38th International Electric Vehicle Symposium and Exhibition (EVS38) Göteborg, Sweden, June 15-18, 2025

Advanced State-of-X Estimators Applied for Heterogeneous Batteries

Juan Alberto Romero Baena[1], Norbert Hevele[1], Corneliu Barbu[2], Farshid Naseri[2]

[1]*Schaeffler Group, Germany, juan.alberto.romero@vitesco.com; Norbert.hevele@vitesco.com*

[2]*coba@ece.au.dk; fna@ece.au.dk*

Executive Summary

Integrating modular, scalable battery management systems introduces an additional layer of complexity, especially when managing applications with heterogeneous cell chemistries. To address this, this paper proposes an effective architecture of state-of-X (SoX) estimators for monitoring two different battery chemistries: Nickel Manganese Cobalt Oxide (NMC) and Lithium Titanate Oxide (LTO). The SoX algorithms—based on Extended Kalman Filter and least squares estimation—are developed, simulated, implemented, and tested to estimate state-of-charge (SoC), state-of-health (SoH), state-of-power (SoP), and state-of-energy (SoE). These algorithms are implemented on an automotive-grade series production microcontroller. The software is designed to be scalable and reconfigurable, capable of handling up to 512 cells, individually. The paper presents and discusses the algorithms' accuracy, real-time computational load, and memory requirements. The results confirm the effectiveness of the proposed structure in managing the complexities associated with mixed-cell battery packs and the challenges posed by large-scale cell integration.

Keywords: Electric Vehicles, Energy Storage Systems, Battery Management System, Modelling & Simulation, Energy Management

1 Scalable and modular heterogeneous batteries

The application of different battery cell chemistries has been increasingly studied in recent years as a strategy to optimize production and leverage economies of scale. This approach enables the use of standardized battery module housings—with identical external dimensions—equipped with different cell chemistries [1]. Introducing such a novel degree of flexibility allows for better optimization of the battery pack's gravimetric and volumetric power and energy densities.

Several concepts have been explored, including the integration of power electronics at the cell or module level. While these solutions offer flexibility, they introduce efficiency losses due to multiple energy conversion stages and lead to increased bill-of-materials (BOM) costs. To balance flexibility, scalability, and modularity, the battery pack architecture studied in the HELIOS project is adopted [2]. This architecture minimizes the use of power electronics while enabling energy exchange between two cell chemistries: Lithium Titanate Oxide (LTO) cells, which enhance the power density of the pack and support high current peaks to the high-voltage traction network; and Nickel Manganese Cobalt Oxide (NMC) cells, which contribute to energy density and support extended driving range. This modular concept facilitates adaptation to different applications and requirements using same-sized battery modules - from passenger cars and light electric mobility to electric bus applications. Fig. 1 shows an overview of the HELIOS battery pack composed of the NMC-based and LTO-based subpacks connected using a DC-DC converter. Current Sensing Monitoring (CSM) device measures high voltage across the external contactors, and current flowing into/out the battery pack, as well as tracks the high voltage insulation of the battery. Cell Supervision Controllers (CSCs) are placed inside each module whose data is then communicated to the

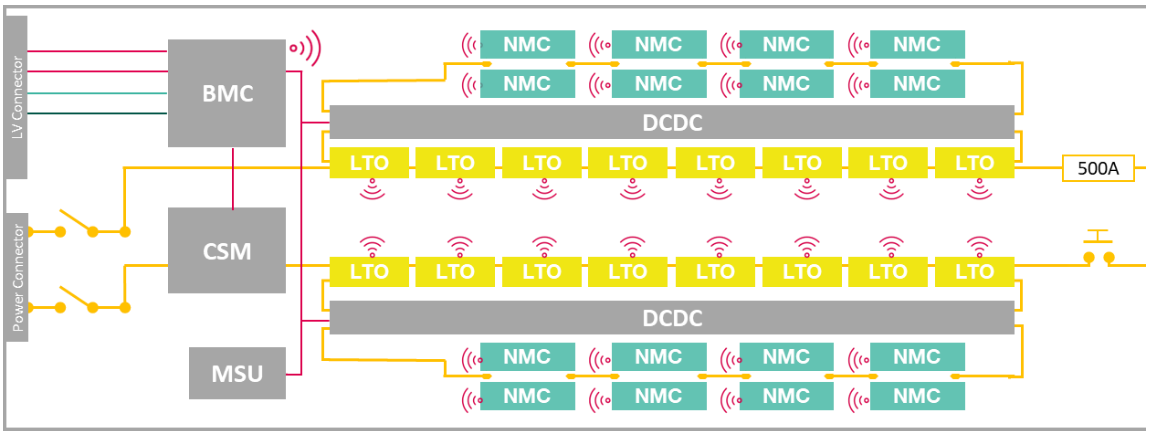


Figure 1: Hybrid heterogeneous battery pack based on LTO and NMC battery modules

central Battery Management Controller (BMC), wirelessly. The CSCs are designed to measure and transmit voltages and temperatures of 16 series connected cells considering both high-energy (HE) and high-power (HP) modules with NMC and LTO cells, respectively. The State-of-X (SoX) algorithms are integrated into the BMC. The multi-sensing unit (MSU) is responsible for measuring and monitoring other critical values in the battery pack including the inner pressure, level of critical gases, and temperature. The key challenge in this context lies in the integrated monitoring of a large number of cells. The HELIOS battery pack for the e-bus demonstrator consists of 4 HE sub-packs, each containing 4 HE modules configured as 16s1p (16 cells per module connected in series), and 4 HP sub-packs, each containing 4 HP modules configured as 16s2p (16 series cells per module). This results in a total of 512 individual cell voltages to be monitored using the CSCs. This introduces a twofold challenge.

