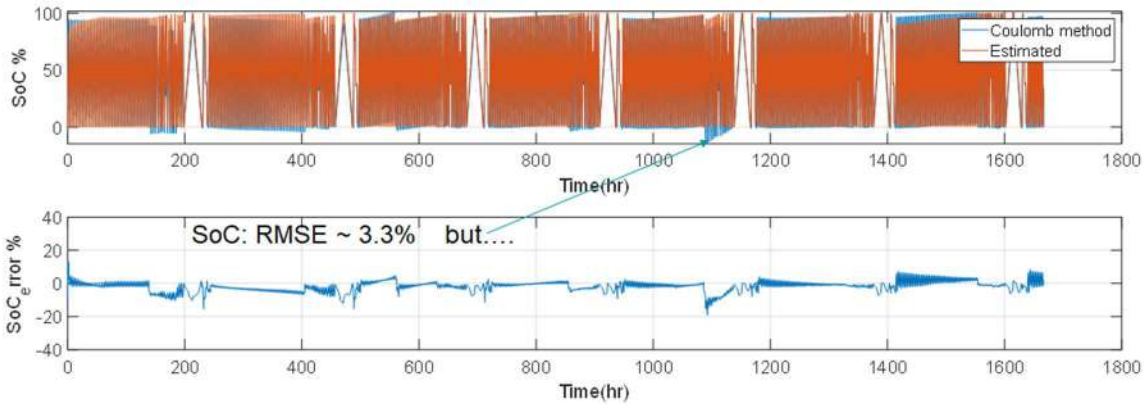

Figure 12 SOC estimation results on the Dataset 3.2 with constant Temperature (second row in Table 1).

The initial validation test, as outlined in Table 1, focused on Dataset 3.1. In this dataset, the EMF function in the model (3-5), as well as the LPV functions, consistently utilized a constant temperature (T = 25°C). However, when dealing with Dataset 3.2, two distinct approaches were considered. The first is the use of a constant temperature of 25°C, similar to Dataset 3.1. The second approach is the inclusion of the measured temperature values directly into the model. Notably, the results in terms of RMSE at the checkup points for Dataset 3.2 were better than in the dataset 1, and there was a slight improvement when measured temperature values were included for the Dataset 3.2. Overall, the SOH estimation results at the checkup points proved satisfactory, suggesting the potential extension of these algorithms to accommodate L1 measurements in Section 4.

Table 1 Quantitative results yielded from SOH estimation algorithm for baseline cells.

Data	Temperature	~RMSE – SoC	~RMSE – SoH (Interpolated)	~RMSE – SoH (checkup points)
Data set 3.1	Constant	3.45%	3.7%	2.3%
Data set 3.2	Constant	3.33%	3.46%	0.9%
Data set 3.2	Temperature	3.1%	3.6%	0.83%

Including 1D temperature measurements

Because these are calculated with reference to coulomb method, (with errors in measurements)

Because this is calculated by interpolating between the checkup points

Note: The SOH estimation algorithm maybe sensitive to the hyperparameters: M, and Δt_{diff}, provided the hyperparameter Δt_{SoC} remains constant, ideally $\Delta t_{SOC} = 1s$. This makes it important for us to perform a sensitivity analysis on these hyperparameters before moving on to the next section 4. This is done in the next subsection.

3.4.3 Sensitivity analysis of SOH estimation w.r.t hyperparameters

Our algorithm incorporates several hyperparameters, and to assess their impact, we conducted a sensitivity analysis through parameter sweeps. Figure 13 and Figure 14 illustrate the RMSE values of SOH estimation as we varied the parameters M and Δt_{diff}, respectively, while keeping other parameters constant. While a 2D parameter sweep is a possibility, our initial results indicate that sweeping individual parameters was sufficient to comprehend the behaviour of our SOH estimation algorithm. This analysis suggests a preference for lower values of Δt_{diff} as seen in Figure 14. This preference arises from the fact that larger values result in less data available for fitting our SOH estimation model. Conversely, larger values of M imply more data considered for fitting, but Figure 13 reveals that the optimal values lie between 500 and 1000. Beyond this range, we observed a gradual increase in RMSE, indicating the importance of finding a balance in parameter selection for optimal SOH estimation performance.

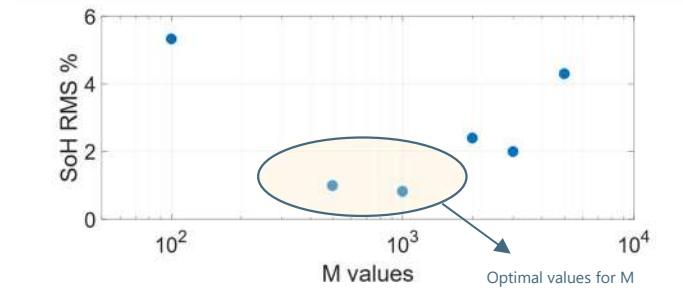

Figure 13 Sensitivity of SOH RMS to different M values

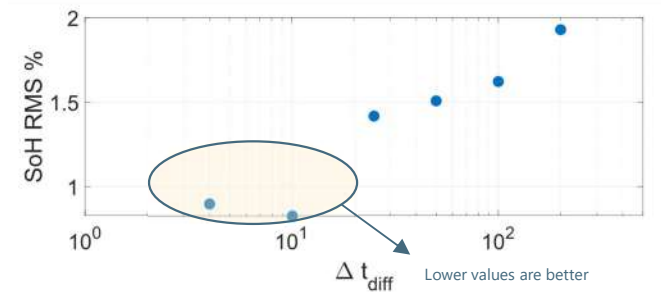

Figure 14 Sensitivity of SOH RMS to different Δt_{diff} values

3.5 SOS

The increased use of lithium-ion battery systems is increasing their safety requirements, as avoiding devastating events such as thermal runaway becomes essential for a wider adaptation of these technologies. In this context, a comprehensive monitoring of the battery state to prevent non-safe and devastating events is necessary to increase the reliability of lithium-ion batteries. Indeed, this is the objective of the State-of-Safety (SOS) algorithm proposed in this section: to provide a numerical quantification of the safety state of a lithium-ion battery. This is contrary to other battery safety-related concepts such as safety standards, which only define pass or fail criteria, or define the safety in a qualitative manner.

In the first approach presented in the current section, the baseline SOS is developed and validated. This baseline version relies on the typical measurements available in state-of-the-art Battery Management Systems (BMS): voltage, current and temperature. On a later stage, in Section 4.3 the algorithm proposed in this section will be improved with the addition of the measurements provided by the L1 sensor technology developed in the SENSIBAT project.

3.5.1 Definition and derivation of SOS formula

The proposed SOS is based on the idea that safety is inversely proportional to the concept of abuse [11]. That is to say, as the battery is subjected to more abuse, its safety state decreases to zero. This leads to the following expression:

$$f_{safety}(x) = \frac{1}{f_{abuse}(x)} \tag{3-11}$$

where *x* represents all types of variables that describe the behaviour of the battery. In the case of the baseline SOS algorithm, this includes voltage, current and temperature.

As previously mentioned, the SOS provides a numerical quantification of the safety state, which ranges as the typical state algorithms (SOC, SOH, SOE or SOP) between 0% (critically unsafe state) and 100% (state guaranteeing safety). Slightly varying the previous expression, the safety function can be comprised between the boundaries 0-1, leading to equation (3-12), where *g(x)* is also a function that can take any value.

$$f_{safety}(x) = \frac{1}{g(x) + 1} \tag{3-12}$$

In the specific case of the SOS algorithm proposed in the SENSIBAT project, *g(x)* is proposed to be a quadratic function, defined by parameters *m* and *d*:

$$f_{safety}(x) = \frac{1}{m[x - d]^2 + 1} \tag{3-13}$$

Additionally, *m* and *d* values can be further derived in order to easily design the shape of the safety function by just the definition of two points. These points are defined as the $f_{safety}(x_{100})$ = 100% and $f_{safety}(x_{80})$ = 80%. 100% is the point that guarantees safety, while 80% is defined as the threshold in which the safety of the battery starts to be an issue. The 80% value is defined to follow the same convention as with the SOH (end of life of typical battery systems is defined at an SOH related to capacity of 80%). The derivation of equation (3-13) according to the definition of points x_{100} and x_{80} leads to equation (3-14). For further information of this derivation, the reader is referred to publication [11].

$$f_{safety}(x) = \frac{1}{0.25 \cdot \left[\frac{x - x_{100}}{x_{80} - x_{100}}\right]^2 + 1} \tag{3-14}$$

Therefore, with equation (3-14), it is possible to derive a safety expression for any x variable, by just defining which are the states of that variable that ensure the safety (x_{100}) and that compromise the safety (x_{80}). As mentioned before, in the case of the baseline algorithm, it will rely on the measurements of voltage, current and temperature. From each of these measurements, it is possible to derive more than one safety expression:

- In the case of the voltage, both high voltages and low voltages may lead to non-safe states. Therefore, a safety function is linked to the high voltages, $f_{safety}(V_{max})$, and another safety function is linked to low voltages, $f_{safety}(V_{min})$.
- Similar to the voltage case, currents may lead to non-safe states both at charge and discharge states. A safety function is linked to charging currents, $f_{safety}(I_{ch})$, and another safety function to discharging currents, $f_{safety}(I_{dc})$.
- Finally, in the case of the temperature, very high temperatures are the highest risk. However, the sole detection of high temperatures may not be valuable in a context of rapid temperature increase (e.g., in a thermal runaway event). This can be partially fixed with the use of a safety expression related to the increase of temperature (or temperature derivative). However, a rapid temperature increase happening in low temperatures may not be as risky as happening in high temperatures. Due to this reason, a hybrid approach is defined: the actual temperature and the temperature derivative are used to estimate the

time left to reach a non-safe temperature (typically, 60ºC in lithium-ion systems). Therefore, $f_{safety}(T)$ is defined as a function linked to the time left to reach 60ºC. Once the temperature is above 60ºC, the value of $f_{safety}(T)$ is kept low, hence there is no need to define a safety function linked directly to the actual temperature value.

Additionally, a last function is also added, which is related to the coherence of the voltage and current measurements. In normal and safe battery operation, while charging the battery, increasing or maintaining the (absolute) current leads to an increased battery voltage. Also, during discharging, increasing or maintaining the (absolute) current leads to a reduction in battery voltage. If these states are not met, the battery may have suffered an internal malfunction, what might lead to a non-safe state. As the coherence is defined as a boolean state (yes or not), in this case another kind of function (f_{fault}) is defined, which just checks the current and voltage derivatives (with a window of 10 seconds). In case there is no coherence between these variables, the fault function is defined at 79%. It is worth noting that, in order to avoid issues with the resolution and precision of the sensors, a tolerance is added to the voltage and current derivatives: changes in voltage or current below that tolerance are not considered to be significant for the fault detection.

Eventually, the SOS is defined as the multiplication of all the safety functions previously defined. The two safety functions related to the voltage (V_{max} and V_{min}) and the current (I_{ch} and I_{dc}) can be gathered in a single expression, as the battery cannot be at the same time at low and high voltage, or at charging and discharging state. Therefore, the SOS expression is defined as in equation (3-15):

$$\boldsymbol{SOS = f_{safety}(V) \cdot f_{safety}(I) \cdot f_{safety}(T) \cdot f_{fault}} \quad \textbf{(3-15)}$$

Since the SOS is a multiplication of the different safety terms, it is possible to define the safe and non-safe SOS thresholds, which may be useful to define alarms:

- $SOS = 100\%$ means that all the individual terms are at 100%. Until $SOS = 80\%$ it may be safe to use the battery.
- $SOS < 80\%$ means that most of the terms are close to their safe limit (80%), or that one of the terms is below that threshold. In the first case, a warning is generated. In the second case, it is considered that the battery is already at a non-safe state. Each time an SOS term falls below 80%, an alarm related to that term is generated.
- $SOS < 41\%$ (0.8^4) means that all the terms (4 terms) might be below their safe limit, or that one of the terms is very low. This is defined as a completely non-safe state.

The following figure depicts a graphical representation that sums up the concept proposed for the SOS.

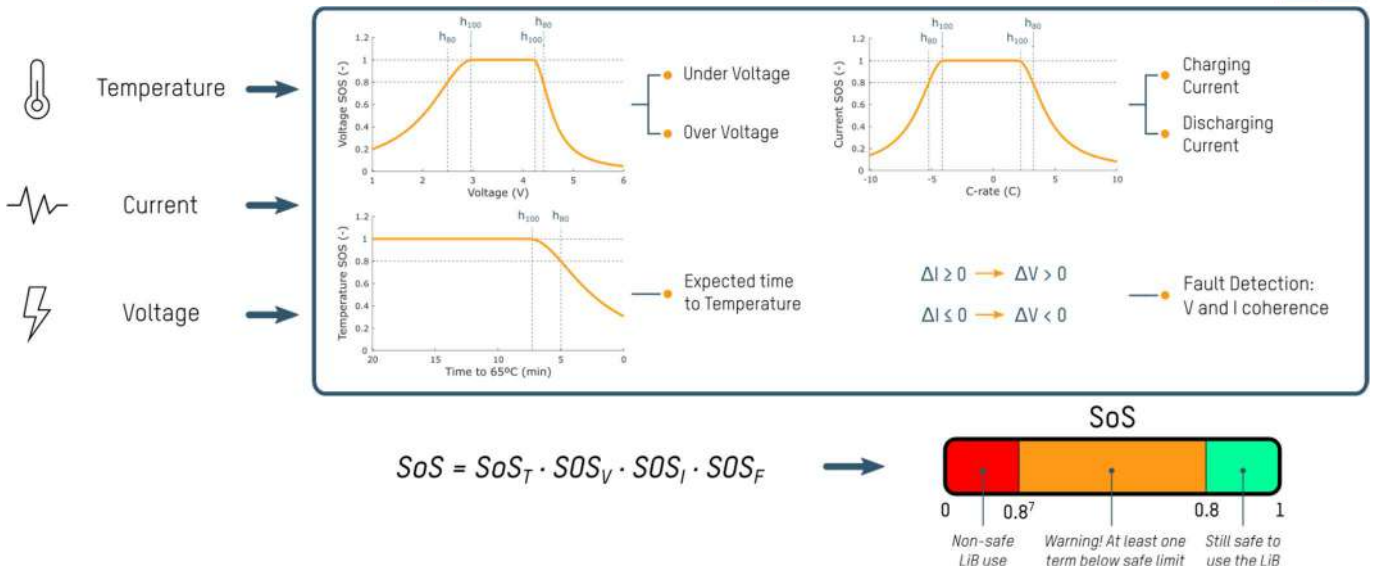

Figure 15: Representation of baseline State of Safety (SOS).

3.5.2 Parametrization of SOS algorithm

As explained in the previous subsection, in order to parametrize the SOS algorithm, it is necessary to define the values of the x_{80} and x_{100} components of equation (3-14), for each of the safety functions composing the SOS overall expression (except the fault expression): $f_{safety}(I_{ch})$, $f_{safety}(I_{dch})$, $f_{safety}(V_{max})$, $f_{safety}(V_{min})$, $f_{safety}(T)$.

The defined x_{80} and x_{100} parameters are depicted in Table 2. The parameters related to the voltage and current have been defined based on the requirements of SENSIBAT cells, which can be found in Deliverable 1.1 [1]. Additionally, the parameters related to the temperature (time to reach 60ºC) have been defined with the objective of having enough room to prevent catastrophic temperature increases. The convenience of these values is tested in the next subsection, with the validation of the SOS algorithm.

SOS Term	***Parameter***	***Value***
Voltage (High)	x_{100}	4.2 V
	x_{80}	4.4 V
Voltage (Low)	x_{100}	3 V
	x_{80}	2.5 V
Current (Charge)	x_{100}	2 C
	x_{80}	3.2 C
Current (Discharge)	x_{100}	4 C
	x_{80}	5.2 C
Time to 60ºC	x_{100}	7.5 minutes
	x_{80}	5 minutes

Table 2. Parametrization of Baseline SOS algorithm.

3.5.3 Validation of SOS algorithm

As specified in section 2, for the validation of the baseline SOS algorithm, data from the safety tests performed during the SENSIBAT project is used, which are denoted as Dataset 2. It is considered that the best framework to validate the usefulness of the SOS algorithm is the one composed by these tests, as they reflect conditions outside the recommended bounds, which are precisely the ones to be detected by the SOS algorithm. As already mentioned, the safety tests consist of three routines: nail penetration test, heating test and overcharge test [4].

From the performed tests, heating and nail penetration are considered not to be representative to evaluate the SOS algorithm. Indeed, during the heating test the cell is forced to increase its temperature by gradually increasing the ambient temperature (up to 200ºC, when thermal runaway happens). This rise is performed slowly, and therefore the algorithm would just detect that the cell is arriving to the threshold of 60ºC. Regarding the nail penetration test, it is performed while the battery is in rest. Therefore, the SOS algorithm would not detect the fault (fault detection is only working while the battery is in operation, as voltage variations are common at rest).

Hence, the SOS algorithm will be validated with the results of the overcharge tests. Overcharge tests were carried out to different cells following this procedure: charge at 2C (2A) until reaching 24V, or a safety-relevant event occurs. The tests were performed only to one baseline cell and two L2 cells, due to the complications with L1 cells (see D5.1 for further information regarding this issue [4]). Two representative cases have been selected: baseline cell (DUT8) and L2 cell #2 (DUT14). In the first case the thermal runaway event happens when the cell has reached 24V and the temperature is above 60ºC. In the second case, however, the thermal runaway event happens before reaching the 60ºC threshold, and the voltage is around 7V. Note that in the case of the L2 cell, even if the data from the novel sensor technology was also recorded, it is not used for the validation of the proposed baseline SOS algorithm (only typical voltage, current and temperature measurements are used). Being aware of the available data in the framework of the SENSIBAT project, it is considered that these two tests are enough to evaluate the SOS algorithm.

It is also worth to mention that, contrary to previous algorithms, it is not possible to compare the SOS estimated by the algorithm with a close-to-real SOS value, as there is not a physical magnitude that measures the “safety”. Due to this reason, the validation of the SOS algorithm is focused on analyzing the capacity of the algorithm to prevent catastrophic events (such as a thermal runaway).

3.5.3.1 Overcharge Test: Baseline Cell

Figure 16: Data recorded in overcharge test (Baseline Cell): current, voltage and temperature.Figure 16 shows the data recorded during the overcharge test of the baseline cell: current, voltage and temperature measurements. Note that a first phase of 120 seconds (with the cell in rest) has been added to the data to let the algorithm be properly initialized. The orange-shadowed part represents the period where the thermal runaway is happening (cell catches fire).

