Defining 'Better' in the World of BMS Algorithms: Scenarios and Requirements for Automotive Applications

Franziska Berger [1,5§], Dominik Joest[1,5], Elias Barbers[4,5], Katharina Quade [1,5], Ziheng Wu [1,5], Dirk Uwe Sauer[1,2,3,4,5], Philipp Dechent[1,5]

[1] Chair for Electrochemical Energy Conversion and Storage Systems, Institute for Power Electronics and Electrical Drives (ISEA), RWTH Aachen University, Campus- Boulevard 89, 52074 Aachen, Germany
[2]Center for Ageing, Reliability and Lifetime Prediction for Electrochemical and Power Electronic Systems (CARL), RWTH Aachen University, Campus-Boulevard 89, 52074 Aachen, Germany
[3] Institute for Power Generation and Storage Systems (PGS), E.ON Energy Research Center (E.ON ERC), RWTH Aachen University, Mathieustrasse 10, 52074 Aachen, Germany
[4] Helmholtz Institute Münster: Ionics in Energy Storage (HI MS), IEK 12, Forschungszentrum Jülich, 52425 Jülich, Germany
[5] Jülich Aachen Research Alliance, JARA Energy, Templergraben 55, 52056 Aachen, Germany

§ Corresponding author: batteries@isea.rwth-aachen.de, franziska.berger@isea.rwth-aachen.de

Abstract (150 words): Lithium-Ion battery

This paper investigates the complexities in the development and implementation of Battery Management System (BMS) algorithms. Understanding the performance of BMS algorithms is crucial, as poor algorithms can lead to customer frustration, safety issues, and acceleration of battery degradation, becoming a significant liability for automotive companies. However, a clear definition of "better" for BMS algorithms remains elusive, complicating validation efforts. There are also challenges in hardware selection, data storage, calibration during development and use, and cost constraints. Furthermore, different state estimation principles have varying data requirements, impacting algorithm performance. To address these issues, we derived a comprehensive methodology of automated test scenario creation obtained from expert interviews, considering aspects like computational capacity and specific usage scenarios for algorithm comparison and validation. Our research underscores the importance of robust validation and the nuanced dependencies of algorithm performance on factors such as hardware, stored data, calibration, and specific usage contexts.

Keywords: lithium-ion batteries, requirements, battery management systems, state estimation

1. Introduction

As lithium-ion technology paves the way for sustainable energy alternatives, its adoption in various sectors - such as automotive, railway, maritime, aviation, and energy storage - is becoming increasingly commonplace. A crucial component that ensures the efficient operation of lithium-ion batteries (LIBs) across these sectors is the battery management system (BMS).

The BMS carefully oversees each battery cell, ensuring safety, reliability, and optimal performance. It consists of hardware and software and estimates the battery's state and implements measures such as cell balancing and thermal management to optimize the operational range and longevity.

However, as direct battery state measurement is unfeasible, the BMS software utilizes various estimation algorithms, such as those for state of charge (SoC), state of health (SoH), and state of power (SoP).

Despite ongoing advancements, the practical improvement of these BMS algorithms in field remains a pressing challenge [1] due to the absence of a clear definition of what constitutes a "better" algorithm. As the numerous potential enhancements (e.g., robustness, speed, accuracy) [2] must be balanced against potential risks like safety issues and accelerated battery degradation[3], the proposed methods necessitate rigorous validation before deployment in real-world BMS applications. Furthermore, the lack of clear guidance regarding test scenarios [4] and algorithm requirements in existing literature adds to the complexity.

Addressing these gaps, this paper discusses the requirements and validation aspects of BMS algorithms, drawing from insights gathered from global battery and BMS specialists. Given the broad range of potential BMS applications, our focus is narrowed to automotive applications, specifically all electric passenger cars. Our goal is to offer a comprehensive overview of general real-world BMS algorithm requirements and detail those for SoC, SoH, and SoP estimators. Moreover, we provide guidance for algorithm selection and demonstrate the intricacy of validation through an example involving Kalman filters for SoC estimation, while also showcasing a methodology for creating automated test scenarios.

<u>Challenges in the validation of BMS algorithms</u>

The process of comparing existing literature on estimation methods and algorithm performances reveals significant diversity in validation procedures [4]. Approaches for accuracy evaluation vary, utilizing different error types like Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), and battery load representation through diverse drive cycles. Beyond standard cycles like the Worldwide harmonized Light vehicles Test Procedure (WLTP) and Urban Dynamometer Driving Schedule (UDDS), new representative drive cycles [5] and real operation data can facilitate realistic application profiles for validation.