- First, there is the algorithmic complexity of adapting state estimation algorithms to heterogeneous cell chemistries. NMC and LTO cells exhibit different nonlinear characteristics and dynamic behaviors, which must be accurately captured and accounted for in a unified SoX software framework to ensure reliable estimation of SoC, SoH, SoP, and SoE.
- The second challenge is the management of state estimation for 512 individual cells in real-time. Performing SoC, SoH, and SoP calculations at the cell level introduces embedded system challenges, including processing limitations and memory constraints. Therefore, the BMS software must be carefully designed to be computationally efficient, scalable, and capable of real-time performance across many heterogeneous cells.

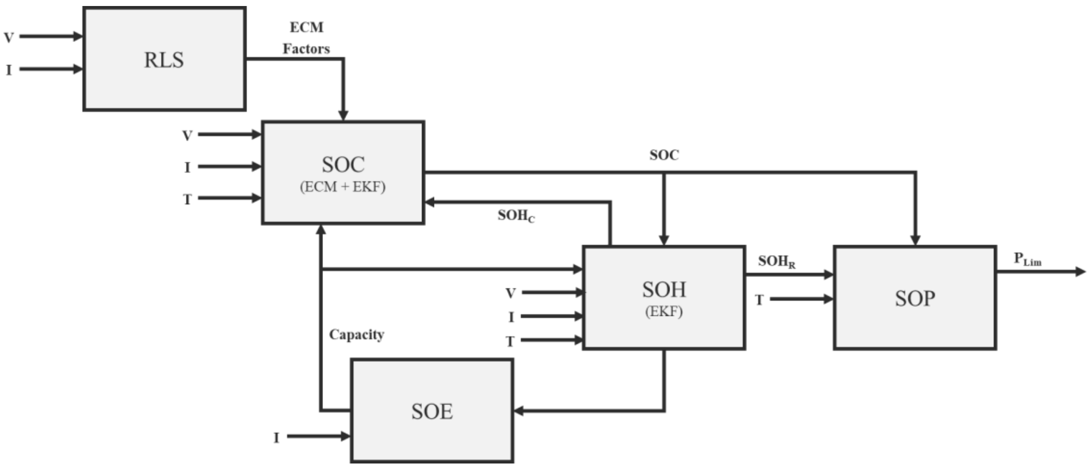
These challenges are rarely addressed in literature, as most battery pack designs assume a single-cell chemistry. This paper presents the implementation details and performance evaluation of a dual-chemistry BMS software developed for the HELIOS project. The rest of this paper is organized as follows: in Section 2, the design of state estimators and algorithmic details are presented. Section 3 presents the implementation steps of the algorithms into the automotive embedded system. The test results are presented and discussed in Section 4. Conclusions and some remarks about future steps are included in Section 5.

2 State estimators for heterogeneous battery packs

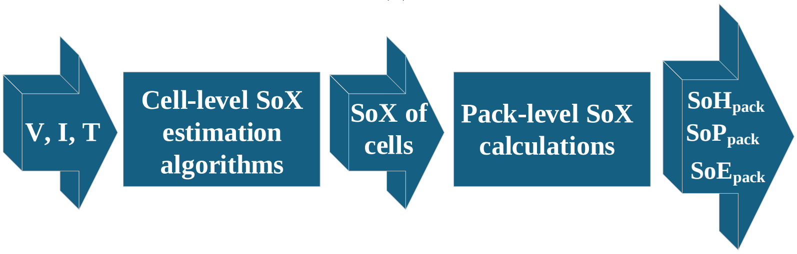
An overview of the SoX algorithm interfaces is shown in Fig. 2. The SoX algorithms are designed to use battery measurements—current, voltage, and temperature—as inputs. As illustrated, the implementation is carried out at the cell level, where individual state estimations are performed. These cell-level estimations are then aggregated to derive the corresponding states at the pack level. All algorithms are implemented in the discrete domain to facilitate efficient digital deployment. The descriptions of the SoX algorithms are presented in the following subsections. Due to the broad scope of the SoX framework, the algorithms are described in a condensed format to keep the paper concise.

2.1 SoC Estimation

The SoC estimation is based on the well-known Extended Kalman Filter (EKF) algorithm and is fulfilled



(a)



(b)

Figure 2: (a) Signal flow diagram of state estimators based on measured and calculated signals (b) Pack-level SoX estimations are calculated from cell-level estimations

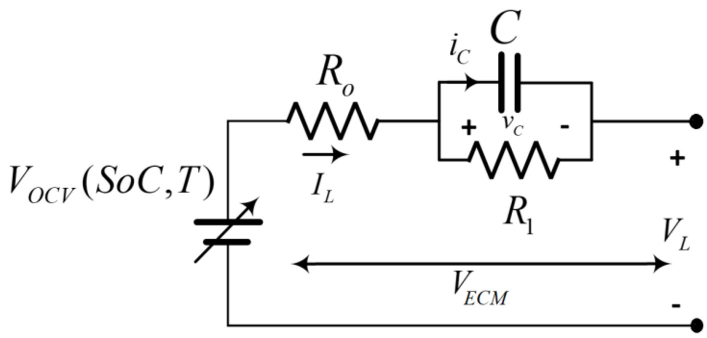


Figure 3: Thevenin-based equivalent circuit model of the battery

as a number from 0 (fully discharge state) to 1 (fully charge state). The EKF relies on a state-space model derived from the widely used Thevenin equivalent circuit model (ECM), where the dynamic behavior of the battery cell is represented using electrical circuit elements as depicted in Fig. 3. The state-space model of the battery is formulated as in (1)-(2).