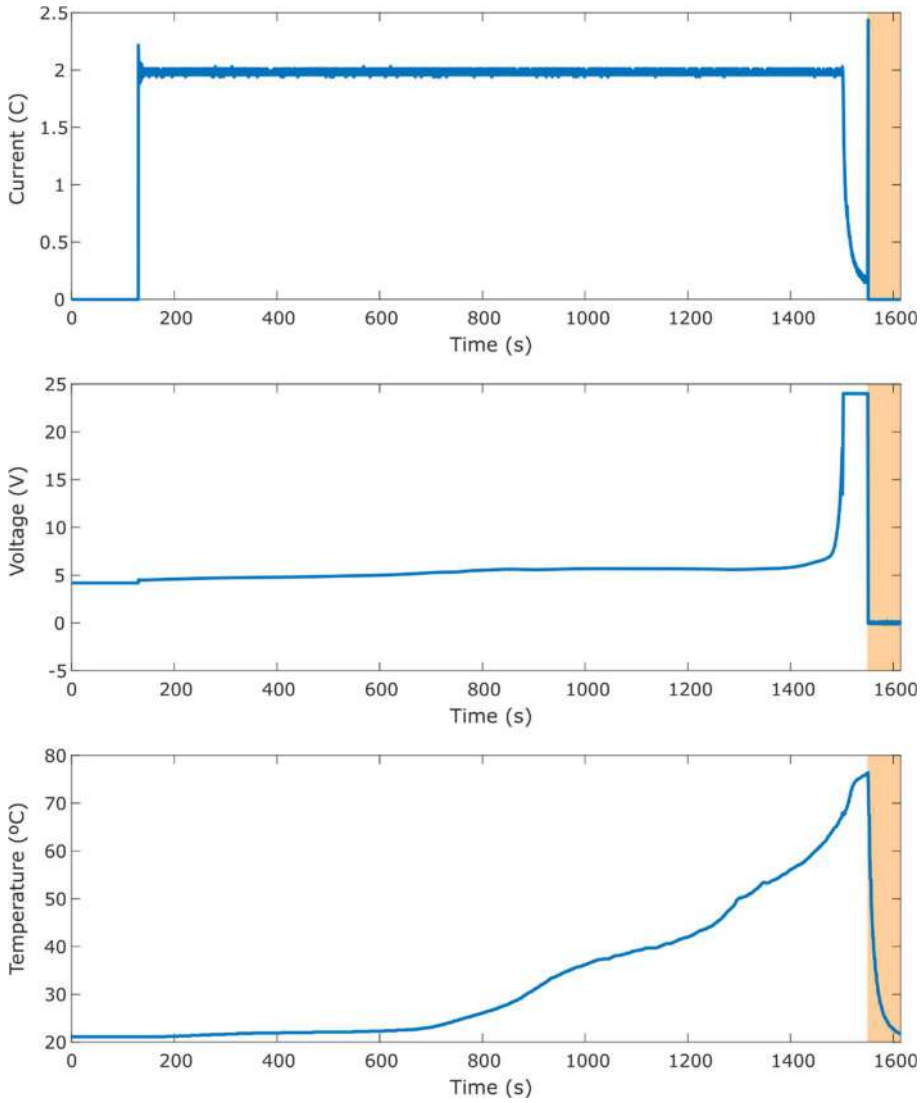

Figure 16: Data recorded in overcharge test (Baseline Cell): current, voltage and temperature. Orange shadowed part represents thermal runaway event.

As it can be seen, the cell already starts at its maximum voltage (4.2V). However, it was possible to continue charging the cell at 2C for more than 20 minutes (1270s) before arriving to 24V. After arriving to 24V, a CV charge was applied, and the thermal runaway event started at nearly the end of this phase. Before the thermal runaway started, the cell was approximately at 76ºC.

Figure 17 shows the different outputs of the SOS algorithm when executing it with the data previously presented. The first graph shows the overall SOS value, i.e., the output of equation (3-15). Besides, the second graph shows the decomposition of the overall SOS into its different terms, i.e., each of the safety functions aggregated in equation (3-15): temperature, current, voltage and fault. Finally, the last two figures show the warning level (completely unsafe, unsafe, warning and safe operation, according to the defined SOS thresholds) and the triggered alarms (i.e., which SOS term is below 80%).

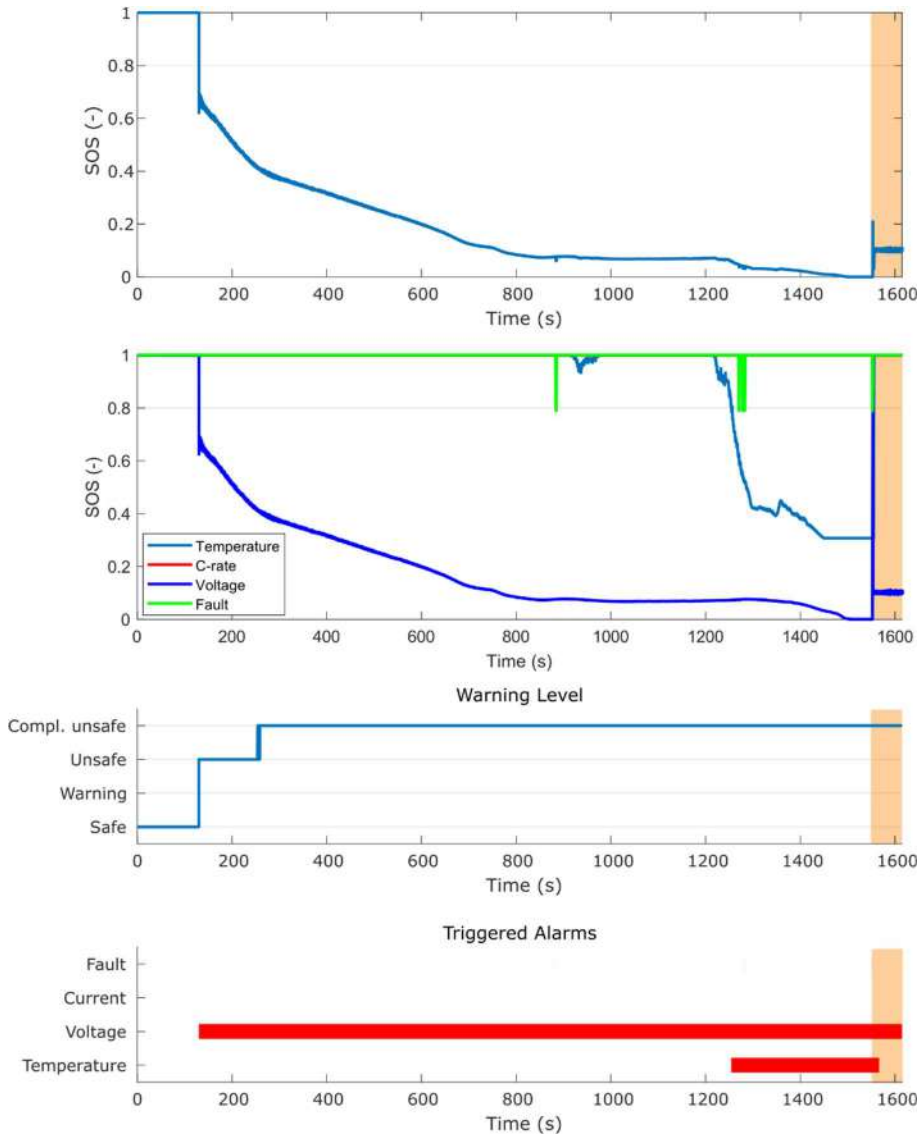

Figure 17: Output of SOS algorithm for overcharge test (Baseline Cell): Overall SOS, SOS by terms, warning level and triggered alarms. Orange shadowed part represents thermal runaway event.

The first conclusion is that the SOS triggers an alarm practically at the very beginning of the overcharge test. Considering that the cell starts at a completely charged state (4.2V) and that rapidly crosses the x_{80} threshold (4.4V), this is a logical conclusion. From the very beginning of the test, the SOS is already at 60%, solely due to the voltage safety function, which triggers an alarm. After approximately 200 seconds, the SOS falls below 41%, leading to a completely unsafe state. At this point the only triggered alarm is related to the voltage.

Anyways, from this point and until the thermal runaway event happens, other two alarms are triggered: one related to the fault detection, and another one related to the temperature. Even if the SOS is already warning about a non-safe state due to the voltage value, the other two alarms are also an important point to be highlighted, as they demonstrate the redundancy of the proposed SOS algorithm. For instance, in a situation when the voltage sensor is providing a wrong value (or is not even providing a value), the BMS is not able to detect the overcharge event. In that case, the SOS would be able to detect that something is happening in the cell due to the temperature-related SOS term. In other words, it is demonstrated that dividing the SOS into different terms (which are also somehow inter-related) gives the algorithm redundancy, as if one of the sensor technologies suffers from a fault, the other sensors are able to give valuable information about a non-safe state.

In the next paragraphs, the detection of the fault-related and temperature-related non-safe states are further investigated. On the one hand, in the case of the fault detection part of the algorithm, it is triggered around t=885s, and again around t=1300s. As an example, Figure 18 depicts the part of the experiment when the first fault detection alarm is triggered. Specifically, the three graphs show the SOS (divided in the four terms composing it), current and voltage measurements between t=860s and t=890s.

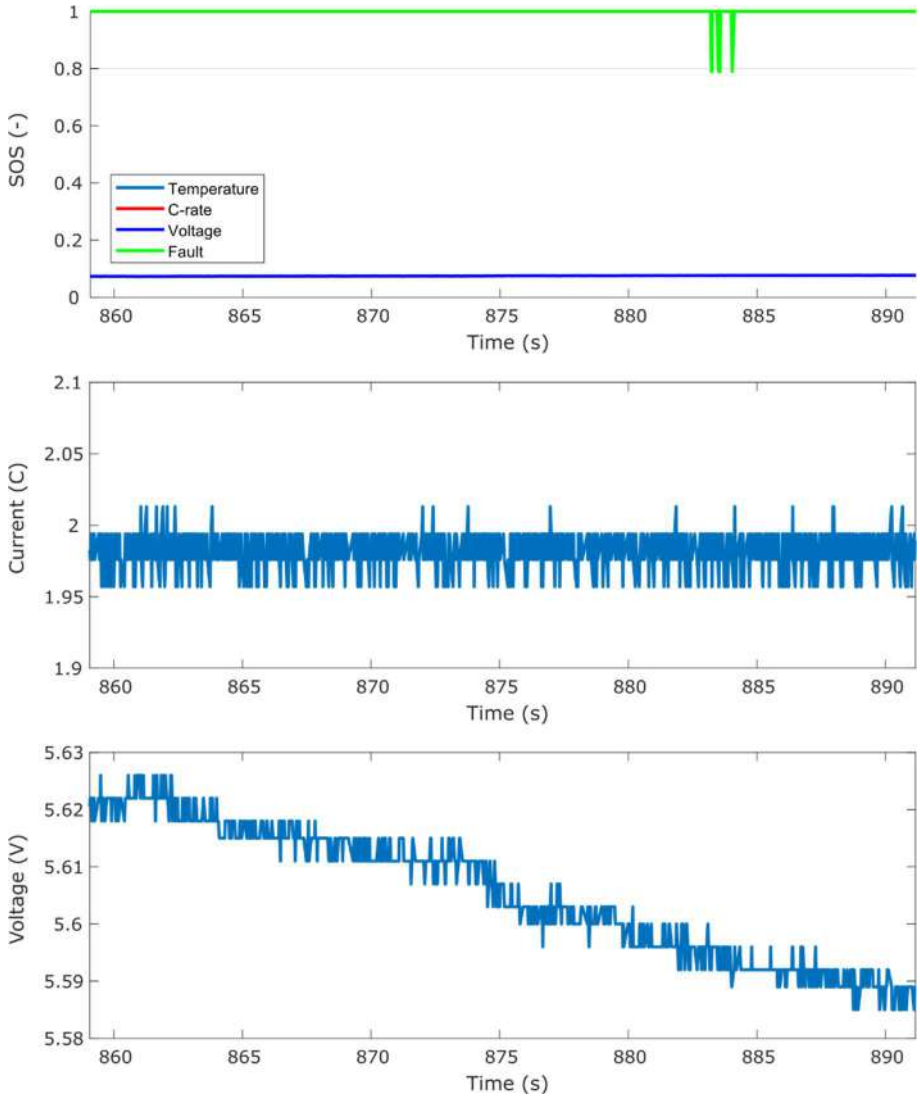

Figure 18: Zoom at first triggering of fault detection: SOS, current and voltage.

The graphs show that during this period, even if the battery is still being charged at a (practically) constant rate, the voltage is slowly going down. This is a behaviour that it is not seen during normal operation, but that can be noticed at overcharge or overdischarge states. Even if the graph shows a clear downwards trend in the voltage, the algorithm is only capable of detecting the fault at some short instants (when the fault-related SOS goes below 80%). This is due to the resolution and precision of both the current and voltage sensors, which show a clear noise. As mentioned in Section 3.5.1, in order to avoid issues with the resolution and precision, a tolerance is added to the calculation of the voltage and current derivatives. In the case of these measurements, tolerances of 0.02V and 0.1A have been defined, respectively. Therefore, the voltage drop in the previous 10 seconds must be higher than 0.02V for the SOS to be able to detect the fault state. This demonstrates the need of sensors with good resolution and precision for the correct performance of the SOS algorithm.

On the other hand, in the case of the temperature-related part of the SOS algorithm, it triggers an alarm around t=1250s, which is maintained until the thermal runaway event happens. Figure 19 depicts the part of the experiment when the temperature-related alarm is triggered. Specifically, the two graphs show the SOS (divided in the four terms composing it) and the temperature measurements between t=1200s and t=1600s.

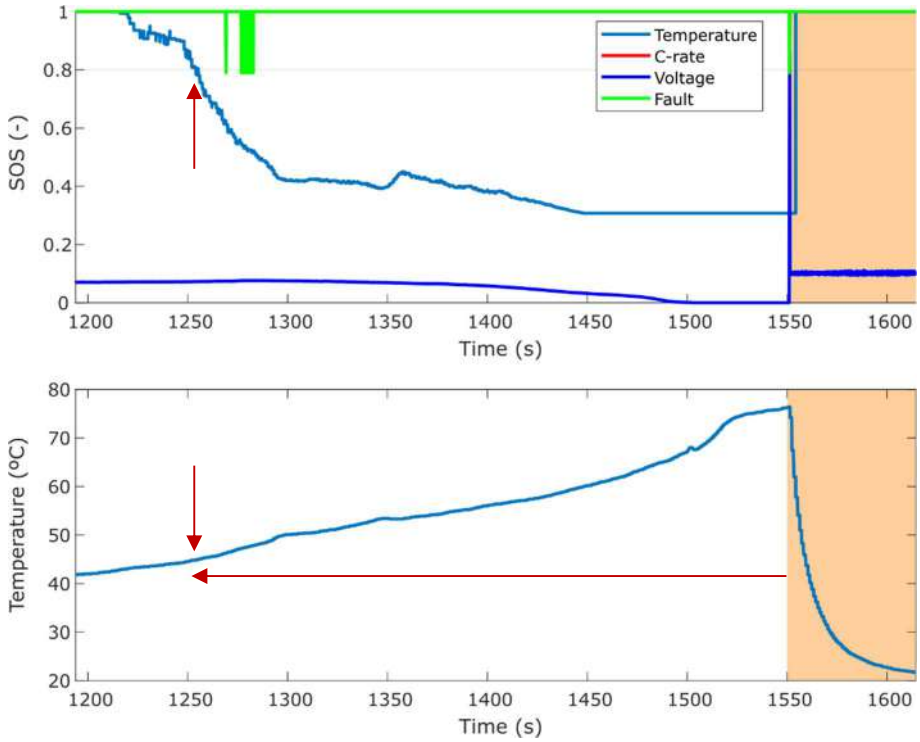

Figure 19: Zoom at period of time where temperature-related SOS is triggered: SOS and temperature.

These results demonstrate that the SOS is able to detect an anomaly in the temperature evolution 300 seconds (5 minutes) before the thermal runaway event happens. From this point on, the temperature needs less than 5 minutes to reach the 60ºC, but this is a normal conclusion, as the SOS predicts the temperature evolution only with the past data (based on the derivative of temperature in the last 60 seconds). Therefore, as the temperature starts to increase faster after the point the alarm is triggered, it reaches faster than expected to the 60ºC threshold. In any case, the alarm is triggered with enough time to take preventive measures. It is also worth to note that when the alarm is triggered, the battery temperature is still at 45ºC.

3.5.3.2 Overcharge Test: L2 Cell #2

Figure 16: Data recorded in overcharge test (Baseline Cell): current, voltage and temperature.Figure 20 shows the data recorded during the overcharge test to one of the L2 cells: current, voltage and temperature measurements (only baseline measurements). A zoom is made to the temperature in order to see with more detail the temperature evolution just before the thermal runaway happens. Note that a first phase of 120 seconds (with the cell in rest) has been added to the data to let the algorithm be properly initialized. The orange-shadowed part represents the period where the thermal runaway is happening (cell catches fire).

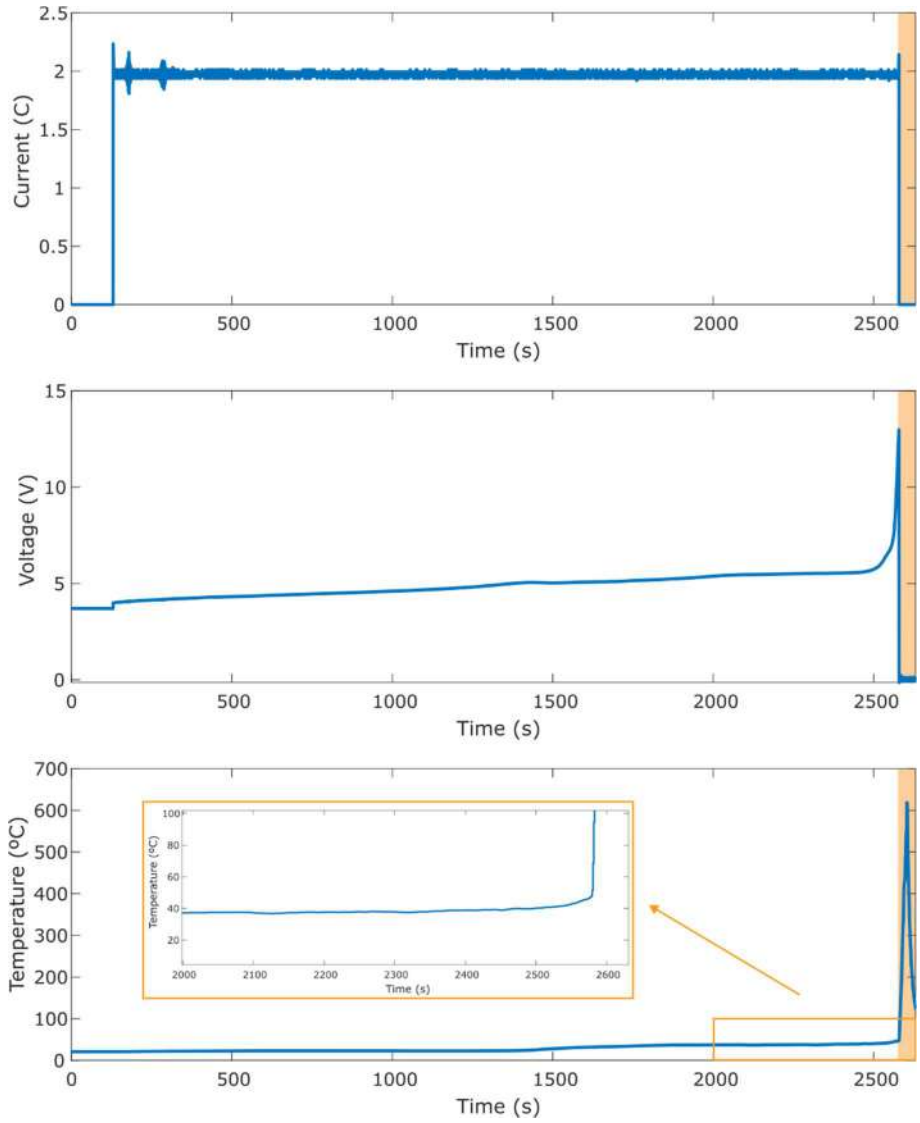

Figure 20: Data recorded in overcharge test (L2 Cell #2): current, voltage and temperature. Orange shadowed part represents thermal runaway event.

Compared to the previous test, in this case the cell starts in a lower charge condition (3.7V). Due to this reason, it requires some minutes to reach the full charge (4.2V, around t=325s). From that point on, the cell continues being charged at 2C for more than 2200 seconds. During all this time, both the voltage and temperature do not increase much (around t=2500s, the cell is still at 5.7V and 40ºC). Then, suddenly the cell voltage starts to increase fast, and eventually the thermal runaway event happens. Therefore, the main differences with the previous test are that: 1) the cell is overcharged for a longer time, 2) thermal runaway event happens more unexpectedly, and when the cell temperature is still below 60ºC. Therefore, it can be considered that this context is more challenging for the SOS, as the cell state changes faster to a completely non-safe state.

Figure 21 shows the different outcomes of the SOS algorithm when executing it with the data previously presented. As in the previous test, the first graph presents the overall SOS value, the second graph the decomposition of the overall SOS into its different terms, the third graph the warning level, and the final graph the triggered alarms.