Algorithm sensitivity to environmental factors like temperature and operational conditions requires thorough validation [4,6]. Given the profound temperature impact on battery performance [7], the validation process cannot rely on a single standard cycle at one temperature [8]. Extreme conditions, which significantly influence performance, must be integrated into BMS algorithm validation.

In addition, the internal conditions and details of the algorithm implementation can also impact the comparability of estimators. For instance, while two implemented Kalman filters for SoC estimation may share a basic principle, their performance may differ based on the battery model and parameter identification method used. Besides various types of equivalent circuit models (ECMs), more complex physical-chemical models may replicate cell behaviour [6]. However, even ECMs can vary in accuracy depending on their complexity [7].

State estimation poses a challenge as the definitions of the states estimated by the BMS algorithms are inconsistent across different studies and due to the absence of a known actual value. For instance, as two main ageing indicators for battery cells are resistance increase and capacity loss [9,10], the definition of SoH varies greatly in literature, being related to one of impedance [11], internal resistance [1213], capacity [14] or combinations [15], or even specific to electric vehicle applications in different sources. Additionally, the reference values of resistance and capacity can vary due to the definition and way of determination of these quantities [6,8].

For SoC, the definition in [16] uses the relation to the nominal capacity measured at the C/30 rate at room temperature, in [17] the reference for the SoC value is the 'actual' capacity without specifying a temperature and discharge condition. Thereby the maximal available capacity depends on ambient temperatures and loaded current rate during operation, as well as on the ageing level [18]. The authors in [9] discuss of the different battery capacity definitions, which is relevant for the different state estimations.
These complexities underscore the importance of establishing definitions and requirements at the outset of validation and maintaining consistency. A clear outline of test scenarios, including validation profile, test temperature, operational SoC range, Depth of Discharge, and state of cell degradation, is also imperative to enable comparability of the algorithms' performance in literature.
Furthermore, estimators must meet certain application-specific requirements. These are very strongly linked to each other so that the topic is more complex than evaluating if an algorithm is robust or accurate, especially without accounting for various test scenarios. While there are broad trends in estimation methods, a closer look at application-specific requirements often reveals a need for compromise. Therefore, the selection of a suitable algorithm is challenging. As existing literature provides limited details about BMS algorithm requirements, this study includes insights from industry scientists and engineers through interviews, the findings of which will be discussed in the subsequent section.

2. Results and Discussion

Requirements on BMS algorithms

The requirements on BMS algorithms, mainly drawn from the interviews, were categorized into performance, hardware, and implementation effort. These three criteria are discussed with respect to various aspects such as memory requirements, complexity, and convergence time with a focus on SoC, SoH and SoP estimation.
Numerous algorithms such as SoC, SoH, and SoP estimators have been extensively presented in the literature, with a diverse array of methodologies proposed for each individual state. These methodologies are traditionally validated using universal performance metrics, such as accuracy, robustness, and real-time capability, to underline their effectiveness, but often neglecting the mutual influence.
However, carefully defining these terms of requirements used in the validation of the algorithms is crucial. Scope for interpretation can lead to serious misunderstandings, making it essential to be precise in terminology for meaningful comparisons across different methodologies [19][Chaoui2017].
Further critical parameters besides accuracy, robustness, and real-time capability are categorized in Figure 1. An overview of the various ways these performance criteria are defined can be found in Supporting Information.