$$\begin{bmatrix} SOC(k) \\ V_{ECM}(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & k_0 \end{bmatrix} \begin{bmatrix} SOC(k-1) \\ V_{ECM}(k-1) \end{bmatrix} + \begin{bmatrix} \dfrac{-T_s\eta}{Q \times 3600} & 0 \\ k_1 & k_2 \end{bmatrix} \begin{bmatrix} I_L(k) \\ I_L(k-1) \end{bmatrix} \tag{1}$$

$$V_L(k) = V_{OCV}(k) + V_{ECM}(k) \tag{2}$$

Here, k is the sample index, T_s denotes the sampling time, η is the charge efficiency, Q represents the cell capacity, I_L denotes the cell current, and V_{ECM} is the voltage across the dynamic section of the ECM. V_L denotes the cell terminal voltage, while V_{OCV} refers to the open-circuit voltage. The parameters k_0 to k_2 are model coefficients and they can be mathematically derived from the ECM's circuit parameters R_0, R_1, and C.

$$k_0 = \frac{2R_1C - T_s}{2R_1C + T_s} \tag{3}$$

$$k_1 = \frac{T_s\left(R_0 + R_1\right) + 2R_0R_1C}{2R_1C + T_s} \tag{4}$$

$$k_2 = \frac{T_s\left(R_0 + R_1\right) - 2R_0R_1C}{2R_1C + T_s} \tag{5}$$

This state-space model is employed within the EKF framework to estimate the SoC. The detailed equations of the EKF are well-established in the literature and are therefore omitted here [3]. To enhance estimation performance, the covariance matrices—representing process and measurement noise—are optimized using a genetic algorithm. The covariance matrices have a crucial role in the EKF algorithm by determining the uncertainties lying within the cell model and measurement noises. The entries of these matrices are treated as optimization variables, and the cost function is defined to minimize the root mean square error (RMSE) of the SoC estimation. Model parameterization is performed separately for NMC and LTO cells, and the corresponding parameters are provided to the algorithm through dedicated lookup tables. This includes model parameters related to the OCV-SoC characteristics curves (V_{OCV}), entries of covariance matrices, etc. To reduce computational load, the SoC estimation block is applied iteratively across different cells rather than in parallel.

2.2 RLS algorithm

In practice, the parameters of the ECM change over time due to battery degradation and varying operating conditions. Since the cell model is directly embedded in the EKF-based SoC estimation, these parameter variations can introduce modeling errors, leading to a reduction in estimation accuracy. To resolve this, a Recursive Least Squares (RLS) algorithm is employed to continuously estimate and update the ECM parameters used within the EKF framework. The RLS algorithm is formulated based on the following regressor equation, derived from the Thevenin-based ECM under the assumption that:$V_{OCV}(k) \sim V_{OCV}(k-1)$ (which is reasonably valid since we don't expect SoC to change too much within Ts).

$$\underbrace{V_L(k)}_{y} = \underbrace{\begin{bmatrix} 1 & V_L(k-1) & I(k) & I(k-1) \end{bmatrix}}_{U} \times \underbrace{\begin{bmatrix} (1-k_0)V_{OCV} \\ k_0 \\ k_1 \\ k_2 \end{bmatrix}}_{\theta} \tag{6}$$

where y is the output (terminal voltage), U is the regressor, and θ is the vector of unknown parameters to be estimated. The regressor can be built from the sensor values (values at the current sample and the previous sample). Upon formulation of the regressor, the RLS algorithm can be applied to estimate the unknown parameter vector, recursively.

2.3 SoH estimation algorithm

The cell capacity and its internal resistance are the most used indices to characterize the health of the battery. The capacity of a battery indicates its energy capability while the battery's internal resistance determines the battery's power capability. Both are important to ensure the effective operation of the EV to meet the driving range and acceleration requirements. The SoH is defined as a number from 1 (new cell) to 0 (end of first life) using the below equation:

$$SoH_X = 1 - \frac{(1 - X_{actual} / X_{rated})}{(1 - EoL_X)} \tag{7}$$

where X could be the capacity Q or the internal resistance R_0. Likewise, X_{actual} denotes the actual value of Q or R_0, X_{rated} denotes the nominal value, and EoL_X denotes the related EoL criteria. Generally, the EoL will be reached for a capacity drop of 20% compared to the fresh battery or when internal resistance increases by 150% compared to the fresh cell, whichever is met first. In HELIOS, both health indicators are estimated using the algorithms described in the following subsections.

2.3.1 Internal resistance estimation

The internal resistance of the battery increases with battery aging. HELIOS deploys a simple algorithm that can estimate the internal resistance based on the available measurements at any given time (battery current, voltage, and temperature). We consider the so-called R-int cell model composed of the V_{OCV} term and the DC internal resistance R_0 leading to $V_L = V_{OCV} - R_0 I = [1 \quad -I][V_{OCV} \quad R_0]^T$. Considering $U' = [1 \quad -I]$ and $\theta' = [V_{OCV} \quad R_0]^T$, the parameter vector is estimated using the sliding window LS algorithm leading to estimations of the DC internal resistance.