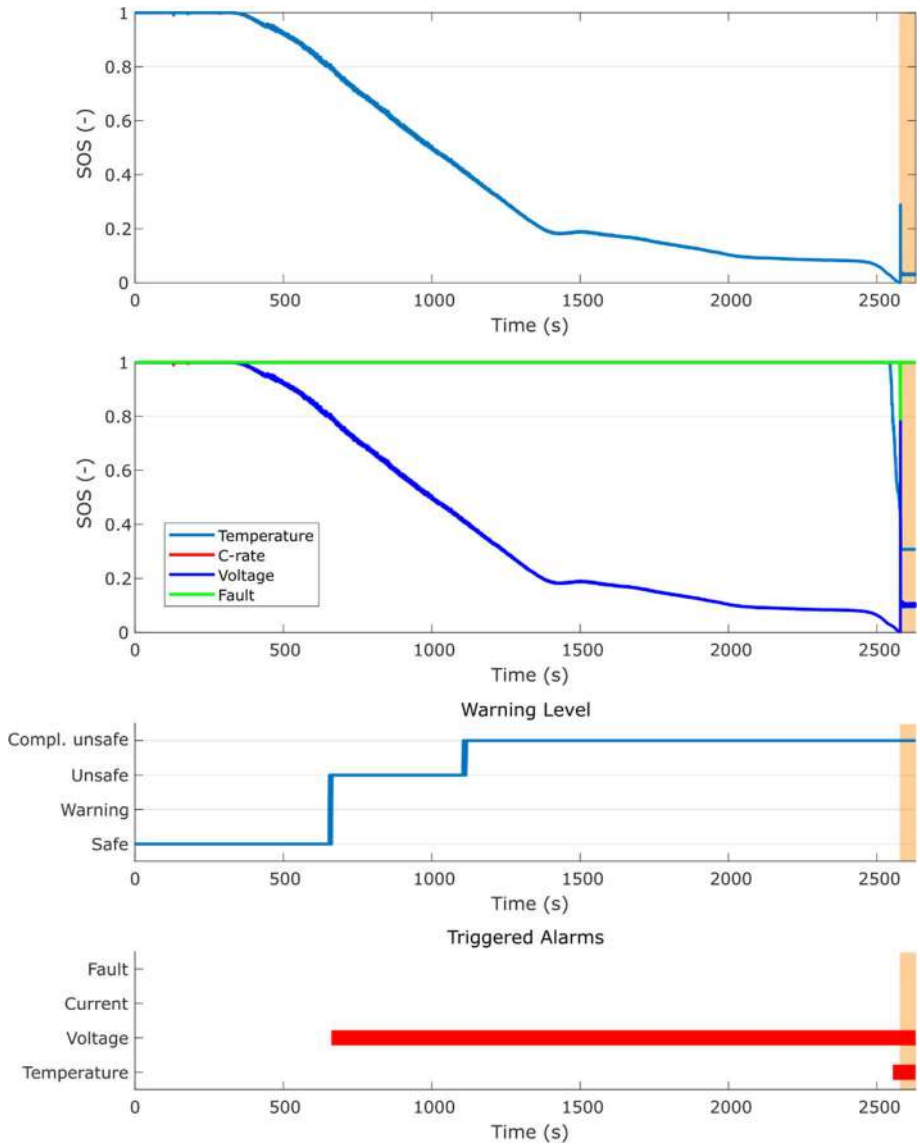

Figure 21: Output of SOS algorithm for overcharge test (L2 Cell #2): Overall SOS, SOS by terms, warning level and triggered alarms. Orange shadowed part represents thermal runaway event.

Being an overcharge test, as in the previous case, the overall SOS is mostly defined by the voltage-related safety function. As previously mentioned, the cell starts below the maximum charge and requires more time to reach the overcharge state. This is the reason why the voltage-related SOS drops slowly, especially comparing with the previous overcharge test. In any case, the SOS algorithm already triggers an alarm around t=700s, when the cell crosses the 4.4V threshold. Around t=1100s, the SOS drops below 41%, entering in the completely non-safe operation state. As it can be seen in the graphs, all this happens much before the thermal runaway event starts (which happens around t=2600s).

In any case, and as highlighted with the previous overcharge test data, it is also important to analyze the activation of the other alarms. Indeed, they demonstrate the robustness and redundancy of the algorithm against a variety of non-safe situations (not only the overcharge state available in the framework of the SENSIBAT project). In the case of this test, the fault detection alarm is only triggered in the very same instant that the thermal runaway event happens, therefore it is not relevant.

Regarding the temperature-related alarm, as previously mentioned, the thermal runaway event is triggered when the cell is still below 60ºC. This complicates the detection of a non-safe temperature related state. As seen

in Figure 21, the temperature-related alarm is triggered close to the thermal runaway event. Figure 22 shows a zoom of the last 100 seconds before the thermal runaway event happens. Specifically, the two graphs show the SOS (divided in the four terms composing it) and the temperature measurements between t=2480s and t=2620s. In the temperature graph, a zoom into de Y-axis is also included, in order to better identify the temperature evolution in the very last seconds before the thermal runaway happens.

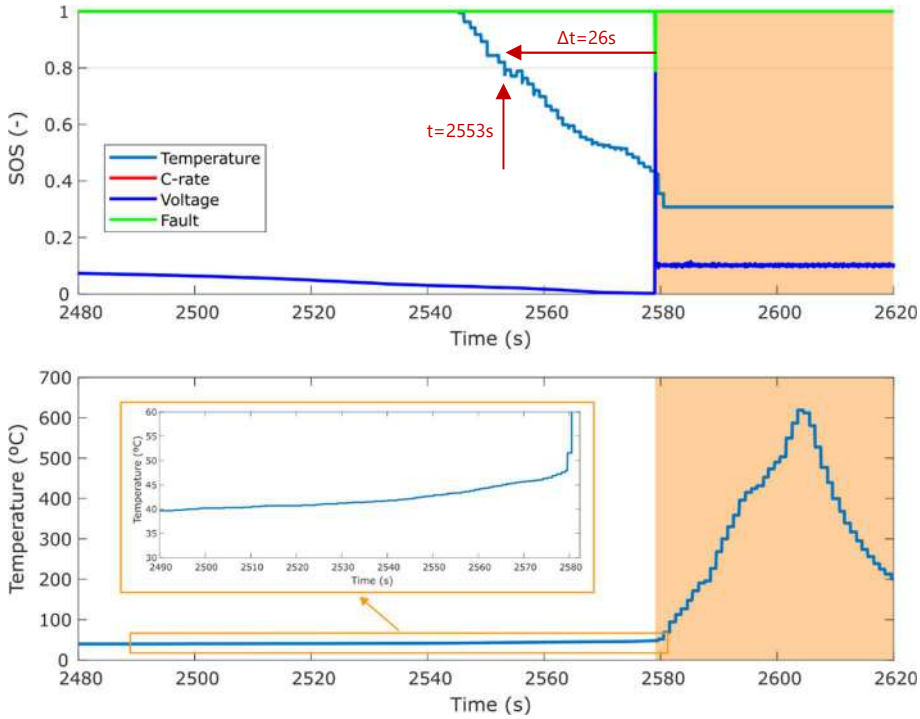

Figure 22: Zoom at period of time where temperature-related SOS is triggered: SOS and temperature graphs.

As the figures show, in the last 100 seconds before the thermal runaway happens, there is a slow increase of the temperature, which is still below 40ºC at t=2480s. Exactly at t=2553s, the SOS detects a faster increase of the temperature, and triggers an alarm. At that point, the cell temperature is still at 43ºC, far from the typical 60ºC threshold. Then, 23s after triggering the alarm, the thermal runaway event happens. As recorded by the laboratory equipment, during the thermal runaway temperatures above 600ºC can be found.

This demonstrates that, even in a sudden thermal runaway event, the SOS is able to foresee this event with enough room (26s in this case) to take preventive measures, even if the temperature increase before the thermal runaway happens is not very pronounced.

4 L1 algorithms

The state estimation algorithms in the present section make use of data from the battery built-in novel sensor technology developed in this project (Level-1, or L1 sensor technology, including 2-dimensional (2D) temperature and mechanical pressure inside the pouch cell). Specifically, improved SOC, SOH and SOS algorithms are presented. In the following subsections, each of the algorithms is presented in detail, and the improvements that the L1 sensor technology are able to provide are evaluated.

4.1 SOC

For the L1-SOC algorithm, we still use the **Filter-based method**. As seen in the section 3, the first step to use this method is to obtain a model. In order to obtain the model, we decided on extending the modelling approach to incorporate 2D temperature and pressure measurement data and to incorporate the 2D data to improve the SOH estimation. We start with 1RC ECM state-space model from section 3 that includes SOC_k and the overpotential o_k at the time instant k as states, which is shown below in (4-1) (repeat from (3-3)), with θ^1, θ^2 and θ^3 representing $A(SoC_k, T_k)$, $B(SoC_k, T_k)$ and $D(SoC_k, T_k)$.

$$\begin{bmatrix} SOC_{k+1} \\ o_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \theta^1 \end{bmatrix} \begin{bmatrix} SOC_k \\ o_k \end{bmatrix} + \begin{bmatrix} \frac{\Delta t_{SOC}}{Q_{cap_{nom}}} \\ \theta^2 \end{bmatrix} u_k \qquad (4\text{-}1)$$
$$y_k = EMF(SOC_k, T_k) + o_k + \theta^3 u_k$$

4.1.1 Model for SOC estimation of cell with L1 sensor

We first start with 2D temperature measurement data and then later on extend this with 2D mechanical pressure data.

4.1.1.1 Incorporating 2D temperature measurement data

There are several ways of incorporating 2D measurements in the state estimation algorithms [12]. We divide them in the following three ways using 2D temperature measurements $T^{(x,y)}$, where x and y represents the spatial coordinate on the sensor matrix, such that there are $n_x \times n_y$ temperature measurements:

1. ***2D thermal modelling of the battery***
2. ***1RC Model dependency with 2D measurements***
3. ***Averaging measurements.***

We subdivide 2D thermal modelling of the battery into three methods based on the literature: a) ***Electrochemical-Thermal (ECT)-modelling-based method b) Thermal modelling with ECM c) Numerical-Analysis-based method***. Out of the three subdivisions, *the* ***ECT-based modelling method*** is the most accurate method and there are several high-fidelity models available in the literature [13] [14]. However, this modelling approach is often used for battery design purposes and is not suitable for onboard BMS hardware [12]. ***Thermal modelling with ECM,*** such as nRC ECM with multi-mode heat generation models, offers good accuracy and less complexity than high-fidelity models, however, incorporating multi-mode heat generation models within a nRC thermal framework can increase the complexity of the state estimation process [12]. ***Numerical-Analysis-based methods*** such as the Finite-Element Method (FEM) offers accuracy and are preferred methods in battery design, they too have intense computations and sensitivity to parameters such as boundary conditions. Eventually these methods are often avoided in cases where there is a requirement to implement models on real-time state estimators with fast sampling times.

On the other hand, using ***1RC model dependency*** to incorporate 2D temperature measurements seems like a sensible choice, due to the fact that these dependencies are merely mathematical functions, and it provides more freedom in terms of computational complexity and accuracy. In order to do so, we use the model as shown in (4-1) such that the output has the dependencies on 2D temperature.

$$y_k = EMF\left(SOC_k, T_k^{1,1}, \dots, T_k^{n_x,n_y}\right) + o_k + \theta^3 u_k \qquad (4\text{-}2)$$

Moreover, θ^1, θ^2 and θ^3 in the model are the LPV matrices/functions with dependency on SOC and 2D temperature measurements such that they represent $A(SoC_k, T_k^{1,1}, \dots, T_k^{n_x,n_y})$, $B\left(SoC_k, T_k^{1,1}, \dots, T_k^{n_x,n_y}\right)$ and $D(SoC_k, T_k^{1,1}, \dots, T_k^{n_x,n_y})$, respectively. Of course, this method is the correct way to move forward and on receiving 2D temperature measurements data, this method must be used to incorporate the 2D data. In order to streamline our approach given time constraints and a preference for simplicity with lower computational overhead in state estimation, we opt for an alternative and simpler method. This decision is guided by the desire to minimize complexity, reduce the computational load associated with state estimation, and avoid the need for re-identification of the model used in the DEKF algorithm.

The suggested alternative and simpler method consists of a simple step, i.e., averaging the 2D temperature measurements and use the average temperature from the 2D readings such that the output is the following:

$$y_k = EMF\left(SOC_k, T_k^{avg}\right) + o_k + \theta^3 u_k\,, \;\; | \;\; T_k^{avg} = \frac{1}{n_x n_y}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} T_k^{i,j} \qquad (4\text{-}3)$$

In terms of the parameters, LPV matrices/functions θ^1, θ^2 and θ^3 represent $A(SoC_k, T_k^{avg})$, $B\left(SoC_k, T_k^{avg}\right)$, and $D(SoC_k, T_k^{avg})$, respectively. Therefore, the existing baseline model cab be used and no re-identification of the model used in the DEKF algorithm is needed.

4.1.1.2 Incorporating 2D mechanical pressure data

The current state of research lacks extensive literature and a well-established methodology for utilizing 2D mechanical pressure data in SOC estimation. In prior studies, such as those referenced in [15], [16], and the references within, researchers have explored the connection between mechanical stress and SOC by leveraging the known phenomenon of electrode expansion during lithium intercalation. The primary focus in these studies has been on measuring stress changes, indicative of electrode thickness changes, to establish the stress-SOC relationship [17]. A similar approach can be adopted to determine the mechanical pressure-SOC relationship.

To proceed with integrating this relationship into our model and, ultimately, an SOC estimation methodology, we propose two distinct approaches similar to the treatment of 2D temperature measurements. This involves utilizing 2D mechanical pressure measurements denoted as $P^{(x,y)}$, where x and y represents the spatial coordinate on the sensor matrix, such that there are $n_x \times n_y$ pressure measurements:

1. ***1RC Model dependency with 2D measurements***

This approach involves incorporating 2D pressure measurements by utilizing the 1RC model dependency. Leveraging 1RC model dependencies for 2D pressure measurements is advantageous due to their nature as mathematical functions, providing flexibility in terms of computational complexity and accuracy. The model, as expressed in equations (4-1) and (4-3) with averaged temperature measurements, integrates the pressure-SOC relationship into the LPV matrices/functions θ^1, θ^2 and θ^3. These matrices/functions exhibit dependency on SOC, averaged 2D temperature measurements, and 2D mechanical pressure measurements, representing $A(SoC_k, T_k^{avg}, P_k^{1,1}, \dots, P_k^{n_x,n_y})$, $B\left(SoC_k, T_k^{avg}, P_k^{1,1}, \dots, P_k^{n_x,n_y}\right)$ and $D(SoC_k, T_k^{1,1}, P_k^{1,1}, \dots, P_k^{n_x,n_y})$, respectively. Of

course, as with 2D temperature measurements, this method would require fitting LPV functions for 2D pressure measurements which in return increases the computational complexity drastically. This motivates us to move towards the next method, i.e., using averaged measurements.

2. ***Averaging measurements.***

The second method involves a straightforward step of averaging the 2D pressure measurements and using the averaged pressure in the model. The LPV matrices/functions θ^1, θ^2 and θ^3 in this case represent $A(SoC_k, T_k^{avg}, P_k^{avg})$, $B(SoC_k, T_k^{avg}, P_k^{avg})$, and $D(SoC_k, T_k^{avg}, P_k^{avg})$, respectively where P_k^{avg} is calculated as the average of all individual pressure measurements: $P_k^{avg} = \frac{1}{n_x n_y} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} P_k^{i,j}$.

Both methods offer distinct advantages, and the choice between them may depend on the specific requirements of the application, considering factors such as computational efficiency and the level of detail needed in the SOC estimation model. Given time constraints, simplicity, reduced computational load for state estimation, and the desire to avoid re-identifying the model used in the DEKF algorithm, we have opted for the latter method, where the average mechanical pressure is used as scheduling variable in the LPV dependence of the model matrices A, B and D.

4.1.2 SOC estimation of cell with L1 sensor

We propose the same DEKF algorithm in the subsection 3.4 which is also shown in the Annex C. The model selected for SOC estimation uses the method of averaging 2D temperature measurement data. The methods used to determine SOE and SOP are the same as used in the subsections 3.2 and 3.3 respectively. We extend the SOH algorithm as mentioned in subsection 3.4.1 in the subsection 4.2.

4.2 SOH

In the previous section, we have introduced the incorporation of 2D measurement data of temperature and mechanical pressure for SOC estimation. Now, we will show how the 2D measurements can be incorporated for SOH estimation.

In the research highlighted by [14], a compelling motivation is presented for the utilization of mechanical stress as a pivotal parameter in estimating SOH. The study establishes that mechanical stress serves as an effective tool not only for monitoring both SOH and SOC but also exhibits a significant linear relationship with the battery's SOH. Noteworthy is the finding that this linear stress-SOH relationship holds consistently across a diverse range of cycling conditions, enhancing its applicability and reliability. The data presented in the study also suggests that the growth of the Solid Electrolyte Interphase (SEI) is a crucial factor influencing the observed stress-SOH relationship, contributing valuable insights into the underlying mechanisms. Building upon this foundation, the researchers propose a phenomenological model, providing a comprehensive framework to explain the linear stress-SOH relationship. Given these promising findings regarding stress as an effective indicator of SOH, the application of similar relationships to mechanical pressure data measurements emerges as a promising avenue for the estimation of SOH.

Note: While awaiting measurements from the L1 sensor within the battery module, we currently possess valuable pressure data from an external sensor. This data corresponds to the Dataset 3.2 presented in Section 2. Our proposed methodology relies on utilizing this available data. With the pressure measurements in hand, we aim to show SOH estimation algorithms by establishing a comprehensive understanding of the pressure-vs.-SOH relationship. This strategic approach enables us to proactively design and optimize the SOH estimation algorithm before the arrival of data from the L1 sensor of the

battery module. To introduce these algorithms effectively, we will commence with a thorough analysis of the data, showcasing how the relationship between pressure and SOH evolves over time as the cell undergoes degradation and if this relationship aligns with the stress-SOH relationship as depicted in [17]**.**

4.2.1 SOH estimation algorithm

The proposed algorithm for incorporating 2D measurement data from the L1 sensor, follows the exact same steps as proposed in the subsection 3.4.1 for baseline measurements as seen in Figure 9. These steps are briefly mentioned below:

1. SOC estimation using DEKF, retaining LPV function θ_1 and estimating θ_2 and θ_3;
2. Stack past SOC and past current data to obtain a comprehensive dataset;
3. Estimate $\hat{Q}_{cap}$ by finding the optimal estimate by solving a numerical optimization problem as formulated in (3-10) using constrained WLS regression based on the dataset of step 2;
4. Iteratively estimate $\hat{Q}_{cap}$ and eventually estimate SOH using (3-4) until EOL.

Given the expectation to establish a relationship between mechanical pressure and SOH, a functional relation between capacity (Q_{L1}), baseline measurements (V, I), estimated SOC and 2D measurement data ($P^{1,1}, \dots, P^{n_x,n_y}$ and $T^{1,1}, \dots, T^{n_x,n_y}$) is proposed as below :

$$\boldsymbol{Q_{L1} = f(P^{1,1}, \dots, P^{n_x,n_y}, V, I, SOC, T^{1,1}, \dots, T^{n_x,n_y})} \qquad (4\text{-}4)$$

For the averaged measurements, this relationship is simplified to:

$$\boldsymbol{Q_{L1} = f(P^{avg}, V, I, SOC, T^{avg})} \qquad (4\text{-}5)$$

This relationship function, determined offline, can be incorporated into the SOH estimation algorithm. The optimization formulation is adjusted accordingly, introducing a new objective function ($\boldsymbol{\mathcal{J}_k}$) that balances the WLS term used in the baseline algorithm (with weight λ_1) with the deviation between the predicted (via the WLS route) and modelled (obtained offline as in (4-5)) relationship (with weight λ_2):

$$\boldsymbol{\mathcal{J}_k = \lambda_1 \chi_{WLS}\left(\hat{Q}_{Cap_k}\right) + \lambda_2 \parallel Q_{L1_k} - \hat{Q}_{Cap_k} \parallel_2} \qquad (4\text{-}6)$$

The new modified optimization formulation to be used in the step 3 of our SOH estimation algorithm, which then becomes:

$$\begin{gathered} \boldsymbol{\arg\min_{\hat{Q}_{Cap_k}} \ \mathcal{J}_k} \\ \boldsymbol{\text{subject to: } Q_{min} \leq \hat{Q}_{Cap_k} \leq Q_{max}} \end{gathered} \qquad (4\text{-}7)$$

The objective term can be divided into two parts:

1. WLS Term (First Part):

   - The term $\boldsymbol{\lambda_1 \chi_{WLS}\left(\hat{Q}_{Cap_k}\right)}$ represents a weighted-least-squares (WLS) term with a tuning parameter λ_1. This part ensures that the estimated capacity $\boldsymbol{\hat{Q}_{Cap_k}}$ aligns optimally with the observed data, considering the constraints imposed on the estimation process just like in the baseline SOH estimation algorithm. This term represents what is applied in the baseline SOH algorithm.