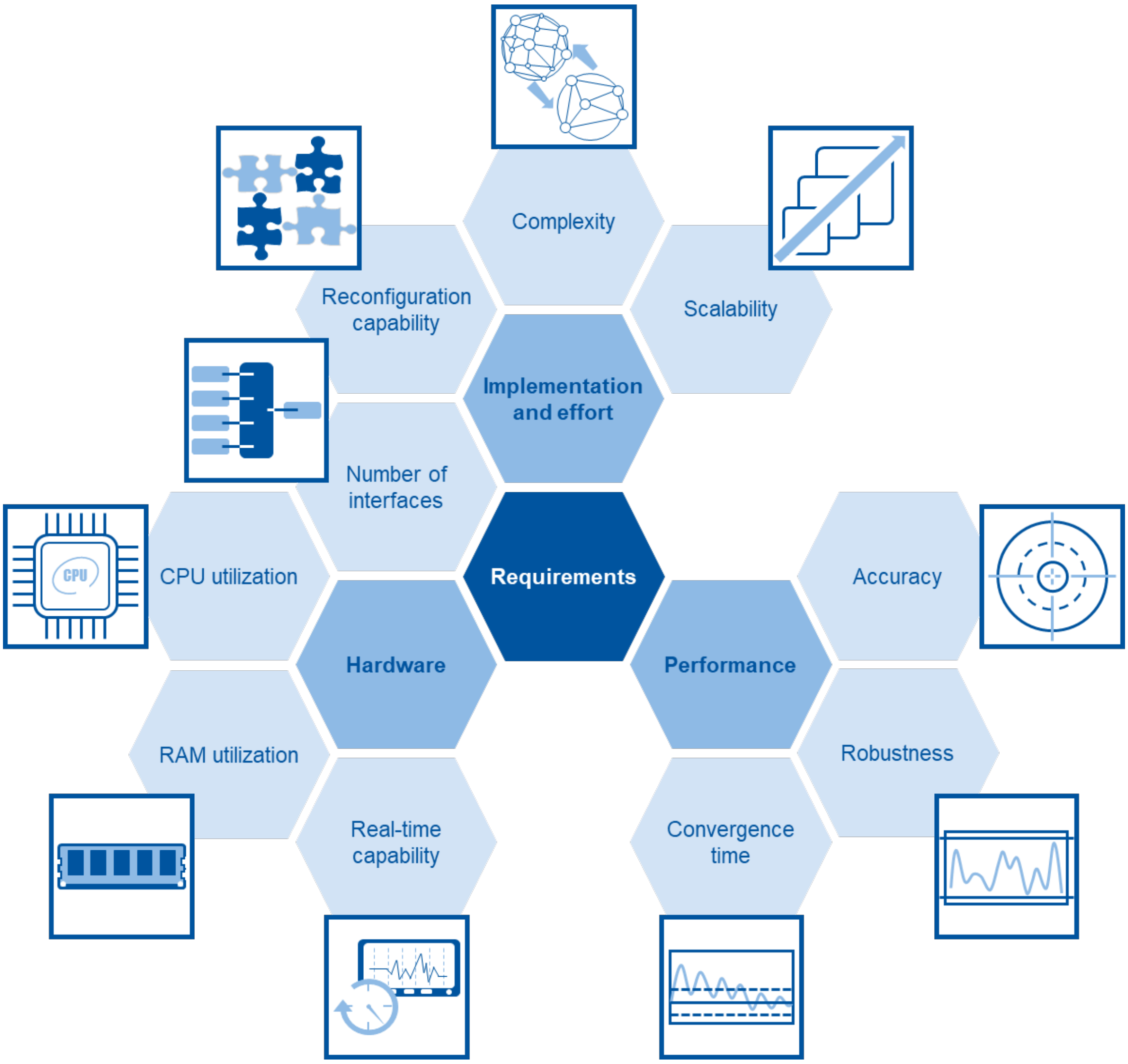

Figure 1: Overview of general real-world BMS algorithm requirements. Beyond accuracy, robustness, and real-time capability, other critical measures come into play for BMS algorithms. These include complexity, confidence, memory requirements, scalability, and iteration steps. A classification of the above criteria in performance, hardware, and implementation and effort is suggested.

Performance

Accuracy, a measure of how closely a measurement aligns with a specific quantity, is crucial in evaluating BMS algorithms. The 'true' value used for comparison can be defined under laboratory conditions to maintain consistency, with the deviation measured as the difference between the estimated and test bench value. Various test scenarios and categorizations of driving cycles need to be considered for a more comprehensive evaluation. Accuracy is also evaluated based on the average error values over a time window or driving cycle, the absolute value at each time point, and a set threshold for the maximum absolute error or MAE, providing a measure of robustness accuracy.
Manufactures often set individual thresholds based on their applications and available hardware. The cell chemistry affects the accuracy requirements for both control and hardware, as does the higher-level system in which the subsystem is embedded. This system must meet vehicle-level requirements such as range, performance, lifetime, availability, safety, and cost, while maintaining a minimum level of accuracy.
A few years ago, an accuracy of 3-5 percentage points was accepted for SoC estimators, but today, 1-3 percentage points is often required, with 1 percentage points over the entire duration being exceptional. The error can be averaged over several cycles, but the maximum error, or error amplitude, is frequently considered and compared to a tolerance band. The given accuracy values in this paper,

drawn from the interviews, are maximum errors. Looking at averaged values is not beneficial because of time dependencies.