2.3.2 Capacity estimation algorithm

To estimate the capacity, we consider the well-known Coulomb equation for batteries as follows:

$$SoC_{k2} = SoC_{k1} - \frac{\sum_{k=k_1}^{k_2-1} \eta_k i_k}{Q \times 3600} \tag{8}$$

The above equation can be used to predict the battery SoC (up to the sample $(k_2 - 1)$) with the assumption of known battery SoC at the sample k_1. We can rewrite (8) as follows:

$$\underbrace{(SoC_{k2} - SoC_{k1}) \times 3600}_{x} \times \underset{\theta}{Q} = \underbrace{-\sum_{k=k_1}^{k_2-1} \eta_k i_k}_{y} \tag{9}$$

Equation (9) has a regression form $y = \theta x$, allowing battery capacity estimation via simple methods like LS, using stored SoC variation and accumulated Ah data. However, accurate capacity estimation relies on precise SoC values, typically with errors below 2%. Since EKF-based SoC estimation used here yields RMSE below 2%, this requirement is met. Traditional LS assumes noise-free inputs, which isn't realistic, as both SoC and current measurements (i.e., x and y) are affected by noise and estimation errors. This can introduce bias in capacity estimation. One workaround is to use SoC values from OCV measurements during rest periods (I_L= 0), but this is limited to offline conditions. To address this more robustly, a Total Least Squares (TLS) method is adopted in this work, which accounts for noise in both input and output, offering a more accurate and realistic capacity estimation framework [4].

2.4 SoP Estimation

The purpose of the SoP estimation algorithm is to determine the maximum permissible power output or input of the battery within a defined time horizon. This ensures that the battery is not charged or discharged beyond safe operational limits, particularly preventing premature violations of voltage thresholds due to transient current spikes—such as those occurring during rapid acceleration. This is especially critical near the SoC limits to guarantee safe operation under high-load conditions. To achieve this, a standard approach is employed that considers constraints related to charge/discharge current, SoC, and terminal voltage. The implementation approach follows the methodology described in [5]. For SoP estimation, other state estimates including SoC, SoH_R (SoH based on resistance), and SoH_C (SoH based on cell's capacity) are considered as inputs. The SoP is estimated in kW of available charge or discharge power.

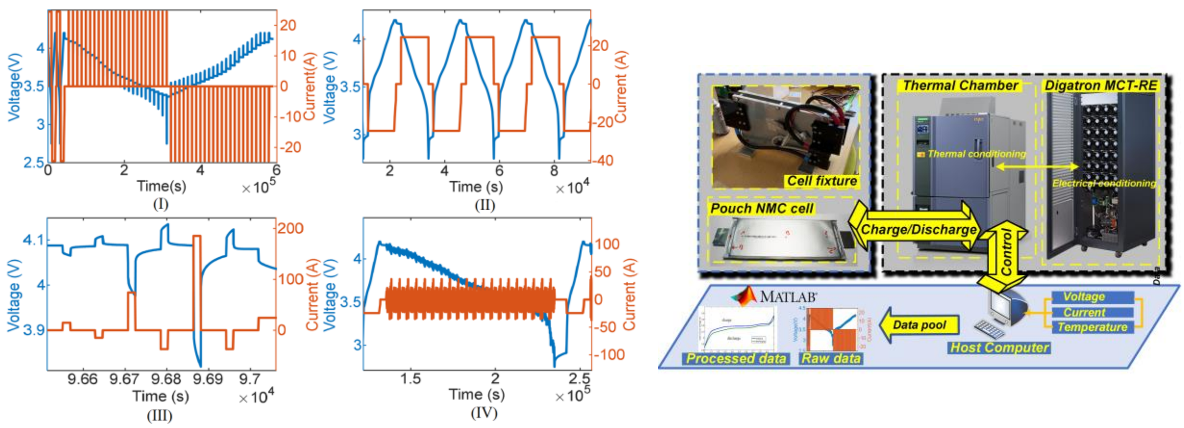
2.5 SoE Estimation

The SoE refers to the remaining Ah in the cell. The SoE can be estimated in Ah from the estimated values of SoC, SoH, and the cell's actual capacity.

Upon completing the cell-level estimations, the pack-level state estimates are aquired. For HP modules configured with two parallel cell strings (16s2p), each pair of parallel cells is treated as a single equivalent unit for estimation purposes. The overall pack performance is constrained by the weakest cell, as it determines the pack's safe operating limits. Furthermore, the HELIOS battery pack includes two distinct types of sub-packs: HE and HP. These sub-packs differ not only in cell chemistry (NMC versus LTO) but also in their dynamic behavior and aging characteristics, adding another layer of complexity to the aggregation of SoX estimations at the pack level. The SoC for the hybrid pack is obtained as follows:

$$SoC = \frac{SoC_{HE}C_{HE} + SoC_{HP}C_{HP}}{C_{HE} + C_{HP}} \tag{10}$$

To accommodate the distinct behaviors of LTO and NMC cells, separate experimental characterizations are conducted. These include capacity tests to determine the reference capacities, Hybrid Pulse Power Characterization (HPPC) tests to identify model parameters, OCV characterization tests to establish the OCV-SoC relationship, and driving cycle tests for model validation. The model design, order selection, and parameterization are carefully tailored to accurately reflect the behavior of each cell type. Additionally, the use of the RLS algorithm ensures that model parameters are individually updated for each cell during operation, allowing for real-time adaptation to aging and changing conditions. Figure 4 illustrates the typical Reference Performance Tests (RPTs) used for model parameterization.