2. Relationship deviation Term (Second Part):

- The term $\lambda_2 \parallel Q_{L1_k} - \hat{Q}_{Cap_k} \parallel_2$ emphasizes on minimizing the (error) difference between estimated capacity $\hat{Q}_{Cap_k}$ in the first part and the modelled relationship using L1 sensor data Q_{L1_k} with a tuning parameter λ_2.

The objective function ($\mathcal{J}_k$) combines these two aspects. The first part ensures a good fit of the estimated capacity to the actual data just like in baseline SOH estimation algorithm, while the second part focuses on reducing the discrepancy between the predicted capacity and the capacity modelled using the relationship with 2D measurements and baseline data. The two parts are balanced using weighting factors λ_1 and λ_2 to create a comprehensive objective that guides the optimization process in estimating the most accurate and realistic capacity within the specified constraints. The optimization process aims to find the optimal $\hat{Q}_{Cap_k}$ that simultaneously satisfies the WLS fit and minimizes the deviation from the modelled relationship obtained from the 2D mechanical pressure sensor.

4.2.2 Pre-analysis of the data

Before diving into the detailed results obtained by merging baseline and 2D measurement data, it is important to have a look at the preliminary analysis of the pressure measurements available in Dataset 3.2. Currently, while awaiting specific measurements from the L1 sensor within the battery module, we already have valuable pressure data from an external one-dimensional sensor, which outputs a single mechanical pressure value over time. In the absence of L1 sensor data, our algorithm progresses by considering this existing external data and treating it as if it represents the average pressure P^{avg}. This allows us to start our analysis and algorithmic processes with the available information.

Returning to Dataset 3.2, we examine the pressure measurements acquired during both the cycling and characterization phases. The initial analysis, shown in Figure 23, exhibits the relationship between voltage and pressure across various charging and discharging cycles. This particular dataset belongs to cell #3, which is initialized at 144 kgf (76kPa) as seen in Figure 23. In this figure, the first cycle is observed as a hysteresis curve, highlighting distinct pressure values for charging and discharging, where the pressure is higher for charging than for discharging. The additional feature, the eye of the hysteresis curve, provides supplementary insights.

As we look into the first 50 cycles, an observation is noted—the hysteresis curve undergoes a shift during the 35th cycle. This signifies a dynamic change in the curve, indicative of potential variations in battery thickness. After the characterization, which also includes a 3C discharge leading to a substantial capacity drop and eventually substantial impact on SOH, we observe drastic shift in the hysteresis curve as seen in cycle 64 and cycle 127.

Figure 24 further proves this phenomenon, presenting a graph of the eye of the hysteresis, pressure range and SOH. The graphical representation in Figure 24 distinctly illustrates a correlation: as SOH diminishes, the hysteresis curve shift intensifies. This analysis provides a comprehensive understanding of the dynamic interplay and the corresponding relationship between pressure range and the decreasing SOH in the battery.

Figure 25 provides an additional insight into the relationship between pressure and SOH. Here, we observe a significant relationship when we plot pressure as a function of SOC for distinct discharge cycles plotted at every SOH%. In general, the pressure-SOC relationship is influenced by the SOH value of the battery. This influence stems from an irreversible increase in thickness as the cell ages, leading to a distinct shift in all pressure measurements towards higher values as the SOH decreases over time. This shift occurs due to the irreversible expansion of electrodes with decreasing SOH. This narrative aligns with the observation in [17]. This analysis

gives us enough motivation to include pressure measurements in SOC and SOH estimation as proposed in the previous subsections. The next steps are to establish the function relation between capacity (Q_{L1}), baseline measurements (V, I), estimated SOC and 2D measurement data ($P^{1,1}, \dots, P^{n_x,n_y}$ and $T^{1,1}, \dots, T^{n_x,n_y}$). Once this part is done, we can perform an improved SOH estimation procedure on the same dataset using the L1 sensor data of mechanical pressure.

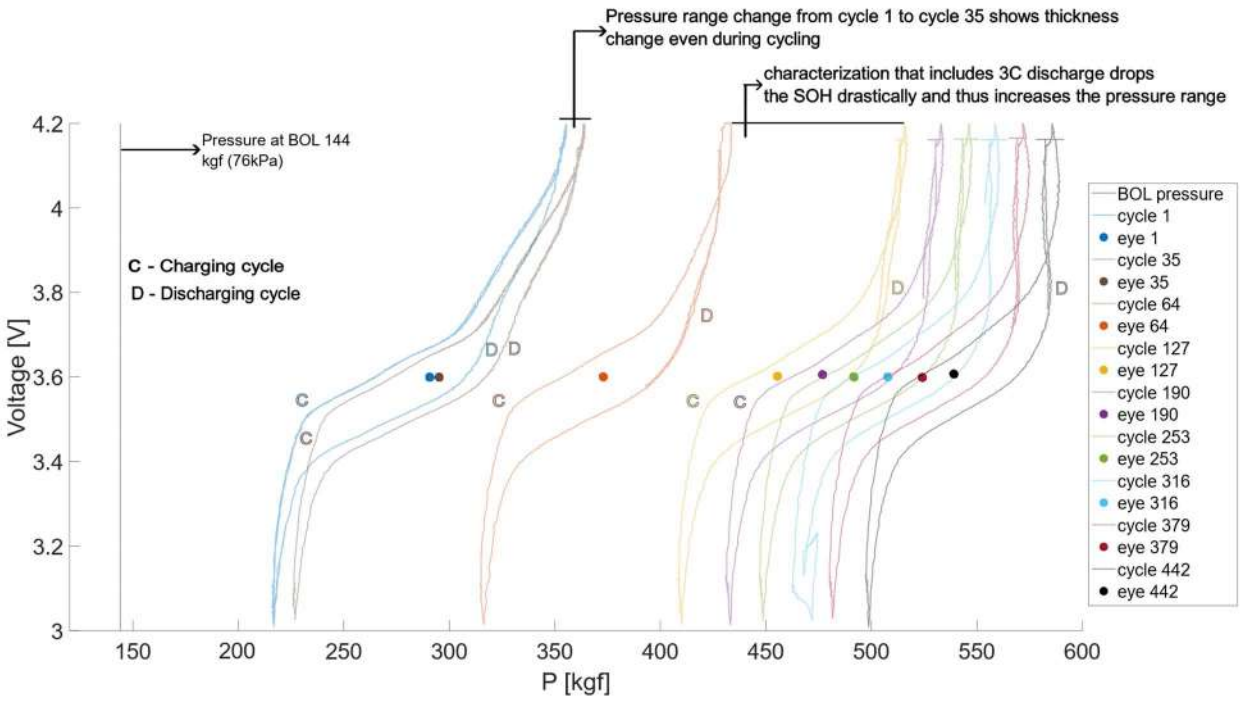

Figure 23 Pressure vs Voltage relationship for distinct charging-discharging cycles obtained from the measurement data of cell #3 (Dataset 3.2)

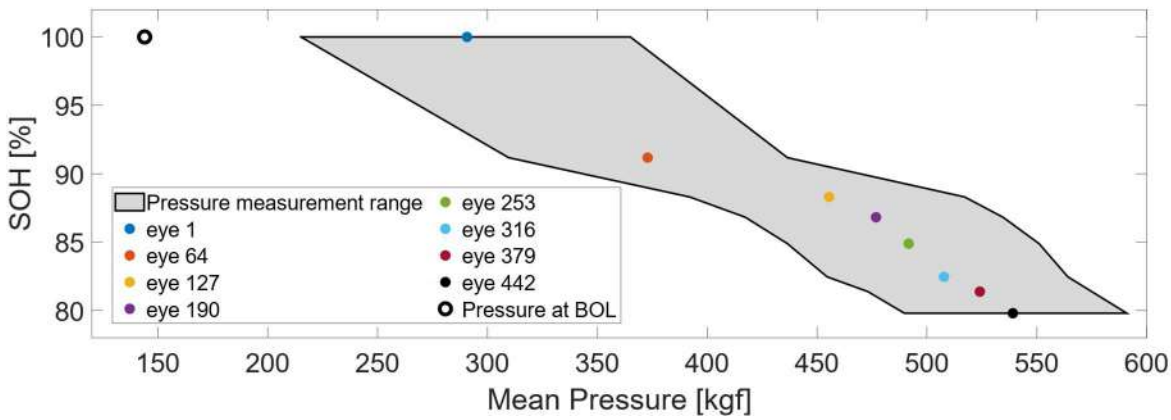

Figure 24 Pressure vs SOH relationship obtained from the measurement data of cell #3 (Dataset 3.2)

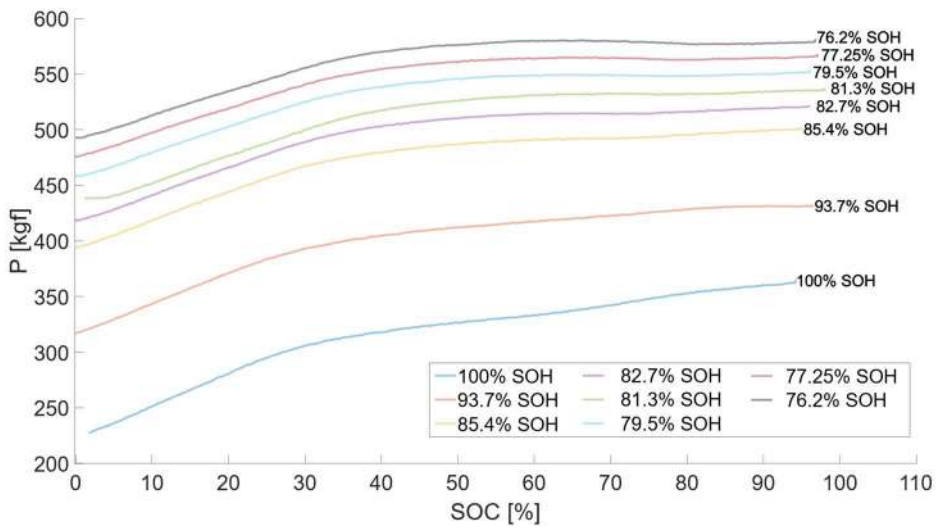

Figure 25 Pressure as a function of SOC for a discharging cycle at different SOH for cell #3 (Dataset 3.2)

Similarly, Figure 26 exhibits the relationship between voltage and pressure across various charging and discharging cycles particularly for another dataset that corresponds to cell #4. Figure 27 further proves the same phenomenon that we observed in Figure 24, presenting a graph of the eye of the hysteresis, pressure range and SOH. The only difference we observe is the fact that cell #4 is initialized at 231 kgf (121 kPa). The graphical representation in Figure 27 distinctly illustrates a correlation: as SOH diminishes, the hysteresis curve shift intensifies.

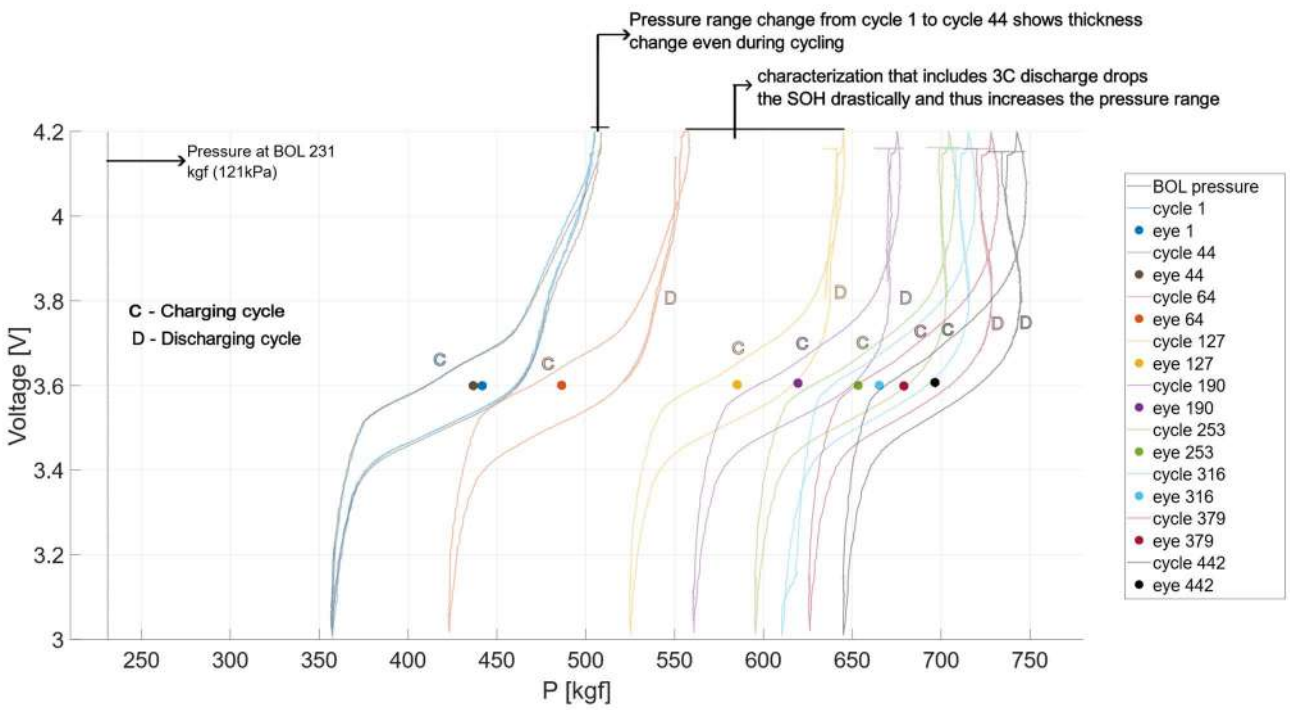

Figure 26 Pressure vs Voltage relationship for distinct charging-discharging cycles obtained from the measurement data of cell #4 (Dataset 3.2)

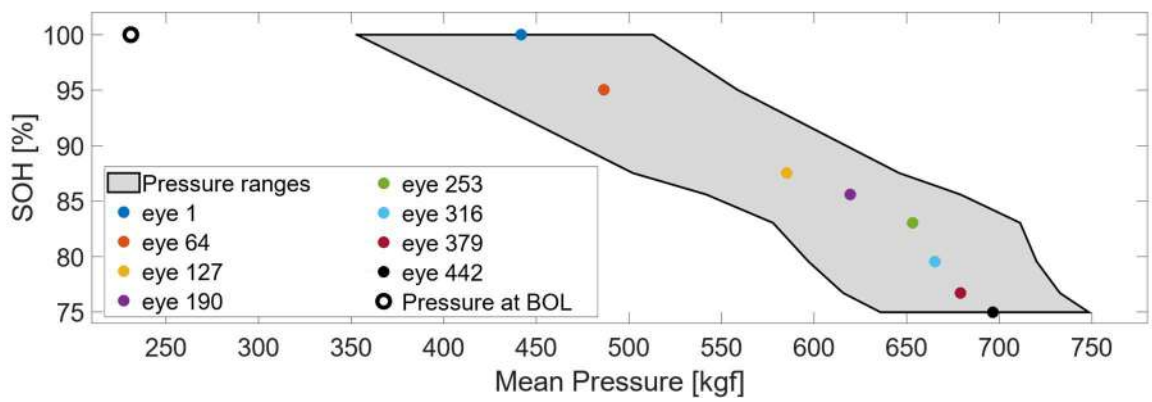

Figure 27 Pressure vs SOH relationship obtained from the measurement data of cell #4 (Dataset 3.2)

4.2.3 Results and Conclusions

In order to see how our SOH algorithm works with incorporating one-dimensional pressure measurements (which corresponds to taking the average value of what would eventually be read from a two-dimensional pressure L1 sensor), we first need to obtain the functional curve relationship denoted by Q_{L1} between capacity and averaged pressure measurements, as seen in (4-5). We simplify the relationship to the following, notice the removal of SOC in the function:

$$Q_{L1} = f(P^{avg}, V, I, T^{avg}) \qquad (4\text{-}8)$$

In order to obtain the above-mentioned relationship, we first attain a linear interpolated capacity-versus-real-capacity curve using the capacity checkup characterizations from the dataset of cell #3. We then choose the input time signal dataset of pressure and voltage, along with current and temperature measurements and derive a model that fits the function (4-8) with the outputs as the linear interpolated capacity. This model allows us to obtain instantaneous capacity interpolating between the input points of pressure, voltage, current and temperature.

Of course, one can also include SOC, but due to brevity and reducing the dimensionality by one, we proceed with the above-mentioned function. In order to do so, we incorporated the whole data as a scattered interpolant function offline to obtain (4-8). We use the same Dataset 3.2 as in the case of Results and Conclusions 3.4.2, with pressure information as in the subsection 4.2.2. This is the same dataset for the cell marked as cell #3. The hyperparameters for SOH estimation algorithm are similar as the ones validated in the Results and Conclusions 3.4.2. To summarize, we first consider the SOC estimation window of 85% to 15% during every available discharge cycle with a random noise of 0.2% added to the SOC starting point of the window and the end of the window to avoid the same SOC error accumulation. As for the results shown below, we chose $\Delta t_{diff} = 10s, M = 10^3$,$\Delta t_{SoH} = 10s$ and $\Delta t_{SoC} = 1s$.

The major differences in SOH estimation when exploiting the L1 sensor data are the optimization formulation as denoted by (4-6) and (4-7) as compared to (3-9) and (3-10), respectively. The optimization formulation has a new objective function ($\mathcal{J}_k$) that balances the WLS term with the deviation between the predicted (via the WLS route) and modelled capacity-versus-pressure relationship. The next step and the most important one is to choose the tuning parameters λ_1 and λ_2. Clearly, $\lambda_2 = 0$, will give us the same results as in the baseline sensor case as seen in the subsection 3.4.2, i.e., no use of the pressure-capacity relationship. On the contrary, using $\lambda_1 = 0$, will give us capacity directly from the relationship as obtained in (4-8). Although, this will give us the best result especially if it is used for the battery for which the relationship in (4-8) is obtained. However, this relation will slightly differ for another battery, as can already be seen by comparing Figure 23 to Figure 26. Eventually, the ideally way is to fuse these two entities as discussed in the previous subsection 4.2.1, such that the first part ensures a good fit of the estimated capacity to the actual data just like in baseline SOH estimation algorithm, while the second part focuses on reducing the discrepancy between the predicted capacity and the capacity modelled using the relationship with pressure.

Note: As mentioned before in subsection 3.4.2, in order to validate the quality of the SOH estimation, we need to validate on the Capacity checkup points that are available to us from the characterization cycles performed after each group of 50 charge-discharge cycles. We assume that in between these checkup points the true capacity follows a linear trend/linear interpolation.