Maintaining accuracy near high and low SoC levels is usually more critical than in the mid-range to avoid overcharging and sudden battery depletion. High accuracy at the end of charging or discharging can therefore extend the usable SoC range. The accuracy should hit the tolerance band relatively quickly instead of adapting and converging over many cycles. Further challenging influence factors on SoC accuracy are cell variations [2], measurement sensor and model accuracy [4,6].

Talking about accuracy of SoH in this paper is related to resistance and capacity estimation accuracy [10,20]. To validate online SoH estimates, cells must be subjected to different ageing conditions, such as varying temperatures, C-rates, and Depth of Discharge [21], while simultaneously tracking resistance and capacity reference values. The SoH estimation algorithms need to be validated under each of these ageing conditions [8]. Perspectives of the interviewees on the desired accuracy of SoH estimations vary, accuracy in the range from 1 percentage points in laboratories to 5 percentage points in application is possible. The objective is to consistently estimate an online value, even if the accuracy is slightly compromised.

Estimating the third state discussed in this paper, the SoP, requires high accuracy due to its correlation with safety. The understanding of SoP becomes crucial during rapid accelerations, such as overtaking manoeuvres, where the battery is expected to deliver maximum current within a specific time frame without breaching constraints like voltage, SoC, or temperature. A slightly lower estimation accuracy may be acceptable if this state is used merely to gauge the likelihood of meeting performance requirements in subsequent cycles. The overarching goal is to minimize the discrepancy between the estimated and actual state values, yet determining the reference value to calculate this error remains an open question.

Robustness in a system signifies its ability to maintain stability amidst unexpected conditions. Applied to BMS algorithms, robustness is less about stability and more about consistent performance and reliability, even with slight inaccuracies. A robust algorithm should also identify errors and deliver consistent estimates under various conditions. In a broader context, robustness relates to whether the algorithm meets customer expectations and functions predictably. The validation process for robustness involves maintaining an acceptable error range across different scenarios, application profiles, start conditions, parameter uncertainties, sensor failures, measurement inaccuracies and environmental factors.

BMS algorithms are expected to consistently provide accurate results, even under atypical scenarios such as usage of taxi with very frequent rides and in marginal SoC areas. However, they can be expected to perform less accurately under extreme environmental circumstances or during abusive events like short circuits in a cell.

Depending on the application, there are specific requirements and for example, the ones on robustness differ. In automotive applications, the necessary robustness is similar across various applications like Hybrid Electric Vehicles (HEV) and Battery Electric Vehicles (BEV) using the same cell chemistry.

Temperature, as it serves as an input to the algorithms, is a critical variable that impacts the robustness of most algorithms. Given the wide-ranging temperature differences around the globe, and seasonal or day-night fluctuations, BMS algorithms must perform reliably across diverse thermal conditions of the battery.

Modern BMS in BEVs do not typically measure each cell's temperature individually [22]. Instead, temperature is tracked across the entire battery pack, making knowledge of its distribution crucial. If unknown, it poses difficulties for algorithms and potential safety risks. Usually, researchers assume a rough knowledge of the pack's minimum and maximum temperatures. For robustness, we need to constantly validate these assumptions, such as when calculating average temperature, and inform the higher-level system if precise computation is not currently possible.

Temperature generally doesn't directly affect the control unit's computing power but instead has a more pronounced impact on sensors. For example, if a shunt is used for measurement and its resistance significantly changes due to temperature extremes, these influences need to be considered.

Modern control units usually account for this, but sensor faults such as noise, offset, bias, and failure must also be examined for robustness validation.
Algorithms are frequently sensitive to their inputs, and state estimators often use outputs from other states as inputs. Therefore, robustness and validation of these estimations are crucial across a variety of settings. For instance, a Kalman filter used for SoC estimation requires cell parameters like resistance and capacity, which depend on the chosen model, as inputs. This necessitates the robust online estimation of these parameters. Besides temperature and sensor faults, factors such as varying drive cycles, different SoC usage ranges, and extreme cases in a designated design area should also be investigated and tested.