(a) (b)

Figure 4: (a) Cell RPTs carried out for separate parameterization of SoX for HE and HP cells (b) Experimental setup used for data collection

3 Implementation into automotive embedded system

Development in automotive industry follows Automotive SPICE® [6] following Software Engineering Process Group (SWE). As part of this process, an architecture tool for software architectural design (SWE.2) is used for generation of an AUTOSAR container [7], which will be synchronized in the software detailed design and unit construction (SWE.3) and software unit verification (SWE.4) under MATLAB/Simulink® environment, tool used for the implementation of the algorithms. Once simulations

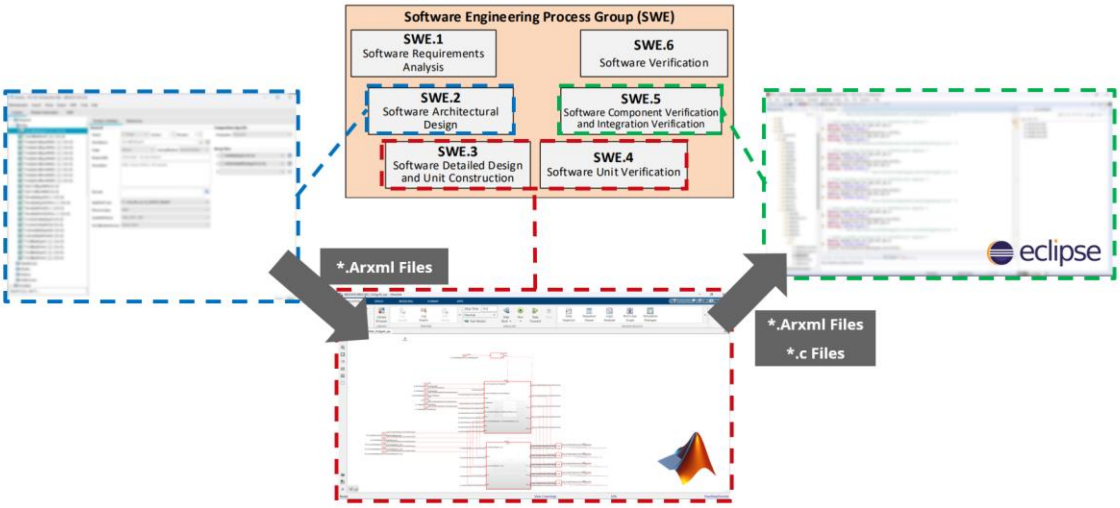


Figure 5: AUTOSAR arxml file (left) imported by MATLAB/Simulink® (middle) used for code generation and integration into software project (right)

are validated, code is automatically generated following AUTOSAR settings. Software is designed in such way, that all relevant variables are scalable, allowing the use of the same code automatically generated for all possible configurable configurations within the system, reducing the development effort by reusing other scenarios. During autocode generation, first static checks for code quality are usually performed, as e.g. linting the code following MISRA rules [8], as well unitary tests are performed, testing the model in Model-In-Loop (MIL), the code as Software-In-Loop (SIL), as well as Processor-In-Loop (PIL).

In software component verification and integration verification (SWE.5), software is compiled under project environment tool, and specifically configured for tree applications, demonstrating the scalability and modularity of the designed software architecture. Three scenarios are selected for this investigation: HIL test setup consisting of 2 HP and 2 HE modules; a passenger car size, consisting in 8 HP and 4 HE modules; and maximum size for e-bus application, consisting in 16 HP and 16 HE modules.

These three scenarios are being compiled, calibrated and tested in the following software testbench shown at Figure 6. In such scenario, the number of CSCs required for the setup can be added and removed easily, allowing a quick validation phase and dynamic architecture validation of the SoX algorithms, without needing real cells. Such software testbench can be further developed soon as a HIL testbench, adding battery ell simulators and high voltage and current signals, achieving a complete simulation of a battery system.

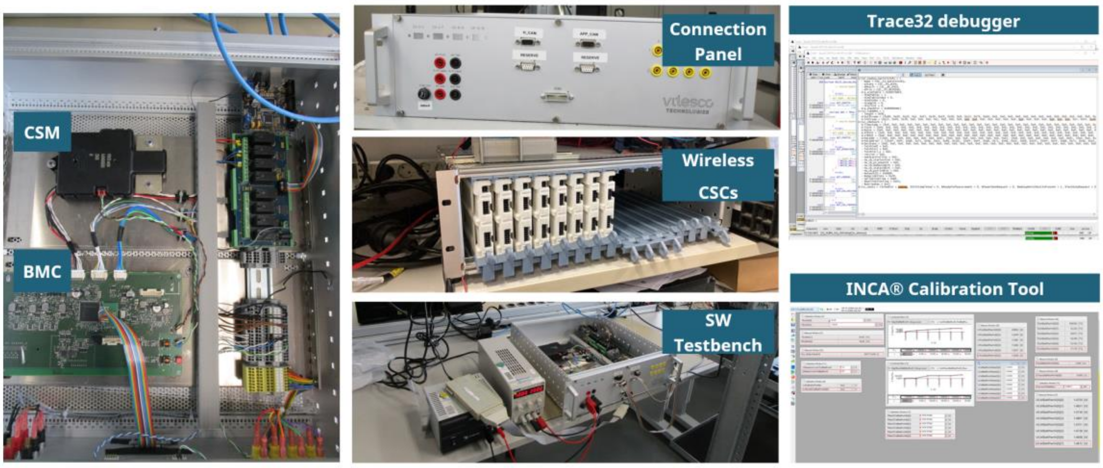


Figure 6: BMS system tested on Software Testbench for static and dynamic validation of algorithms.

4 Results and discussions

4.1 Static and Dynamic Performance Measurements

Software implemented for handling the heterogeneous battery packs is kept scalable and modular until the final configuration is applied and compilation for the application is started. Different parameterization files can be tracked separately for the different final applications, reducing the effort of handling several software projects in parallel. Additionally, additional instrumentation software is enabled during compilation time to facilitate the measurements of the dynamic behavior of the SoX algorithms within the embedded platform.

As a static key performance indicator (KPI) for the state estimator features embedded, software is measured in terms of RAM, ROM and calibration needs. Target is to detect in advance any potential issue in terms of resource capabilities of the selected target embedded platform microcontroller. Usually, a series approach application tends to define memory budgets for the allocation of the features. In following Figure 7, a presentation of the measured performance of different parts of the state estimators is shown, indicating the memory resource needs.