The RMSE results obtained are shown in Figure 28Figure 28. As discussed, the best RMSE result is shown by the green marker in Figure 28Figure 28 which corresponds to $\lambda_1 = 0$. This gives us capacity directly from the relationship as obtained in (4-8), since it minimizes the error between the predicted capacity and the capacity modelled from the relationship for this exact cell #3. For reference, we have also shown the grey marker, that corresponds to $\lambda_2 = 0$, which is the exact result obtained from the baseline SOH algorithm, i.e., WLS route and it is ***0.83% RMSE***, which, as expected, corresponds to the error found in section 3.4.2, see Table 1. As seen in Figure 28, for $0.5 > \lambda_1 > 0$ and $1 > \lambda_2 > 0.5$, which means more focus is given to minimizing the error between the predicted capacity and the modelled relationship, the RMSE results are amongst the best, i.e., RMSE $< 0.1\%$. One may be tempted to use this scenario, but one should always avoid this. This is because the modelled

relationship as shown in (4-8) is derived for the cell #3 and the results are validated also on the cell #3 and hence are biased. On the contrary, for $0.5 > \lambda_2 > 0$ and $1 > \lambda_1 > 0.5$, we focus more on the WLS route, i.e., the real-time data and the past measurement data for the cell are considered.

The tuning parameters $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$, yields ***0.11% RMSE for SOH estimation***, while the tuning parameters $\lambda_1 = 0.75$ and $\lambda_2 = 0.25$, yields ***0.62% RMSE for SOH estimation.*** Of course, an in-depth analysis can be done on finding an optimum set of tuning parameters, however, this will be out of scope of this deliverable. For now, we can assume the range of optimal set of tuning parameters as $0.75 \geq \lambda_1 \geq 0.5$ and $0.5 \geq \lambda_2 \geq 0.25$. Figure 29Figure 29 shows the comparison between the Capacity checkup points and the estimated capacity for each iteration of checkups for the case of tuning parameters as $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$. **In conclusion, these results show that incorporating pressure information in SOH algorithm yields improved RMSE as compared to just using the WLS route, i.e., $\lambda_2 = 0$.**

The optimization problems have been solved offline and the offline simulations have been performed using "The Optimization toolbox" using *fmincon* framework in Matlab R2021a. The simulations were carried out on a PC with an Intel Core i7 and 16GB memory. It is important to also comment on the computation or the solving time taken to solve and find out the optimal capacity at every sample. In our simulations, the capacity estimation is performed every second and the solving time is recorded. Figure 30Figure 30 shows the computation time vs every sample run for the entire dataset for the case of the tuning parameters $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$ along with a zoomed version of the first part of the data set. It is easily observable that the computation time is within 0.5 seconds, except initially where it took 7.5 seconds, and occasionally it took 1 second, which are still within the 10 second time allotted to estimate the capacity with Δt_{SoH}. This shows the real-time deployment capability of our proposed algorithm for SOH estimation.

We steer the readers to the fact mentioned before, the modelled relationship as shown in (4-8) is derived for the cell #3 and the results are validated also on the cell #3 and hence are biased. In order to have proper validation, the SOH estimation algorithm needs to be validated on the dataset obtained from cell #4.

In order to validate on a different cell, we first re-derive the functional relationship as shown in (4-8) with the relative pressure measurements instead of absolute averaged pressure measurements from the cell #3. The RMSE for SOH estimation for cell #4 is 1.61% for $\lambda_1 = 1$ and $\lambda_2 = 0$, i.e., focussing only on the WLS route with historical measurements. The higher RMSE as compared to cell #3 might be because of the missing data in the third and fifth 50 discharging-charging cycles and in the eight performance characterizations. This has introduced discrepancies, uncertainties and discontinuities in our SOH algorithm. This demonstrates that even though in the presence of discrepancies, our algorithm has a capability to achieve result way under ~2% RMSE. On inclusion of functional relationship Q_{L1} derived from the measurements of cell #3, and choosing $\lambda_1 = 0.8$ and $\lambda_2 = 0.2$, the RMSE for SOH estimation for cell #4 is 1.53%.

Note: With the tuning parameters $\lambda_1 = 0$ and $\lambda_2 = 1$, i.e., focussing only on the functional derived from cell#3, the RMSE for SOH estimation for cell #4 was reduced to 1.39%. To summarize, we found that incorporating pressure measurements in the SOH estimation algorithm for L1 cell only slightly improved our results. In general, using only the functional relationship yielded the best results, with asn RMSE of 1.39%. This RMSE is a bit high and the explanation is as follows: We are limited to only two datasets with pressure measurements. An ideal methodology will be to obtain the functional relationship shown in (4-8) for at least two sources, so at least two datasets, and then validate on a third dataset. This strengthens this function and reduces the chance of adding bias in our algorithm. We are awaiting L1 measurements from the module in development and once we have attained the measurements the first task would be to

re-derive the functional relationship using datasets from cell #3 and cell #4 and validate the SOH estimation algorithm for the L1 cell on these measurements. Due to brevity and time constraints, we only use scattered interpolations to derive the functional relationship, however it is recommended and preferrable to obtain and compare several models other than interpolants, like regression models, Support Vector Machines, Gaussian models, or even Neural Networks to validate our proposed SOH algorithm.

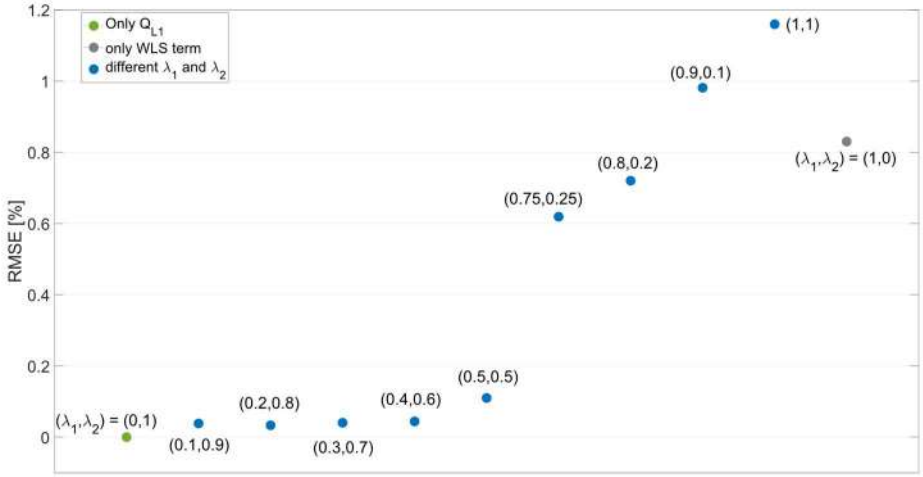

Figure 28 RMSE results with different tuning parameters λ_1 and λ_2 for the cost function $\mathcal{J}$ for cell #3 (Dataset 3.2)

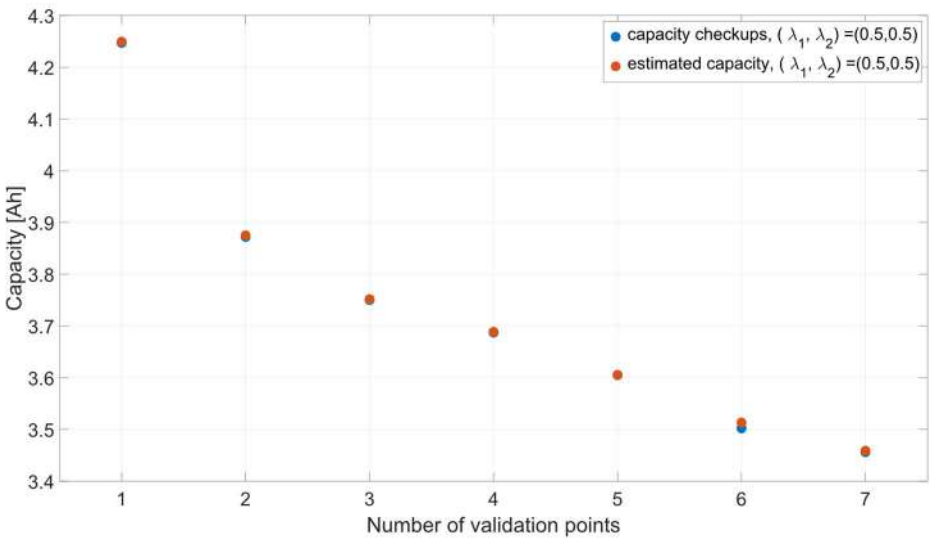

Figure 29 True and estimated capacity with SOH estimation algorithm results for $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$ for cell #3 (Dataset 3.2)

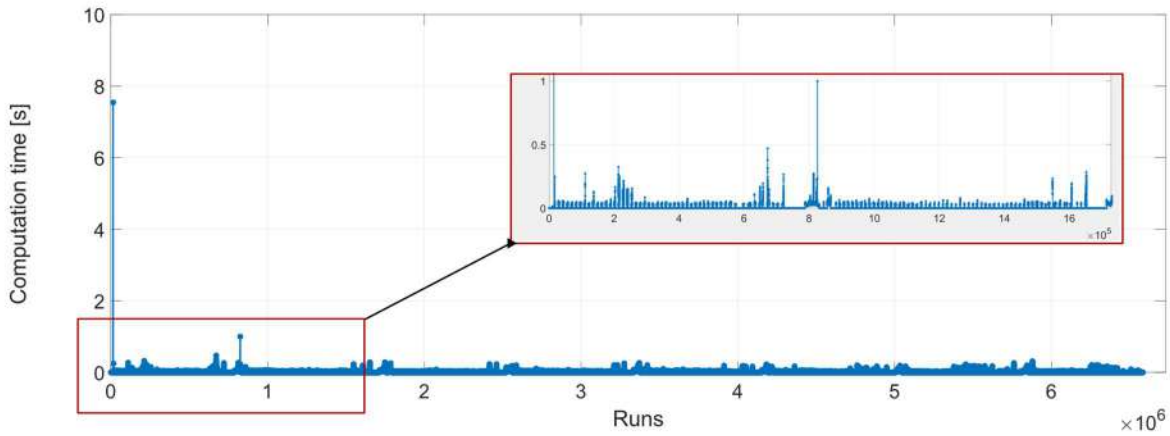

Figure 30 Computation time for each run for $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$ for cell #3 (Dataset 3.2)

4.3 SOS

In this section, the L1 version of the State-of-Safety (SOS) algorithm is presented, which makes use of the data from the battery built-in novel sensor technology developed in the SENSIBAT project (L1 sensor technology, including 2D temperature and pressure measurement inside the pouch cell). First, the updated definition of the SOS is presented, which is based on the same modular concept presented in Section 3.5 for the baseline

algorithm. Then, the new version of the algorithm is parametrized. Finally, the algorithm is validated with data from the laboratory tests performed to the L1 cells (Dataset 3.2).

4.3.1 Updated SOS algorithm with L1 sensor

The L1 version of the SOS algorithm is based on the same concept of abuse presented in Section 3.5.1, which was derived into equation (3-14). This expression can be used for any term affecting the safety state of the battery. The aggregation (via multiplication) of the different terms allows a modular definition of the SOS, which can be extended as more physical quantities are measured in the battery.

In the case of the baseline algorithm, only 1D measurements of the voltage, current and temperature were available, what lead to the SOS expression presented in equation (3-15). In the case of the L1 cells, the temperature measurement is extended to a 2D matrix, and additionally, pressure measurements are available (also in 2D). This brings two improvements to the SOS concept: one related to the new available measurement (pressure), and the other one related to the 2D matrix of measurements.

Starting with the new available measurement, the pressure is conceived to be a similar concept as the temperature: both the absolute pressure value and the pressure derivative are assumed to be dangerous for the battery, as they may be a sign of collapse. However, contrary to the temperature case, the pressure shows a non-reversible increase nature through the battery lifetime, what is typically linked to the increase of the SEI layer. This phenomenon can be seen in Figure 23 and Figure 26 of Section 4.2, and was already previously analysed.

Therefore, it is considered that the time-to-pressure concept proposed for the temperature is not valid for the pressure case. Indeed, inevitably, the time-to-pressure value would be much higher at beginning of life, and rapid pressure increases may be shadowed. Therefore, instead of the single safety function related to the time-to-pressure, it is proposed to add two pressure-related safety functions to the general SOS expression: one safety function for the absolute pressure value, $f_{safety}(P)$; and another safety function for the pressure derivative, $f_{safety}(dP)$.

In the case of the absolute pressure, it is also important to highlight that this value is affected by the pressure applied to the cell at beginning of life. Figure 23 and Figure 26 of Section 4.2 help visualizing this issue. As it can be seen, the two L1 cells cycled in SENSIBAT project (cell #3 in Figure 23, and cell #4 in Figure 26) were pressured at different initial values: 144 kgf and 231 kgf, respectively. In both cases, the cells were at the same rest voltage at this point (3.65V). After 442 cycles, the step of pressure was similar in both cells: from 144 kgf to approximately 580 kgf in cell #3 (~440 kgf), and from 231 kgf to approximately 750 kfg in cell #4 (~520 kgf). But of course, due to the difference at beginning of life, the absolute pressure value was different. Therefore, instead of the safety function affected by the absolute pressure, the use of the relative pressure (in relation to the beginning of life pressure) is proposed: $f_{safety}(P_{rel})$.

This leads to a new conception of the SOS expression:

$$\boldsymbol{SOS = f_{safety}(V) \cdot f_{safety}(I) \cdot f_{safety}(T) \cdot f_{safety}(P_{rel}) \cdot f_{safety}(dP) \cdot f_{fault}} \quad \textbf{(4-9)}$$

Due to the addition of two new terms, the thresholds in the SOS overall value slightly vary:

- $SOS = 100\%$ means that all the individual terms are at 100%. Until $SOS = 80\%$ it may be safe to use the battery.

- $SOS < 80\%$ means that most of the terms are close to their safe limit (80%), or that one of the terms is below that threshold. In the first case, a warning is generated. In the second case, it is considered that the battery is already at a non-safe state. Each time a term of the SOS falls below 80%, an alarm related to that term is generated.
- $SOS < 26\%$ (0.8^6) means that all the terms (6 terms) might be below their safe limit, or that one of the terms is very low. This is defined as a completely non-safe state.

The following figure depicts the graphical representation of the L1-SOS algorithm, with the addition of the new two terms.

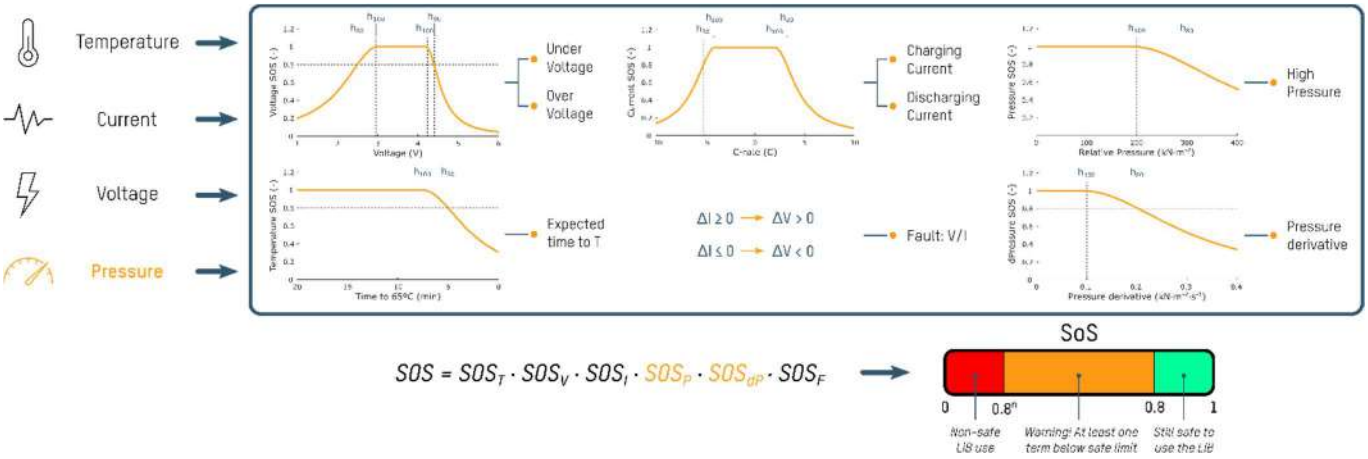

Figure 31: Representation of L1-based State of Safety (SOS).

On the other hand, and as stated at the beginning of this section, the disposal of temperature and pressure measurements in a 2D matrix brings the possibility of further improving the L1-SOS algorithm. In the case of the previously presented L1 algorithms (SOC in Section 4.1 and SOH in Section 4.2), different approaches were proposed in order to integrate the 2D matrix measurements. In short, the two most likely approaches to be replicated with the SOS algorithm consist of:

1. Estimating a single SOS value with the average measurements of the 2D matrix, or
2. Estimating a SOS value for each of the points of the 2D matrix.

In the case of SOC and SOH algorithms, it was decided to use the first approach, due to the high computational burden of these algorithms. However, in the case of the SOS, as the algorithm is computationally light, it is preferred to go for the second approach. It is also worth to point out that, in the case of the SOS, it is considered to be crucial to detect any point of the cell whose pressure or temperature is increasing (rather than using the average of the matrix).

In conclusion, an SOS value is generated for each spot of the 2D matrix ($SOS_{i,j}$), and the overall SOS is equal to the worst $SOS_{i,j}$ value. This is represented as the following expressions:

$$\boldsymbol{SOS = min(SOS_{i,j}) \quad \forall i \in I, \forall j \in J} \tag{4-10}$$

$$\boldsymbol{SOS_{i,j} = f_{safety}(V) \cdot f_{safety}(I) \cdot f_{safety}(T_{i,j}) \cdot f_{safety}\left(P_{rel_{i,j}}\right) \cdot f_{safety}(dP_{i,j}) \cdot f_{fault}} \tag{4-11}$$

Being i, j any point in the 2D matrix of measurements (I and J represent the sets of rows and columns of the 2D matrix, respectively).

4.3.2 Parametrization of L1 SOS algorithm

As in the case of the baseline SOS algorithm, it is necessary to parametrize the x_{80} and x_{100} components of equation (3-14), for all the individual safety functions composed the overall SOS expression, which was previously presented in equation (3-10).

For the safety functions already used in the baseline algorithm, the same parameters have been defined (as their effectiveness was already validated). In the case of the two safety functions introduced in the L1-SOS algorithm, $f_{safety}(P_{rel})$ and $f_{safety}(dP)$, the safety thresholds have been defined based on the data generated in the SENSIBAT project, specifically the data from Dataset 3.2 (ageing tests to L1 cells). On the one hand, the thresholds for the relative pressure term have been defined based on the pressure increase that the cells suffered during their useful life, which was already showed in Figure 23 and Figure 26. On the other hand, in the case of the pressure derivative term, the data from the performance tests has been analysed. As already mentioned, the performance or characterization tests are the other routines included in the ageing tests, which are executed between each cycling routine. Indeed, the highest peaks in the pressure derivative were found during these test routines, due to the fact that the cell was charged at 3C, above the recommended maximum current (2C). After each performance test, an irreversible pressure increase was also found, which approximately coincides with the gaps between the cycling tests depicted in Figure 23 and Figure 26.

In conclusion, Table 3 shows the set of parameters defined for the L1 version of the SOS algorithm.

SOS Term	***Parameter***	***Value***
Voltage (High)	x_{100}	4.2 V
	x_{80}	4.4 V
Voltage (Low)	x_{100}	3 V
	x_{80}	2.5 V
Current (Charge)	x_{100}	2 C
	x_{80}	3.2 C
Current (Discharge)	x_{100}	4 C
	x_{80}	5.2 C
Time to 60ºC	x_{100}	7.5 minutes
	x_{80}	5 minutes
Relative Pressure	x_{100}	230 kPa (435 kgf)
	x_{80}	330 kPa (624 kgf)
Pressure derivative	x_{100}	0.1 kPa/s
	x_{80}	0.2 kPa/s

Table 3. Parametrization of L1-SOS algorithm.

4.3.3 Validation of L1 SOS algorithm

As mentioned in the baseline SOS section, no safety tests were performed to L1 cells, due to the issues generated with these cells (see deliverable 5.1 [4]). Therefore, it has been decided to validate the L1-SOS algorithm with the only available data for the L1 cell, i.e., the ageing tests (Dataset 3.2). As already mentioned during the document (e.g., see note at the beginning of Section 4.2), due to the unavailability of L1 sensor read-out circuits when executing these tests, the temperature and the pressure were externally measured, with a single measurement point per cell. Therefore, it is only possible to validate the 1D version of the L1-SOS algorithm, which corresponds to the expression (4-9).