Hardware

Real-time capability is a critical feature of a system or algorithm, representing its ability to execute tasks on the BMS within a predefined cycle or at a designated frequency and to make decisions swiftly based on prevailing conditions. This requirement for real-time execution means that an algorithm must be capable of determining desired values in tandem with operations, adapting as the state variables change.
In signal processing, the real-time functionality is closely tied to the Nyquist criterion, a fundamental rule in digital signal processing. The criterion dictates that a system or algorithm should operate at a speed at least twice that of the highest frequency component of the underlying process. This allows for accurate sampling and avoids the aliasing effect, which can distort signal representation and analysis. Accordingly, the dynamic of the battery parameters, more detailed the time constants of the dynamic, specifies the sampling rate and differs for the three measured variables voltage, current and temperature.
Typically, the industrial standard for current measurement in vehicles is 10 ms (100 Hz), but a rate of 1 ms (1 kHz) is recommended in the absence of integration of the current within the measuring circuit for the charge throughput. This rate can differ for applications such as grid-scale energy storage or fast charging. For voltage sensors 100 ms is acceptable, the algorithms should be able to handle this. Synchronized measurement of current and voltage is common, albeit challenging due to separate integrated circuits. For temperature measurement, a slower sample rate of 1 s is sufficient as it does not change as fast as the electrochemical dynamics and its time constant is bigger.
The sampling and algorithm update periods are usually the same, but they could differ. Importantly, all computations need to be completed within the given sample rate. For instance, SoC estimates must be updated in real-time and are needed continuously for automotive applications. The update rate depends on the performance of the specific vehicle application: For high-energy use cases update rates of 10-60 s are proposed, whereas for high power update intervals should be more frequent than every 10 s.
In general, estimating the SoH needs less real-time capability than the SoC, given SoH changes slowly over months while SoC changes within seconds to minutes. Even if calculations aren't completed in one cycle, distributing them over several can still maintain real-time capability for SoH.
The update interval for SoH largely depends on the application and the speed at which ageing parameters like capacity or resistance change. Generally, the goal is to prioritize accuracy over speed by averaging estimation values over a suitable time window. In vehicles the update interval ranges between several minutes to one hour. Some interviewees suggest updating SoH after each driving cycle or plug-in cycle, i.e., charge event. Advanced techniques for SoH estimation like Electrochemical Impedance Spectroscopy need higher sample rates to obtain more information about the impedance spectrum. Alternatively, transmitting measurements to the cloud provides more computational resources but can't replace legally mandatory real-time on-board calculations.
Compared to SoC and SoH, the SoP estimation is time-sensitive due to safety risks and needs rapid updates. Update rates of 10 to 50 ms are required, and valid SoP values must be available at every execution step. The time window and requirements for predicting SoP depend on the specific application, potentially employing multiple windows for different planning strategies.

Views on the impact of hardware on BMS algorithm requirements differ. Typically, hardware isn't the limiting factor since high-quality technology is readily available. However, industries often resort to less expensive hardware to cut costs. The trade-off between performance and cost, particularly with sensors, is a crucial factor during hardware selection.
Requirements from the application and customers theoretically dictate the hardware. However, this selection can affect requirements and significantly influence BMS algorithm performance. Some methods, like the ampere-hour counter, are directly influenced by current and voltage measurements and hence the input accuracy. Conversely, certain methods can compensate for specific deviations, like the model-based Kalman filter [4,6].
If error compensation isn't possible, the measurement equipment limits the minimum achievable errors of the algorithms. While initial calibration can be done for current and voltage sensors, their accuracy diminishes over time due to ageing. Additionally, hardware requirements can vary with cell chemistry. For instance, lithium iron phosphate (LFP) cells demand higher voltage sensor accuracy due to their flat SoC-OCV curve. A 15 mV voltage error can lead to more than 20% difference in the SOC estimation.
Limitations may also arise regarding storage frequency or transport frequency through CAN bus. The increasing number of battery cells necessitates more computational steps, potentially leading to time delays. Furthermore, memory storage on the BMS is limited due to cost constraints. While data is typically processed directly and seldom stored for extended periods, dynamic data with a high sample rate present a challenge. Non-volatile memory is also needed to store initialization parameters.