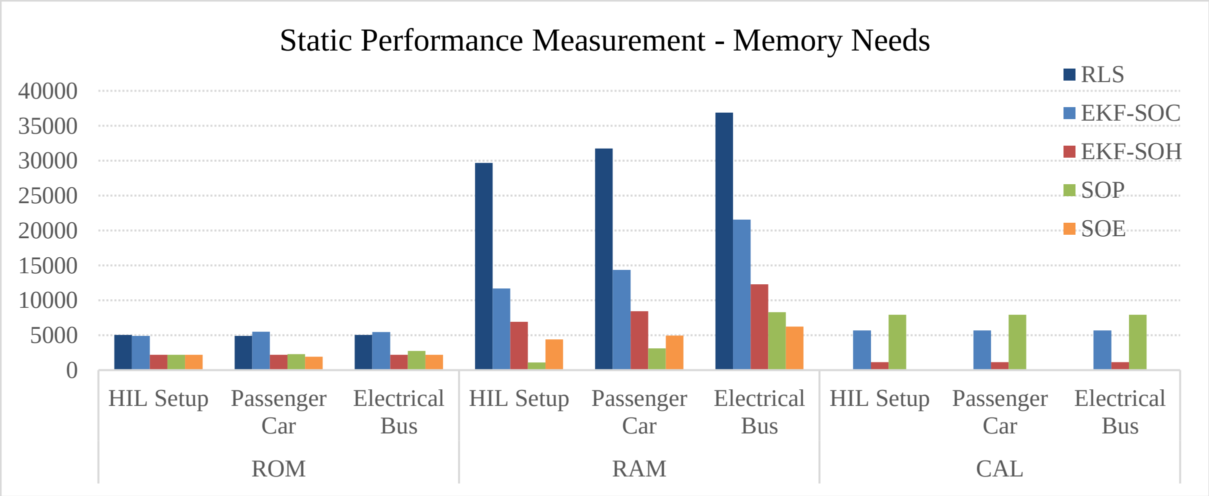


Figure 7 ROM/RAM/Calibration memory for different heterogeneous battery configurations with same software [Bytes]

It is observed that while the increase in the number of cells does not proportionally impact ROM or calibration memory, it does affect RAM usage. As the core algorithms are similar for both chemistries, and different performance behavior on cells for the SoX will be reflected during calibration time evaluation. It can be pre-located for each of the SoX algorithms a memory budget in the next future, answering the static memory needs when scaling the concept to a desired target platform in advance. This enables the best cost decision making approach for the embedded platform microcontroller, capable of covering these algorithm needs.

As a dynamic KPI for the state estimator features, software is measured under real operation at software testbench, measuring the minimum, average and maximum timing execution, as well as the overall execution time of each of the microcontroller's cores. Target is to detect any potential execution performance in embedded platform, which could add the risk of losing the real time execution of the microcontroller, essential for an embedded platform under a real time environment (RTE). Any performance issue which might affect the normal behavior of a safe critical system BMS, classified as ASIL C(D), is essential to be avoided and is preliminary step to test the hardware with real battery cells.

The test is defined as follows: it will be measured at least 32 iterations of the control function, programming a time out for protection about 10 % of the expected calculation time. SoX Features are implemented as a 1 second periodic task. Tests will be performed automatically and executed for all execution tasks within the SoX feature set. To ensure real-time operation of the embedded software, 300 us are set as a default limit window for triggering a warning and 600 us are defined as a critical threshold in execution. If any of the algorithms exceed this execution times, it shall be observed the overall microcontroller load, evaluating whether a real risk of loosing real time execution is present.

In the following Table 1, measurements are presented for each of the relevant SoX features. It is to be noticed that the differences of the configuration of the battery packs (HIL, passenger car, e-bus) reflect directly in the execution time of the desired platform. No major inconsistency can be found between the minimum, maximum and average measurement values. It can be observed that for the passenger car and e-bus application, some parts of the SoX algorithm exceed the warning and critical level. A maximum execution time evaluation is done at

Table 1: Profiling of SoX execution performance in target embedded platform [us]

		RLS		EKF-SOC			EKF-SOH			SOP			SOE
		LTO	NMC	LTO	NMC	Pack	LTO	NMC	Pack	LTO	NMC	Pack	Pack
HIL Setup	Min	104	104	93	102.8	9.2	37.3	37.4	0.12	30.5	31	0.3	1.4
	Max	106	106	96.4	105.6	9.3	37.8	37.9	0.14	31.3	32	0.3	1.5
	Avg	105	105	93.7	103.9	9.2	37.4	37.5	0.13	30.8	31.3	0.3	1.4
Passenger Car	Min	413	207	357	201	25	147	76	0.12	113	60	0.3	1.5
	Max	420	210	367	207	25	149	78	0.16	114	62	0.4	1.7
	Avg	415	208	362	204	25	148	76	0.14	116	61	0.3	1.6
Electrical Bus	Min	825	819	737	265	65	303	294	0.13	240	254	0.3	2.1
	Max	836	834	750	275	67	306	295	0.18	246	260	0.4	2.2
	Avg	824	824	742	269	66	304	297	0.15	242	256	0.3	2.2