Compared to the cycling phases, the performance phases of the ageing tests (Dataset 3.2) consisted of more heterogeneous conditions, as cycles with different rates and lengths were concatenated. Therefore, the

validation is carried out with two performance tests: one near beginning of life (named Performance Test #1), and the other one near the end of life (named Performance Test #12). The main difference between both tests lies in the fact that near the end of life the cell has already suffered an irreversible pressure increase (see Figure 23 and Figure 26), and therefore the pressure-related SOS term will be around its safe limit. In both cases data from cell #4 is used, as it shows a slightly bigger pressure increase during its useful life.

4.3.3.1 Performance Test #1

Figure 32 shows the data recorded during the Performance Test #1, for Cell #4 of Dataset 3.2: current, voltage, temperature and external pressure. The whole test lasts for more than 100 hours (>4 days). In a first step, the whole test will be analysed. Then, a zoom will be made to specific parts of the test in order to analyse with more detail the effectiveness of the algorithm.

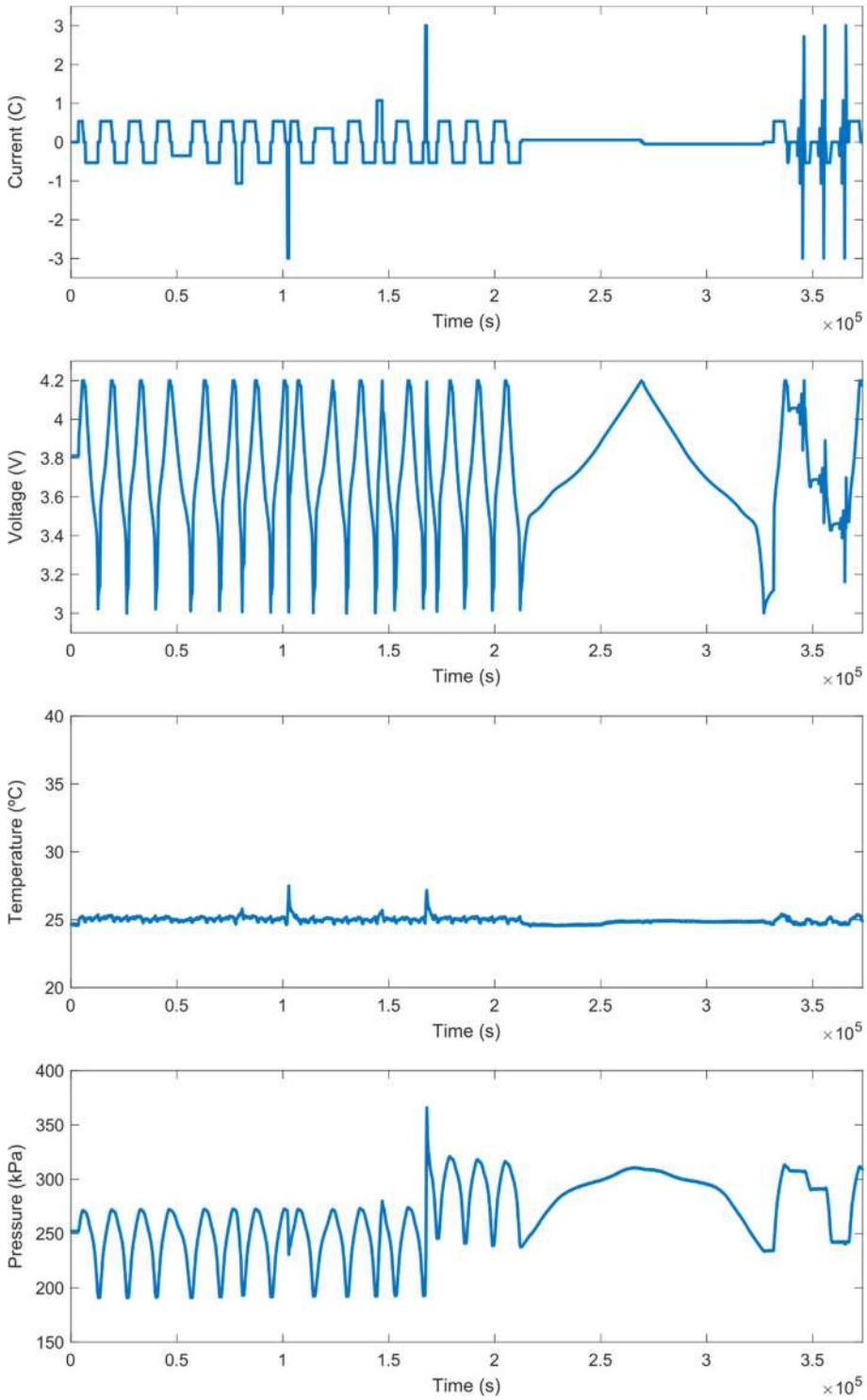

Figure 32: Data recorded in Performance Test #1 (Cell #4, Dataset 3.2): current, voltage, temperature and pressure.

As it can be seen, the whole performance test consists of a series of different cycles. In the first part, full charging and discharging cycles at different C-rates are concatenated (including one charging cycle at 3C, and one discharging cycle at 3C). After these full charge and discharge cycles, pulses at different C-rates and at different SOC values are also executed, in order to evaluate the internal resistance of the cell. Some of these pulses reach also the 3C.

The graphs show that the temperature follows a practically constant trend, with only two small peaks (around 3ºC increases) when executing the 3C charge (around t=1.7x10^{5}s) and discharge cycles (around t=1x10^{5}s),

respectively. Besides, the pressure follows a shape similar to the voltage: as the cell is charged, the pressure increases; and as the cell is discharged, the pressure decreases. The point to be highlighted is the charge at 3C: during this charge period, the pressure increases much more than in previous cycles; and when the cell is discharged, it does not recover the original pressure (irreversible pressure increase). This phenomenon is found during the 3C charge, but not during the 3C discharge.

Figure 33 shows the outputs of the L1-SOS algorithm when executing it with the data previously presented. The first graph shows the overall SOS value, i.e., the output of equation (4-9). This graph also shows the SOS value that the algorithm would return if only baseline measurements were available (i.e., without pressure measurement). Besides, the second graph shows the decomposition of the overall SOS into its different terms: temperature, current, voltage, fault, pressure and pressure derivative. Finally, the last two figures show the warning level (completely unsafe, unsafe, warning and safe operation, according to the defined SOS thresholds) and the triggered alarms (i.e., which SOS term is below 80%).

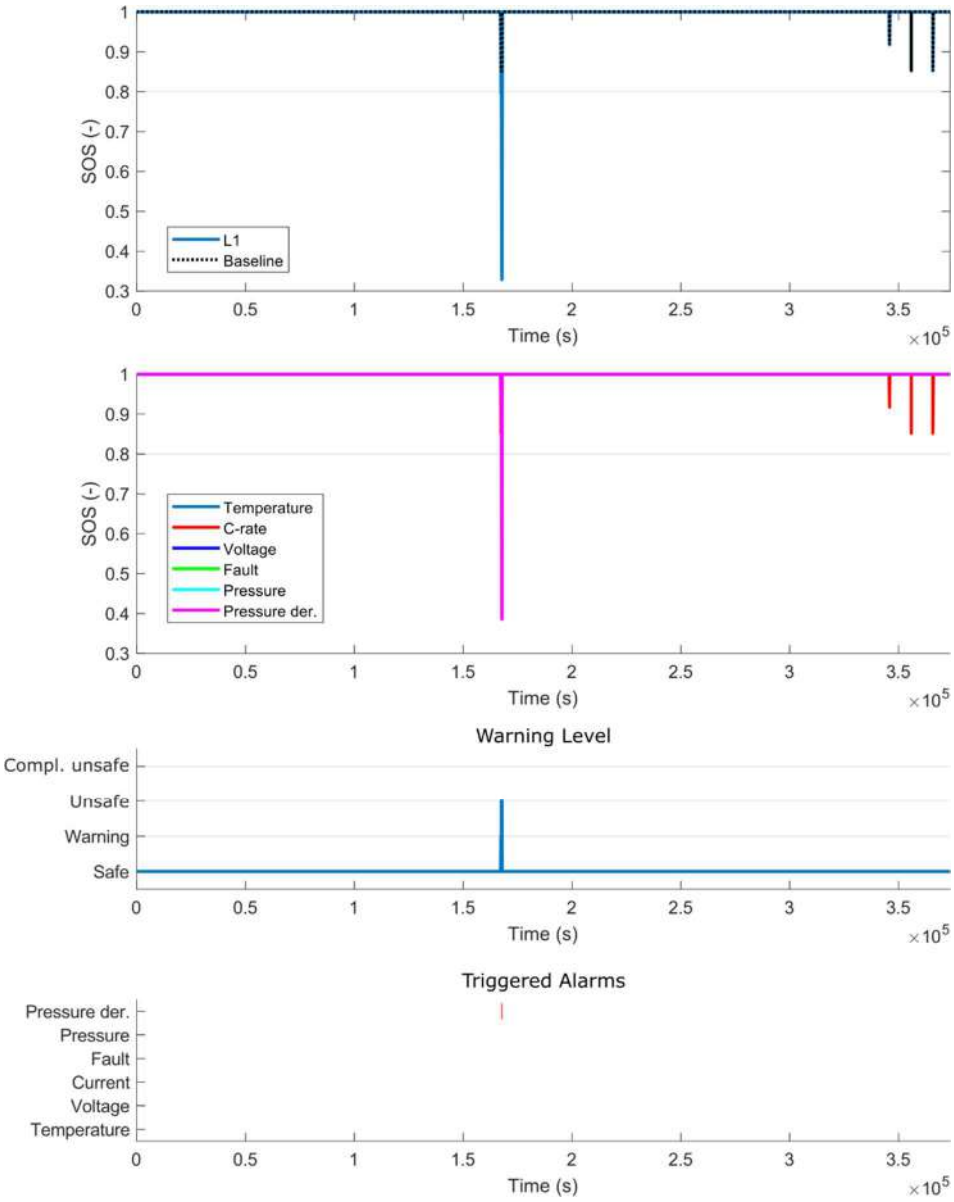

Figure 33: Output of SOS algorithm for Performance Test #1 (Cell #4, Dataset 3.2): Overall SOS, SOS by terms, warning level and triggered alarms.

The results show that there is only one part of the test where an alarm is generated, which coincides with the charging cycle at 3C. During this charge, the pressure increases fast, and the SOS drops until almost the 30%. During the remainder of the performance test, the SOS remains at 100%, except during the charge pulses at 3C, were the SOS drops a little bit due to the current-related SOS term (C-rate). Due to the length of the performance test (>4 days), it is not easy to see the evolution of the different terms during these charges at 3C in Figure 32 and Figure 33. In the following paragraphs, each of the phases is analysed in more detail.

On the one hand, Figure 34 shows the part of the performance test where the 3C charge is executed, from 3V until reaching 4.2V. Specifically, the three graphs show the current, pressure and SOS.

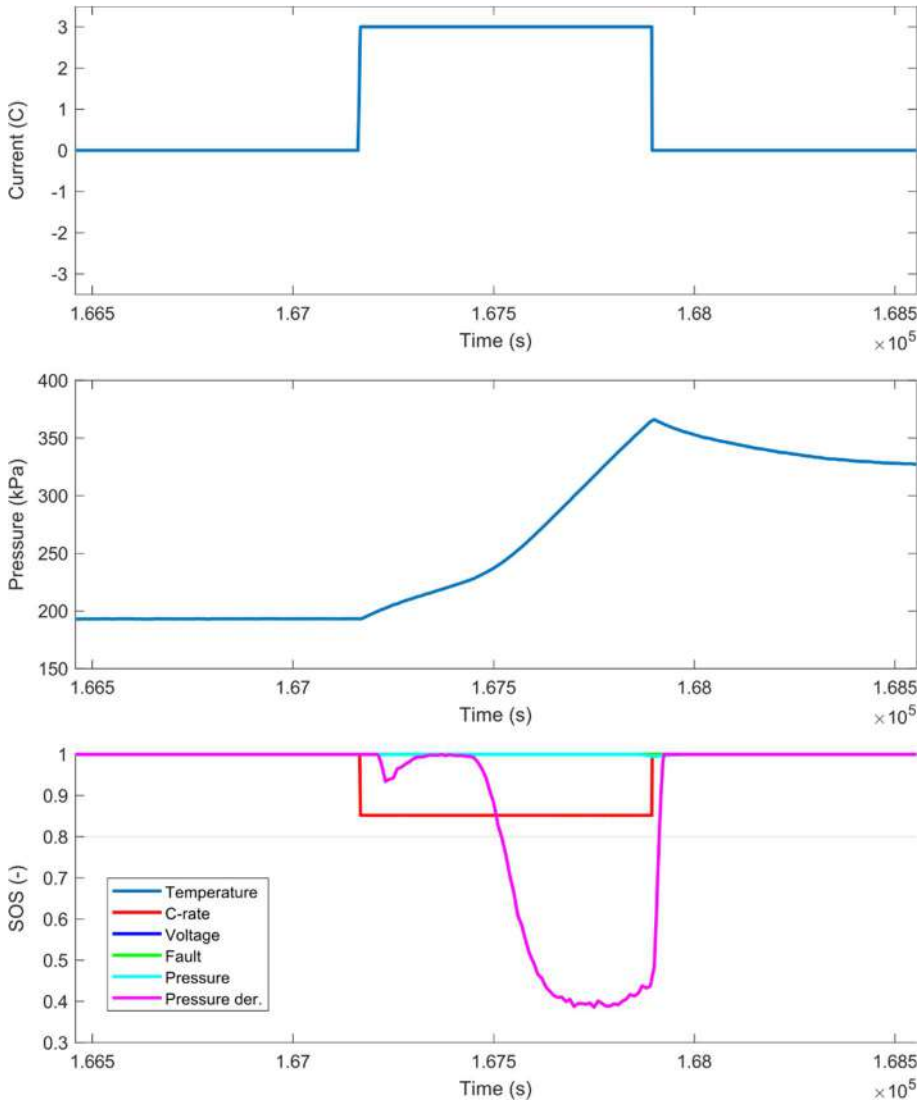

Figure 34: Zoom at charge cycle, 3C: Current, Pressure and SOS by terms.

As it can be seen in the pressure graph, the pressure starts increasing slightly, but around the middle of the charge the evolution gets steeper. At this point, the SOS drops below 80%, due to the term related to the pressure derivative. This term falls until the 40%, and it is maintained there until the charge ends, what inevitably triggers an alarm. During this part, the only other term that goes below 100% is the one related to the current. Anyways, as the x_{80} limit is defined at 3.2C, no alarm is triggered. Regarding the other terms, as it was previously mentioned, there is no significant temperature increase during this charge (around 3ºC), and the voltage is maintained between the safe limits. Also, considering that it is the first performance test, the absolute pressure is still low.

This part of the test demonstrates the additional information given by the L1 sensor, and how it can be successfully interpreted by the SOS algorithm. If only baseline measurements were available, the SOS would remain above 80%, without triggering any alarm, due to the fact that no important temperature or current thresholds were crossed. However, with the pressure measurements it is possible to see that the 3C charge is damaging the cell, as an irreversible pressure increase happens after charging the cell at this rate. Integrating the pressure measurement in the SOS algorithm enables triggering an alarm when the pressure increases too fast, what would prevent the cell from suffering the mentioned damage if appropriate measures are adopted.

On the other hand, Figure 35 shows the part of the performance test where the charge and discharge pulses are executed. As with the previous figure, in this case also current, pressure and SOS (divided in terms) graphs are depicted.

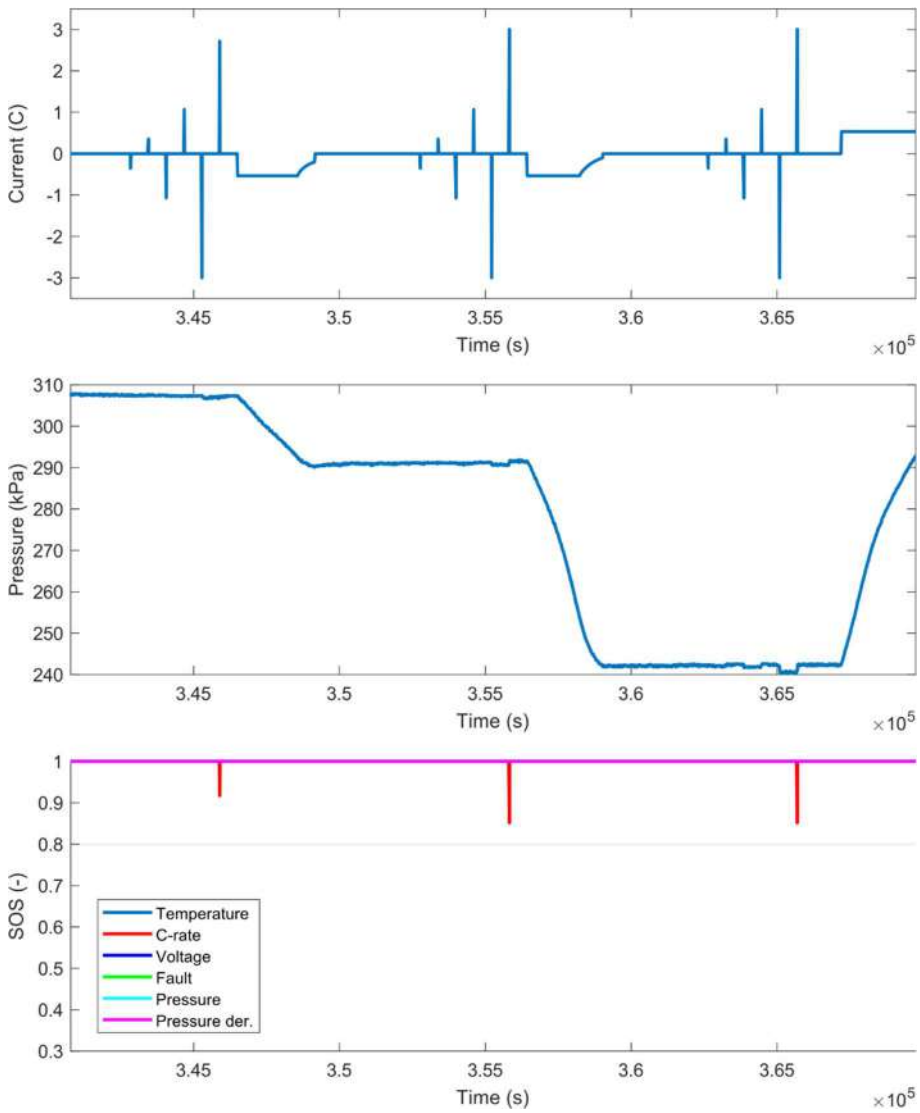

Figure 35: Zoom at charge pulses, 3C: Current, Pressure and SOS by terms.

The pulses depicted in the figure are around 30 seconds long, much shorter than the charge phase showed in Figure 34. Due to this fact, in this case there is no significant pressure increase during the charge pulses, even in the cases of 3C. What it is more, as the cell is discharged between each set of pulses, the absolute pressure drops. As it can be seen in Figure 32, during these phases there is neither significant temperature increase. Therefore, as it can be seen in the SOS graph, the only term that drops below 100% is the current-related one. In any case, as the values do not fall below 80%, no alarm is triggered. During this charge/discharge pulses,

there is no difference between the baseline and L1 SOS algorithms, as there is no significant difference in the pressure.

4.3.3.2 Performance Test #12

Figure 36 shows the data recorded during the Performance Test #12, for Cell #4. As in the previous test, the figure shows the current, voltage, temperature and pressure measurements.

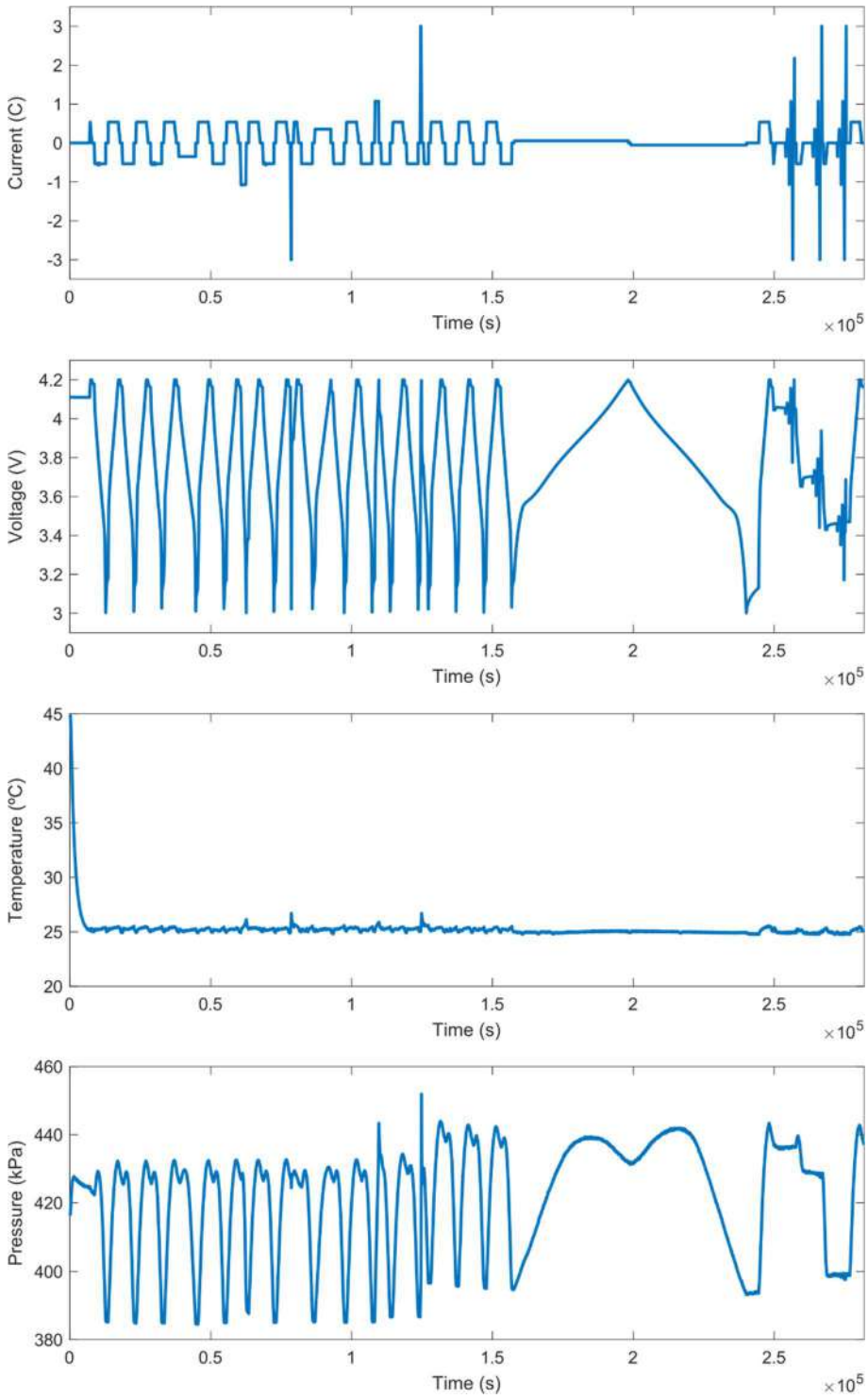

Figure 36: Data recorded in Performance Test #12 (Cell #4, Dataset 3.2): current, voltage, temperature and pressure.

At the time this test was held, the cell was already aged (around 75% SOH). Due to this reason, it reaches faster the voltage limits, and the whole test length is shorter: 78 hours (around 3,25 days). Before this test, the cell

performed a cycling test, which was held at 45ºC. This is the reason why at the beginning of the test the temperature drops from 45ºC to 25ºC, which is the ambient temperature at which the performance test was held. Another point to be highlighted is that, compared to the previous test, in this case the step in pressure after the 3C charge cycle is smaller. This is a trend that was noticed in both ageing tests performed to L1 cells: during the first tests, the irreversible pressure increase after the 3C charge is higher; but it is reduced in the next cycles.

Figure 37 shows the different outcomes of the SOS algorithm when executing it with the data previously presented. As in the previous test, the first graph presents the overall SOS value, the second graph the decomposition of the overall SOS into its different terms, the third graph the warning level, and the final graph the triggered alarms. The overall SOS also shows, as in the previous case, the SOS value that would be obtained if only baseline measurements were available.

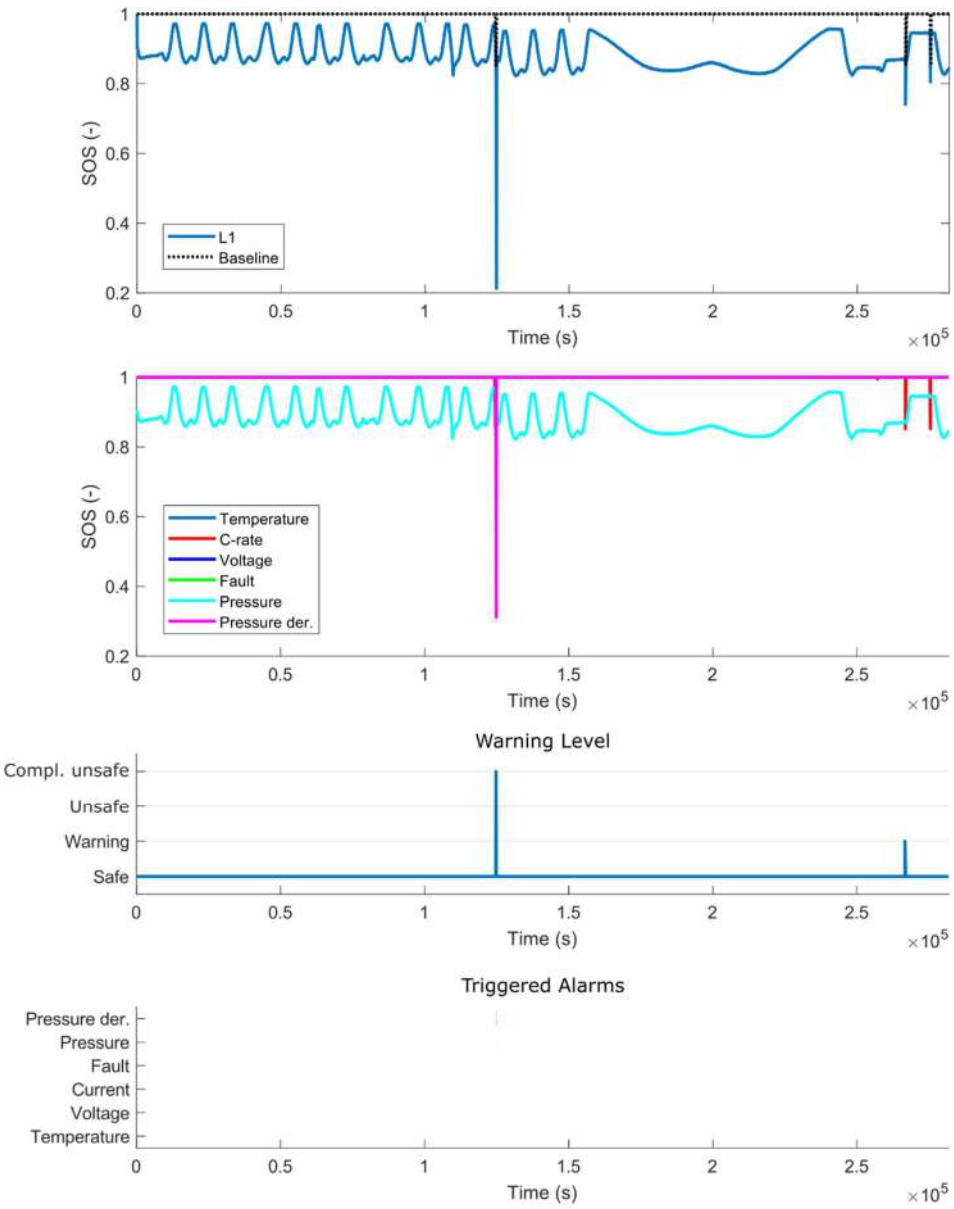

Figure 37: Output of SOS algorithm for Performance Test #12 (Cell #4, Dataset 3.2): Overall SOS, SOS by terms, warning level and triggered alarms.

The results show three main differences compared to Performance Test #1:

1) As the cell is already aged, its relative pressure from beginning of life has increased, getting closer to the defined safe threshold. Due to this reason, during the whole test, the pressure-related SOS term is below 100%. When the cell is around its full charge, it gets closer to the 80% safe limit. Anyway, the 80% threshold is only exceeded during the charging cycle at 3C, as will be analysed with more detail afterwards.
2) During the charging cycle at 3C, the overall SOS drops below the 21% threshold, triggering a "completely unsafe" alarm. In this case, the pressure derivative SOS reaches similar values (around 30%). But as the overall pressure is also below 80%, the different terms are aggregated, and the overall SOS is lower.
3) During the 3C pulses, a warning alarm is generated. Contrary to Performance Test #1, in this case the pressure-related SOS is below the 100% threshold (without reaching the 80% safe limit). Current-related SOS is also between 100% and 80%. As both terms are close to the 80% threshold, the overall SOS drops below the 80%, but without triggering an individual alarm.

During the following paragraphs and figures, the two phases of Performance Test #12 where alarms are triggered are analysed with more detail. On the one hand, Figure 38 shows the part of the performance test where the 3C charge cycle is executed. As mentioned in the previous performance test, the 3C charge is performed from 3V to 4.2V. The depicted graphs depict the current, pressure, SOS (divided by terms), warning level and generated alarms.

As the graphs show, the pressure has a two-step evolution during the charge: first, the pressure increases slightly, but then it gets steeper. This evolution was also noticed in Performance Test #1. Once the pressure starts to increase faster, the pressure derivative related SOS term starts to drop. Before crossing the 80% safe limit, the overall SOS is already below 80%, generating a warning alarm. This is due to the fact that the pressure related SOS is already around 90%, and the current related SOS around 85%. Consequently, the overall SOS falls below 80%, while none of the SOS terms are yet below 80%.

Once the pressure derivative-related SOS term falls below the 80% value, the SOS triggers an unsafe alarm. From this point on, this term falls rapidly, until almost reaching the 30% value around the end of the 3C charge. As the overall pressure has also increased, the pressure-related SOS term gets closer to the 80%. This makes the overall SOS to fall below 21%, and therefore a completely unsafe alarm is triggered. Around the end of the 3C charge, the pressure-related term also crosses the 80% threshold, and therefore an additional alarm is also triggered. Once that the 3C charge finishes, the pressure starts to drop, and consequently the SOS is again increased, turning off all the alarms.

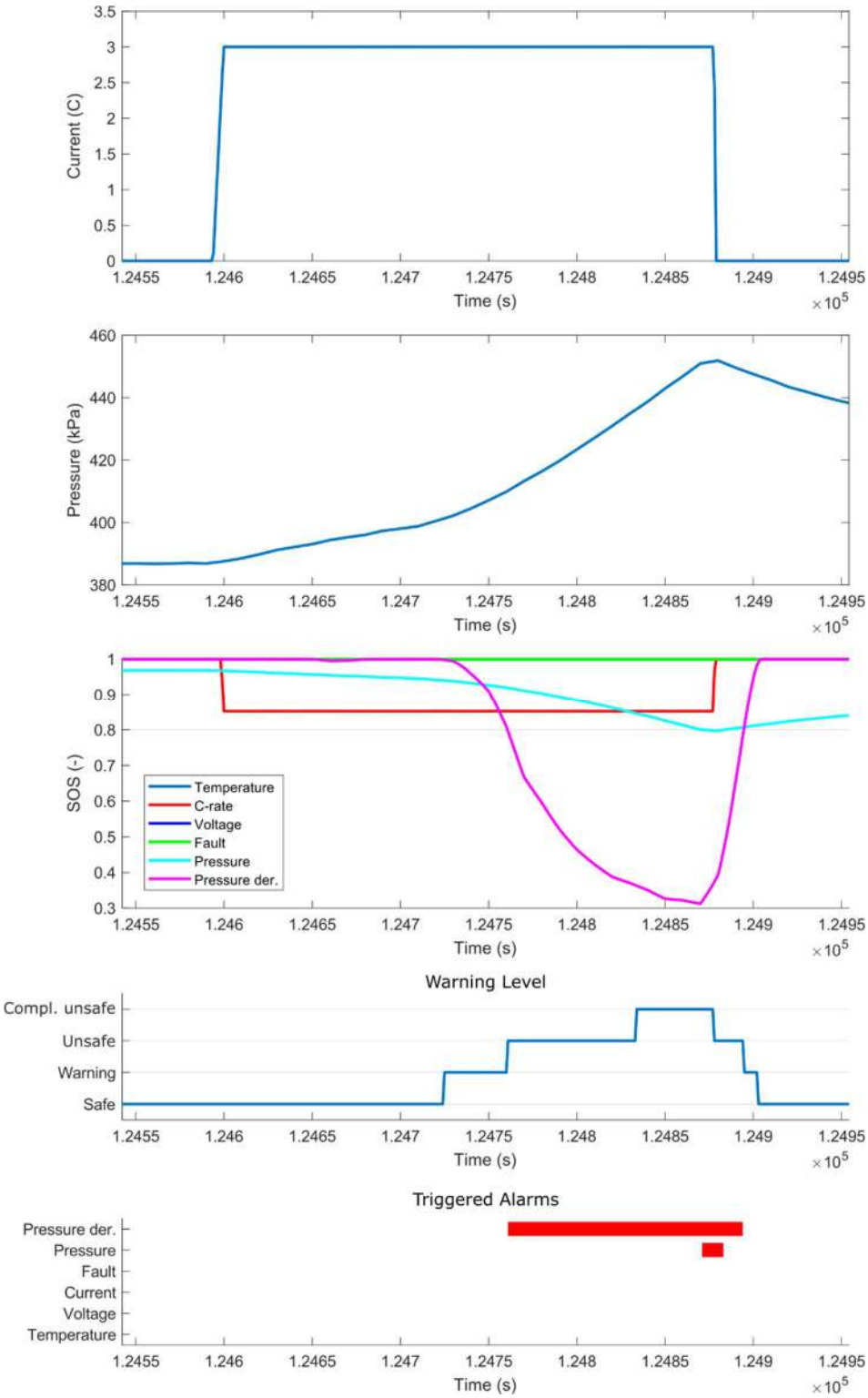

Figure 38: Zoom at charge cycle, 3C: Current, Pressure, SOS by terms, warning level, and triggered alarms.

On the other hand, Figure 39 shows the part of the performance test where two of the charge and discharge pulses are executed. The graphs depict the current, pressure, SOS (divided by terms), and warning level. As in this case none of the SOS terms fall below 80%, no alarms are generated.

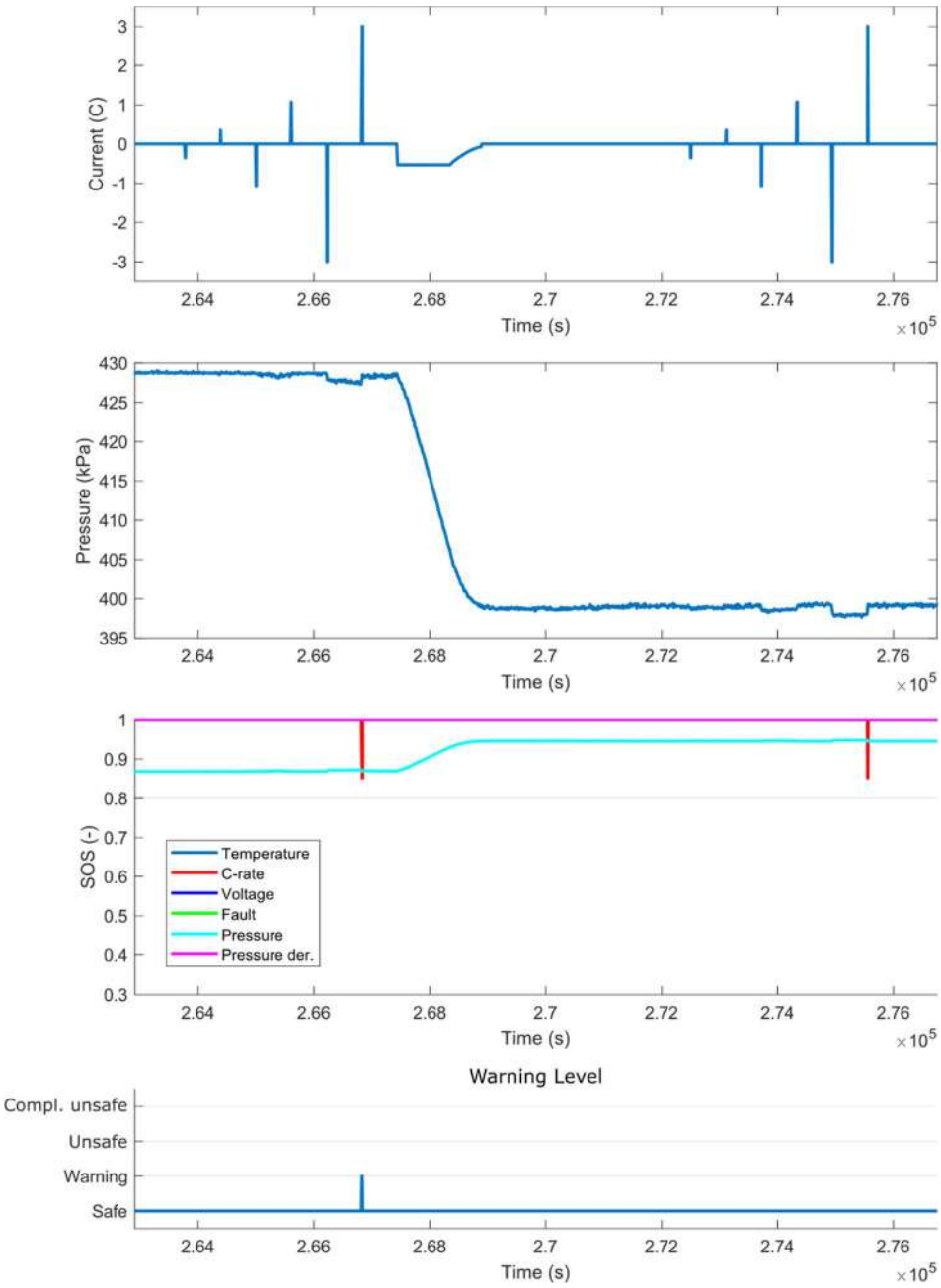

Figure 39: Zoom at charge pulses, 3C: Current, Pressure, SOS by terms, and warning level.

The pressure evolution during the pulses is the same as in Performance Test #1: there is not significant pressure increase when the 3C pulses are executed, due to the short length of these pulses (around 20 seconds). What is more, the main difference in pressure is produced by the discharge of the cell that happens between the two sort of pulses: as the battery is a lower state of charge, the overall pressure is lower in the second set of pulses.

During the 3C pulses, the current-related SOS term falls below the 100%, but as in the previous performance test, it does not cross the 80% threshold. However, in this case, the absolute pressure is high, and therefore, during both pulses the pressure-related SOS is below 100%. As previously mentioned, in the second set of pulses the cell is at a lower state of charge, and therefore the pressure-related SOS term is at a higher level. During the first pulse at 3C, both the current-related SOS and the pressure-related SOS are close to the 80% threshold. Consequently, the overall SOS falls below the 80%, triggering a warning alarm. This is the main difference compared to Performance Test #1.

In short, the main conclusion of this test is that, due to the addition of the relative pressure and pressure derivative terms, the SOS triggers more alarms as the cell is more aged. Indeed, as the cell ages, its relative pressure increases, what affects the pressure-related SOS term. And eventually, this affects in the overall SOS. In this context, it is worth to mention that the pressure thresholds (x_{100} and x_{80}) have been defined based on the data from the ageing tests, just to maintain, even close to the end of life, the pressure SOS inside the safe area. In the future, further experiments would be required to better define the pressure safe limits.