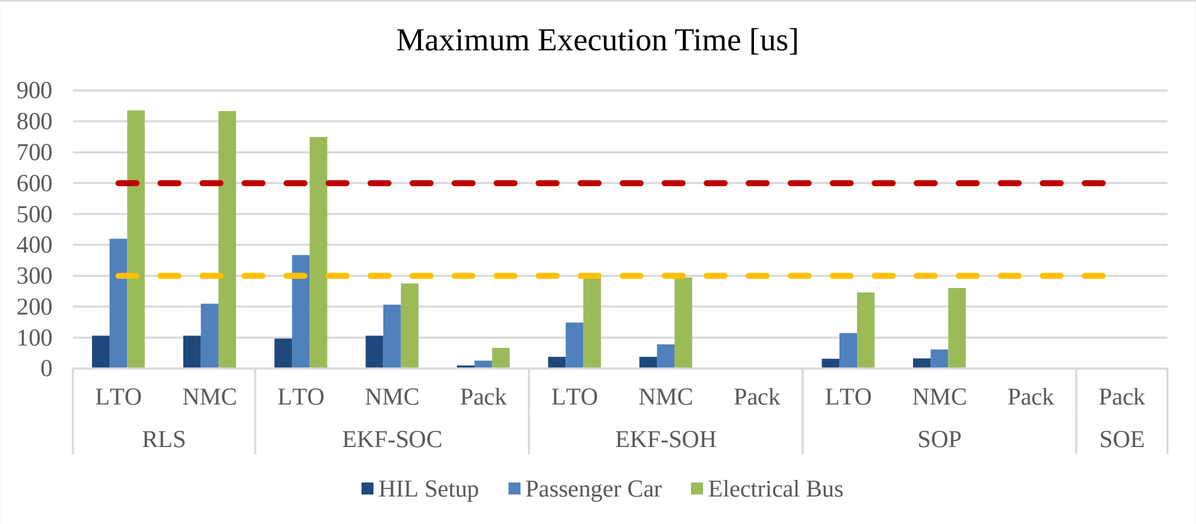


Figure 8 Maximum execution time of SoX algorithms within a core of the microcontroller [us]

Microcontroller overall execution performance in Table 2 presents an acceptable scenario, as the execution times of the overall software inside the BMC do not exceed 20.5% of overall capability in the worst-case scenario, as the e-bus would be. Overall execution time of the SoX algorithms executed in the Core 0 represent only 0.39% of the overall microcontroller load.

Table 2: Overall microcontroller core load KPI [%]

		Core 0	Core 1	Core2	SoX on Core 0
HIL Setup	Min	10.1	6.9	0.9	0.058
	Max	15.0	13.0	1.5	
	Avg	14.3	7.7	1	
Passenger Car	Min	11.7	6.9	0.9	0.173
	Max	16.6	12.5	1.4	
	Avg	14.9	7.8	1	
Electrical Bus	Min	14.8	6.9	0.9	0.390
	Max	20.5	8.2	1.5	
	Avg	15.9	7.3	1	

This ensures confidence in continuing the tests with real cells, after the definition of countermeasures for improving the execution performance of the SoX algorithms.

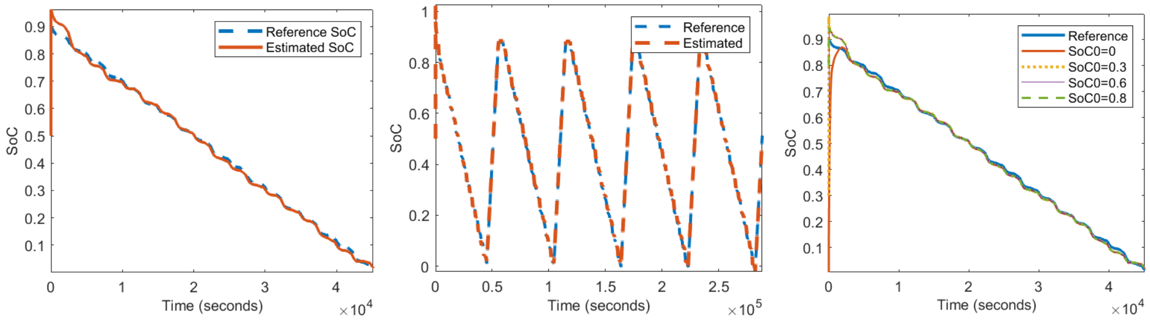
4.2 SoX Algorithm Validation

To validate the effectiveness and real-time feasibility of the SoX algorithms, a series of tests have been planned. The first round of testing focused on algorithmic validation, including the accuracy of state estimation, convergence behavior, and the stability of the estimated states. For this purpose, two cell types were evaluated under dynamic mission profiles in laboratory conditions at HELIOS partner facilities, including the Danish Technological Institute (DTI), Denmark. The tested cells included an NMC pouch cell rated at 3.65 V and 73 Ah, and an LTO prismatic cell rated at 2.3 V and 20 Ah. A comprehensive test matrix was developed to generate the required data for algorithm development and validation under various operating conditions, including different temperatures, SoC ranges, and load profiles. The battery states are co-estimated under a dynamic mission profile test called Aarhus Driving Cycle Profile or ADCP shown in Fig. 4(a). The results of co-estimations under this profile are summarized in Table 3.

Table 3: RMSE of SoX co-estimation under ADCP at 25°C

	SoC	SoH_R	SoH_C	SoPCh	SoP_{Dch}
NMC	~0.008	~0.041	~0.015	~ 10 W	~ 8W
LTO	~0.012	~0.052	~0.021	~15 W	~ 18W

Typical SoC estimation results related to the NMC cell at 25°C are shown in Fig. 6 demonstrating the effectiveness of the EKF-based approach under various operating scenarios. These include multiple ADCP cycles to evaluate the stability of the estimates, as well as tests with different initial SoC values to validate the convergence behavior of the algorithm. The results demonstrate that co-estimation algorithms are applicable to different cell chemistries, requiring only calibration of the software—such as updating lookup tables and model parameters—for each specific cell type.



(a) (b) (c)

Figure 9: (a) SoC estimation of NMC cell under ADCP (b) SoC estimation over several cycles of ADCP demonstrating SoC stability (c) SoC estimation with different initial values showing the convergence of SoC filter.