5 Discussion & Conclusion

This final section of the deliverable reviews the main findings and discussions that emerged during the development of the advanced state algorithms. The conclusions are reviewed for the SOC, SOH and SOS algorithms, both for baseline and L1 versions. Considering that the proposed SOE and SOP algorithms are related to the SOC algorithm, they are kept out of the current section.

5.1 Baseline SOC algorithm

The baseline SOC algorithm is based on an Adaptive-Extended Kalman Filter (AEKF). The proposed AEKF algorithm estimates the SOC and the overpotential states of a battery. The validation was performed with Dataset 1, a short data batch provided by TU/e for the baseline 5Ah SENSIBAT cell. The results showed a low RMSE for SOC estimation, which ranged between 0.5% and 1%. Besides, most of the estimations are in the 95% confidence ellipsoid. It was also found that the algorithm is robust to different initial conditions and can converge to the actual SOC even when starting with a large deviation from the actual SOC.

As mentioned, Dataset 1 included short experiments, where the capacity decreases slowly. Therefore, it is concluded that the SOC algorithm needs to the validated on a longer dataset. This has been carried out together with the development of the baseline SOH algorithm, whose conclusions are presented in the next subsection.

5.2 Baseline SOH algorithm

As a battery ages, its impedance increases and its capacity decreases. Hence, the model used for the SOC algorithm needs to be updated to account for these changes. Eventually, it is decided to tweak the baseline SOC algorithm, so it also accounts for model parameter adaption. Hence, the AEKF-based baseline SOC algorithm has been adapted into a Dual – Extended Kalman Filter (DEKF). The proposed SOH algorithm relies on the past data to obtain the real-time capacity. This solves a constrained optimization problem whose objective is to obtain the optimal capacity using the Weighted-Least-Square (WLS) method.

The algorithm has been validated on two different data batches (Dataset 3.1 and Dataset 3.2), which include the experimental aging performed in the SENSIBAT project to the 5Ah baseline cell. Dataset 3.1 only includes the tests to the baseline cells (with baseline measurements), while Dataset 3.2 includes measurements obtained with external pressure sensors applied to the baseline cells on the outside. The quality of SOH estimation has been validated on capacity checkup points available from the characterization cycles performed after each group of 50 charge-discharge cycles. On Dataset 3.1, considering only constant temperature of 25ºC, the SOH estimation algorithm yields 2.3% RMSE, while on Dataset 3.2 (considering only constant temperature of 25ºC), the algorithm yields 0.9% RMSE. When including the information of the 1D temperature measurements on Dataset 3.2, instead of assuming a constant temperature of 25ºC, the RMSE for SOH estimation is reduced from 0.9% to 0.83%. These results are also represented in Table 1.

Therefore, it is concluded that appropriate baseline SOC and SOH algorithms have been set, which provide a basis for the development of the L1-based state algorithms.

5.3 Baseline SOS algorithm

The proposed SOS algorithm is based on the concept of abuse. A modular expression has been presented for the estimation of the SOS, which is based on the aggregation of different safety functions. Each safety function is linked to a specific physical state of the battery. In the case of the baseline algorithm, safety functions related to the temperature, voltage and current have been proposed. Additionally, a function that focuses on the

detection of fault states has also been added to the SOS expression. The SOS expression outputs a value ranging from 0% to 100% relative to the overall safety state of the battery. However, due to the additional information provided by each of the individual safety functions, the algorithm is also able to provide warning levels or specific alarms, which give additional value to the proposed SOS algorithm.

The algorithm has been validated with the information from Dataset 2, which corresponds to the safety tests carried out in the SENSIBAT project. As it has been analysed, only overcharge tests were valuable for the validation of the SOS. Obviously, the validation showed that the term of the SOS that detects the overcharge works correctly, and since the cell crosses the maximum voltage, it detects that something is working wrong. However, apart from the validation of the overcharge detection, the validation tests demonstrated that even in a scenario where one of the sensors is not working correctly, the modular concept proposed for the SOS gives redundancy to the detection of non-safe states. In the specific case of the overcharge tests, it was demonstrated that the algorithm is also able to detect that something, such as the voltage sensor, is not working correctly by the detection of fault states or fast temperature increases.

On the one hand, regarding the temperature-related SOS term, the results have demonstrated that the algorithm is able to detect unsafe temperature increases prior to a thermal runaway event. In one of the scenarios, the thermal runaway event happened once the battery was above 60ºC, and in that case the algorithm triggered an alarm 5 minutes before the catastrophic event. In the other scenario, the thermal runaway event happened while the battery was still at 45ºC, but even in that case the SOS triggered an alarm 30 seconds before the catastrophic event.

On the other hand, regarding the detection of the fault state, the results demonstrated that the algorithm is sensible to the resolution and precision of the voltage and current sensors. Due to this reason, a tolerance was added, but this makes harder the detection of faults (they are only triggered for very short periods).

In any case, the results have validated the approach proposed for the baseline SOS. This has set a basis for the development of the L1-SOS algorithm.

5.4 L1-SOC algorithm

The DEKF algorithm proposed for the baseline SOC estimation has been extended for the L1 version of the algorithm. Among the different possibilities to integrate the 2D temperature and pressure measurements, it has been decided to opt for the method of averaging the 2D measurement data. Therefore, the average temperature value is used in the model, and additionally the pressure is added to the mentioned model.

Due to the unavailability of enough data to correctly parametrize the pressure effect in the battery model, the L1-SOC algorithm is conceived as a theoretical proposition of how the L1 sensors are able to improve the SOC estimation.

5.5 L1-SOH algorithm:

For the development of the L1-SOH algorithm, the first step consisted of analysing the available data from the SENSIBAT project, i.e., Dataset 3.2 (ageing tests to L1 cells, in fact baseline cells with externally applied one-dimensional pressure sensor). This analysis unveiled a significant relationship between mechanical pressure and SOC for distinct charge-discharge cycles at different SOH values. As the cell ages, there is an irreversible increase in thickness, leading to a shift in all pressure measurements towards higher values.

The proposed approach is based on obtaining a functional curve between capacity and averaged pressure measurements, in line with the single pressure value available in Dataset 3.2. The proposed L1-SOH algorithm

is also based on an optimization formulation, as the objective function previously presented for the baseline SOH algorithm is adapted. Indeed, the objective function balances the WLS term from the baseline algorithm with the deviation between the predicted and modelled capacity-pressure relationship.

The crucial factors in the proposed L1-SOH algorithm are the tuning parameters that decide which entity to focus on: the WLS route as used in the baseline algorithm that depends on past data or the modelled capacity-pressure relationship. The quality of the SOH estimation depends on these tuning parameters. Focusing more on modelled capacity-pressure relationship will work the best for the particular cell that have been used to obtain this relationship. However, for a completely different cell this will not be perfect since this relationship is biased. Therefore, it is recommended to choose tuning parameters to focus more on the WLS route, in order to avoid any bias.

Dataset 3.2 has been used to evaluate the quality of the L1-SOH estimation. Particularly, cell #3 has been used for this issue. Different tuning parameter scenarios have also been analysed. For instance, with ideal tuning parameters, RMSE values between 0.22% and 0.62% are obtained. Overall, the validation tests show that incorporating pressure information in SOH algorithm yields to improved RMSE, as opposed to using only baseline measurements. We saw improvement in RMSE from 0.83% to 0.62% or 0.11% RMSE using our proposed method for SOH estimation.

Additionally, the computation time for the entire dataset was also analysed, with one particular set of tuning parameters. The results showed that most of the computation time is below 0.5 seconds. And even when it is now below that value, it is still below the allotted time to estimate SOH. This shows the real-time deployment capability of the proposed algorithm for SOH estimation.

The SOH estimation algorithm needs to be validated on the dataset obtained from cell #4 for further validation with confidence. The RMSE for SOH estimation for cell #4 is 1.61% focussing only on the WLS route with historical measurements as in the baseline algorithm, while focussing only on the function derived from cell #3, the RMSE was reduced to 1.39%. Choosing a different tuning parameter for mixing the routes of WLS and the function derivative, we saw improvement in RMSE from 1.61% to 1.53%. This can only be improved further by improving the functional relationship. At the moment, we cannot improve it further, because we are limited to only two datasets with pressure measurements. An ideal methodology will be to obtain the functional relationship for at least two sources, so at least two datasets, and then validate on a third dataset. This will in return strengthen the function, reduces the chance of adding bias in our algorithm and eventually improve our SOH estimation algorithm for the L1 cell.

5.6 L1-SOS algorithm

The proposed L1-SOS algorithm is based on the SOS concept developed for the baseline version of the algorithm. Indeed, the proposed modular expression allows to easily add further safety functions as more physical measurements are available. In the case of the L1 cells, it was decided to add two safety functions related to the pressure: one for the relative pressure increase since beginning of life, and another one for the pressure derivative. The L1 version of the algorithm also integrates the information of the 2D matrix: an SOS value is defined for each spot of the matrix, in order to better identify the exact location of the battery were a critical temperature or pressure related state is happening.

For the validation of the L1-SOS algorithm, Dataset 3.2 has been used (ageing tests to L1 cells), due to the unavailability of L1 data from safety tests. During these tests, pressure and temperature were externally measured, and therefore only the 1D version of the L1-SOS algorithm was validated.

The results demonstrated the additional value given to the SOS by the addition of the pressure measurements. For instance, Dataset 3.2 included some charging patterns at 3C, above the recommended maximum charging of 2C. During these phases, no large temperature increase was noticed. However, the pressure increased fast, and after the charging phase was finished, an irreversible pressure step increase was identified. These patterns were identified by the L1-SOS, contrary to the baseline version of the SOS algorithm, thanks to the addition of the pressure derivative-related safety function. The analysis of the ageing results demonstrated that the 3C charging patterns accelerated the ageing of the battery. Therefore, the L1-SOS would be able to detect that this charging rate is not appropriate for the battery.

The validation of the L1-SOS algorithm also showed the added value of the absolute pressure-related safety function. When the battery is closer to end of life, this safety function is close to the safe threshold, and therefore it is easier for the SOS to trigger alarms. As it was already mentioned during the discussion of the validation results, it is important to point out that the pressure-related safety functions were parametrized with the same data used for validation, due to the unavailability of further data. In order to better define the thresholds of each of the proposed additional safety functions, it is recommended to collect further data that includes pressure measurements.

6 References

[1] "Sensibat Project, D1.1 - Requirement Specification (Use cases, KPIs and cell, module requirements)," 2021.

[2] "Sensibat Project, D1.2 - Testing plan for cells and modules," 2021.

[3] F. S. Hoekstra, M. Donkers and H. Bergveld, "Rapid empirical battery electromotive-force and overpotential modelling using input–output linear parameter-varying methods," *Journal of Energy Sources,* vol. 65, p. 107185, 2023.

[4] "Sensibat Project, D5.1 - Test report on cell and module performance and safety," 2023.

[5] Z. Ren and D. Changqinp, "A review of machine learning state-of-charge and state-of-health estimation algorithms for lithium-ion batteries.," *Energy Reports,* vol. 9, pp. 2993-3021, 2023.

[6] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation," *Journal of Power Sources,* vol. 134, no. 2, pp. 277-292, 2004.

[7] P. den Boef, P. B. Cox and R. Tóth, "LPVcore: Matlab toolbox for LPV modelling, identification and control," *IFAC-PapersOnLine,* vol. 54, no. 7, pp. 385-390, 2021.

[8] H. Beelen, H. Bergveld and M. Donkers, "Beelen, H., Bergveld, H. J., & Donkers, M. C. F. (2020). Joint estimation of battery parameters and state of charge using an extended Kalman filter: A single-parameter tuning approach.," *IEEE Transactions on Control Systems Technology,* vol. 29, no. 3, pp. 1087-1101, 2020.

[9] Y. Zhang, X. Rui, H. Hongwen and S. Weixiang, "Lithium-ion battery pack state of charge and state of energy estimation algorithms using a hardware-in-the-loop validation.," *IEEE Transactions on Power Electronics,* vol. 32, no. 6, pp. 4421-4431, 2016.

[10] T. Zhang , G. Ninguan, S. Xiaoxia, F. Jie, Y. Naifeng, S. Junjie and Z. Yuan , "A systematic framework for state of charge, state of health and state of power co-estimation of lithium-ion battery in electric vehicles.," *Sustainability,* vol. 13, no. 9, p. 5166, 2021.

[11] E. Cabrera-Castillo, F. Niedermeier and A. Jossen, "Calculation of the state of safety (SOS) for lithium ion batteries," *Journal of Power Sources,* pp. 509-520, 2016.

[12] A. Samanta and S. S. Williamson, "Samanta, Akash, and Sheldon S. Williamson. "A comprehensive review of lithium-ion cell temperature estimation techniques applicable to health-conscious fast charging and smart battery management systems," *Energies,* vol. 14, no. 18, p. 5960, 2021.

[13] K. E. Thomas, J. Newman and R. M. Darling, "Mathematical modeling of lithium batteries," *Advances in lithium-ion batteries,* Vols. -, no. -, pp. 345-392, 2002.

[14] M. Yazdanpour, T. Peyman and M. Abraham , "A distributed analytical electro-thermal model for pouch-type lithium-ion batteries," *Journal of the electrochemical society,* vol. 161, no. 14, p. A1953, 2014.

[15] R. Fu, X. Meng and C. Song-Yul, "Modeling, validation and analysis of mechanical stress generation and dimension changes of a pouch type high power Li-ion battery," *Journal of Power Sources,* vol. 224, no. -, pp. 211-224, 2013.

[16] J. B. Siegel, A. G. Stefanopoulou, P. Hagans, Y. Ding and D. Gorsich, "Expansion of lithium ion pouch cell batteries: observations from neutron imaging.," *Journal of the Electrochemical Society,* vol. 160, no. 8, p. A1031, 2013.

[17] J. Cannarella and C. B. Arnold, "State of health and charge measurements in lithium-ion batteries using mechanical stress," *Journal of Power Sources,* vol. 269, no. -, pp. 7-14, 2014.

[18] B. D. Anderson and J. B. Moore, Optimal filtering, -: Courier Corporation, 2012.

[19] C. T. Fraser and S. Ulrich, "Adaptive extended Kalman filtering strategies for spacecraft formation relative navigation.," *Acta Astronautica,* vol. 178, no. -, pp. 700-721, 2021.

7 Acknowledgement

The author(s) would like to thank the partners in the project for their valuable comments on previous drafts and for performing the review.

Project partners

#	PARTICIPANT SHORT NAME	PARTNER ORGANISATION NAME	COUNTRY
1	IKE	IKERLAN S. COOP.	Spain
2	BDM	BEDIMENSIONAL SPA	Italy
3	POL	POLITECNICO DI TORINO	Italy
4	FHG	FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V.	Germany
5	FM	FLANDERS MAKE VZW	Belgium
6	TUE	TECHNISCHE UNIVERSITEIT EINDHOVEN	The Netherlands
7	NXP NL	NXP SEMICONDUCTORS NETHERLANDS BV	The Netherlands
8	NXP FR	NXP SEMICONDUCTORS FRANCE SAS	France
9	ABEE	AVESTA BATTERY & ENERGY ENGINEERING	Belgium
10	VAR	VARTA MICRO INNOVATION GMBH	Germany
11	AIT	AIT AUSTRIAN INSTITUTE OF TECHNOLOGY GMBH	Austria
12	UNR	UNIRESEARCH BV	The Netherlands

DISCLAIMER/ ACKNOWLEDGMENT

Copyright ©, all rights reserved. This document or any part thereof may not be made public or disclosed, copied, or otherwise reproduced or used in any form or by any means, without prior permission in writing from the SENSIBAT Consortium. Neither the SENSIBAT Consortium nor any of its members, their officers, employees or agents shall be liable or responsible, in negligence or otherwise, for any loss, damage or expense whatever sustained by any person as a result of the use, in any manner or form, of any knowledge, information or data contained in this document, or due to any inaccuracy, omission or error therein contained.

All Intellectual Property Rights, know-how and information provided by and/or arising from this document, such as designs, documentation, as well as preparatory material in that regard, is and shall remain the exclusive property of the SENSIBAT Consortium and any of its members or its licensors. Nothing contained in this document shall give, or shall be construed as giving, any right, title, ownership, interest, license, or any other right in or to any IP, know-how and information.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957273. The information and views set out in this publication does not necessarily reflect the official opinion of the European Commission. Neither the European Union institutions and bodies nor any person acting on their behalf, may be held responsible for the use which may be made of the information contained therein.

Annex A – EKF Algorithm

The two-step EKF [8] and [18] is divided into:

1. a measurement update as shown below:

$$L_k = P_k^+ C_k^T \left(C_k P_k^+ C_k^T + R_k\right)^{-1}$$
$$v_k = (y_k - C_k\, \hat{x}_k^+ - D_k u_k)$$
$$\hat{x}_{k+1}^- = \hat{x}_k^+ + L_k v_k$$
$$P_{k+1}^- = (I - L_k C_k) P_k^+$$

2. the time update, along with the update of the process covariance matrix [19], which is shown as:

$$v_{k+1} = (y_k - C_k x_{k+1}^- - D_k u_k)$$
$$x_{k+1}^+ = A_k\, \hat{x}_{k+1}^- + B_k u_k + S_k R_k^{-1} v_{k+1}$$
$$Q_k = L_k \left[\frac{1}{N}\sum_{i=1}^{M} v_i v_i^T\right] L_k^T$$
$$P_{k+1}^+ = (A_k - S_k R_k^{-1} C_k) P_{k+1}^- (A_k - S_k R_k^- C_k)^T + Q_k - S_k R_k^{-1} S_k^T$$

where Q_k and R_k denote the adaptive time-varying process-noise covariance and measurement-noise covariance, respectively, x_k represents the states, P represents the prediction matrix and L is known as the optimal Kalman gain. The state-transition matrix A_k and the observation matrix C_k are formulated as:

$$A_k = \frac{\delta f(x_k, u_k)}{\delta x_k} = \begin{bmatrix} 1 & 0 \\ 0 & \theta_k^1 \end{bmatrix}, C_k = \frac{\delta g(x_k, u_k)}{\delta x_x} = [dEMF(SoC_k) \quad 1]$$

Annex B – DEKF Algorithm

The two-step DEKF with forgetting factor [8] is shown as below:

1. a measurement update is shown below:

$$L_k = P_k^+ C_k^T \left(C_k P_k^+ C_k^T + R_k\right)^{-1}$$
$$\hat{x}_{k+1}^- = \hat{x}_k^+ + L_k(y_k - C_k\, \hat{x}_k^+ - D_k u_k$$
$$P_{k+1}^- = \frac{1}{\gamma}(I - L_k C_k) P_k^+$$

2. Q_k, R_k and S_k, as suggested in [8], which retains the adaptive covariance matrices and the time update, is shown as:

$$Q_k = \begin{bmatrix} \frac{1}{Q_{cap}^2} & \frac{\theta_2}{Q_{cap}} & 0 & 0 \\ \frac{\theta_2}{Q_{cap}} & \theta_2^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$R_k = 1 + \theta_3^2$$
$$S_k = \begin{bmatrix} \frac{\theta_3}{Q_{cap}} \\ \theta_2 \theta_3 \\ 0 \\ 0 \end{bmatrix}$$
$$x_{k+1}^+ = A_k\, \hat{x}_{k+1}^- + B_k u_k + S_k R_k^{-1}(y_k - C_k x_{k+1}^- - D_k u_k)$$
$$P_{k+1}^+ = (A_k - S_k R_k^{-1} C_k) P_{k+1}^- (A_k - S_k R_k^- C_k)^T + Q_k - S_k R_k^{-1} S_k^T$$

Notice, the tuning parameter γ, which is chosen as a constant, $\gamma = 0.9999$. The state-transition matrix A_k and the observation matrix C_k are formulated as:

$$A_k = \frac{\delta f(x_k, u_k)}{\delta x_k} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \theta_k^1 & u_k & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, C_k = \frac{\delta g(x_k, u_k)}{\delta x_x} = [dEMF(SOC_k) \quad 1 \quad 0 \quad u_k]$$