4.3 Real system validation

Initial hardware setup integration is performed with a system consisting of one battery module equipped with LTO chemistry cells, which is measured for validation of the measured signals acquired by the BMS and later used for virtual validation of the proposed algorithms before testing real cells in a bigger setup.

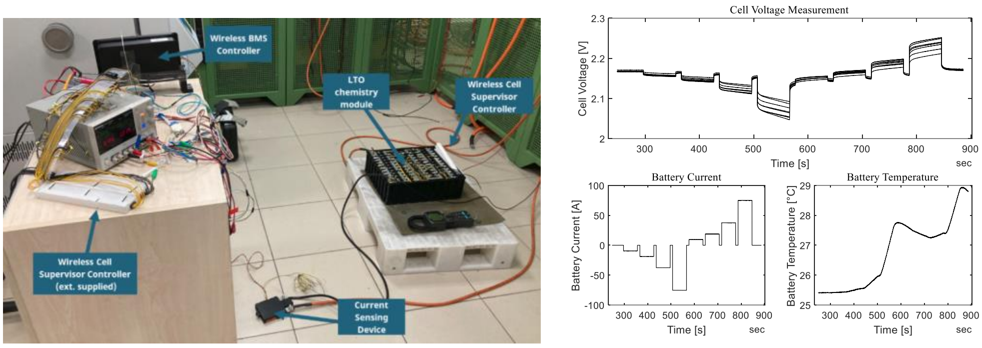


Figure 10: a) Setup of HIL for validation of first prototype based on LTO cells, b) Measurement of cell voltages, current and temperature

5 Conclusions and future work

This paper presents a modular and scalable framework of SoX estimators designed for heterogeneous battery packs integrating multiple cell chemistries. The developed SoX algorithms—including SoC, SoH, SoP, and SoE—work in synergy to co-estimate key battery states in real time. They are implemented on an automotive-grade microcontroller within a flexible BMS architecture capable of managing up to 512 individual cells. The algorithms are adapted for the unique characteristics of NMC and LTO cells and validated using experimental cell data on dynamic test profiles emulating actual driving situation. The results demonstrate that the proposed SoX estimators operate effectively under heterogeneous battery configurations and are well-optimized in terms of memory and computational requirements. Real-time checks confirm the embedded viability of the solution, with algorithm execution adhering to the resource constraints of ASIL C(D)-level safety-critical systems. Countermeasures are scheduled to be implemented for further robustness of execution of algorithms.

Future work will focus on the functional validation of the SoX algorithms using actual battery modules and full battery packs developed under the HELIOS project, after improving the performance of the algorithms due to introduced countermeasures. Empirical measurements and performance benchmarking are projected to be done on both passenger car and electric bus applications. These results will further validate the robustness, accuracy, and scalability of the proposed estimators in real-world operational environments.

Additional future work is to evaluate the migration of these SoX algorithms towards a more powerful platform, enabling the concept of software defined vehicle, adapting the implementation done in Autosar classic to adaptive. Additionally, as the execution of the algorithms for SoX prediction are array based, the potential of SIMD and vector processing will be investigated with aim of understanding the performance increase potential of the introduction of these parallel processor units available.

Acknowledgments

The research work is conducted as part of the project Horizon Europe HELIOS – "High-Performance Modular Battery Packs for Sustainable Urban Electromobility Services". The HELIOS project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 963646.

References

[1] F. Naseri, C. Barbu, and T. Sarikurt, "Optimal sizing of hybrid high-energy/high-power battery energy storage systems to improve battery cycle life and charging power in electric vehicle applications," *Journal of Energy Storage*, vol. 55, p. 105768, 2022.

[2] T. Jezdinsky, Schweizer-Berberich, M., Romero Baena, J. A., Grass, P., Braun, "T. B. The EU project HELIOS – Improvement of BMS functionality," doi: 10.5281/zenodo.6675889.

[3] F. Naseri, E. Schaltz, D.-I. Stroe, A. Gismero, and E. Farjah, "An enhanced equivalent circuit model with real-time parameter identification for battery state-of-charge estimation," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 3743-3751, 2021.

[4] G. L. Plett, "Recursive approximate weighted total least squares estimation of battery cell total capacity," *Journal of Power Sources*, vol. 196, no. 4, pp. 2319-2331, 2011.

[5] G. L. Plett, "High-performance battery-pack power estimation using a dynamic cell model," *IEEE Transactions on vehicular technology*, vol. 53, no. 5, pp. 1586-1593, 2004.

[6] A. Spice. Automotive Spice. https://vda-qmc.de/automotive-spice/ (accessed 2024-10-30, 2024).

[7] AUTOSAR. https://www.autosar.org/, (accessed 2024-10-30, 2024).

[8] MISRA. https://misra.org.uk/, (accessed 2025-04-27, 2025).

Presenter Biography

Juan Alberto Romero Baena obtained the B.Sc. degree in Industrial Electronics from the Polytechnic University of Catalonia (UPC) in 2007, M.Sc. degree in Electronics Engineering from UPC in 2010, and a Master Engineering in Electromobility by THI / WZS (2016). He is associated researcher by IREC, where he pursues his industrial PhD Program in Electronics Engineering by UPC, applied to BMS and related power electronics. His focus research area are battery electronics for heterogeneous batteries (balancing, reconfiguration) and energy conversion for charging (AC/DC, DC/DC, HPC and wireless).

He worked uninterruptedly in the automotive TIER-1 industry since 2010, first in Adasens; since 2012 by Continental Automotive, spin-off Vitesco Technologies (2020), and Schaeffler (2024).