Implementation and effort

While costs often dictate storage decisions, the type of state estimation principles in play determines the historical data needed. Recursive algorithms, which base computations on the last computed value, don't require historical data storage, unlike batch processes that do. Machine learning algorithms especially need large data sets to capture unusual scenarios.
As storing all data, measured and estimated, isn't practical, a selective portion of data is captured using techniques like sliding windows. For functionalities as lifetime predictions, historic data is essential, most likely requiring a cloud system for storage and sub-sampling due to onboard limitations.
Histograms—mapping usage behaviour, current-voltage curves, temperature curves, full cycles, and possibly other vehicle parameters—are essential to monitor the battery pack usage for second-life use and calendar ageing, and to identify abnormal use cases, but the limited storage and processing resources pose the challenge of data condensation.
As previously noted, the availability of state estimators' outputs at system restart must be ensured. For SoC estimation, historical data is less relevant as estimations are based on present values. However, for SoH estimation, reference parameters at the start of life, such as initial capacity and resistance, and histograms for calendar ageing (if no value is tracked) are necessary. Logged current, given its influence on cell behaviour, is crucial for SoP estimation as well as pulse duration limits for maximum currents.

Efficiency in algorithm implementation necessitates adaptability to minimize time and resources. Different applications present diverse performance profiles, particularly regarding dynamics and C-rate. Ideally, a single algorithm should cater to all these varying application areas with minor adjustments. Tailoring separate algorithms for individual applications is overly complex. For instance, within automotive applications, the aim here is to employ the same algorithms across different vehicle types as BEVs, HEVs, and PHEVs, making any necessary adjustments to the calibration, such as initializations and parameters, ideally in an automated fashion. Thus, limiting the complexity of BMS algorithms becomes beneficial.
Regardless of the type of state estimator, calibration data can be divided into two types: those collected during development and those obtained from users or workshops. In the automotive field, initial parameterization and calibration of algorithms are time-consuming and costly due to complex dependencies on factors such as SoC, temperature, current, and internal resistance over ageing. Minor

cell changes can lead to additional testing and calibration, emphasizing the goal to reduce parameter numbers.
In terms of in-field recalibration, maintaining small estimation errors for different states, especially for LFP/C cells, poses a major challenge. One possible solution is to recalibrate state estimation algorithms like SoC after a full charge. However, this presupposes a certain behaviour from the user, which is not always practical or desirable. Although it could work if the user naturally fully charges their vehicle, or if a suggestion is given. For instance, in plug-in Hybrid Electric Vehicles (PHEV), it isn't a reliable method as it depends heavily on user behaviour and requires knowledge of the maximum allowable algorithm deviation to conduct a recalibration.
Workshop calibration is less desirable due to time constraints during maintenance and the need for optimal vehicle-system level tuning. Ideally, a system should perform reliably throughout its lifespan without user restrictions. However, in industrial applications, regular maintenance intervals, such as every 2-3 months in energy storage systems, make systematic calibration methods more feasible.

Impacts through application scenarios

The necessity of applying a load profile to a battery for validating BMS algorithms cannot be overstated. Load profiles and their validation profiles are distinct to each electromobility application, including vehicles, trains, and ships, given the varying power requirements and constraints. For instance, standardized vehicle cycles like UDDS, WLTP, and US06 exist, yet they vary based on factors such as geography, regional legal stipulations, and climate conditions. Even within a single country, driving cycle characteristics can differ significantly. In addition to the application in general, the usage scenario and the individual driver also provide for variances. Exemplary, the load profiles of daily passenger cars and taxis can contrast drastically. Taxis, being in continuous operation, demand maximum battery capacity and minimal rest phases, thus reducing recalibration opportunities. On the other hand, sports cars, with fewer driven miles, long phases without use, and dynamic load, pose a different challenge for resistance estimation.
These different use cases impact the prerequisites and requirements of BMS algorithms. Ideally, all usage scenarios are known when designing an algorithm, as well as against which and how many load profiles the developed algorithms should be tested and validated. If the algorithm is used in industry, companies often have their own internal cycles to cover the most common scenarios.
In theory, a BMS algorithm should possess a degree of generality [1] that can be specifically enhanced for a particular application. This optimization can take various forms, such as robust optimization, ensuring every value in a set satisfies a constraint, or stochastic optimization, where a probability mass within a set validates a constraint, which is a slightly weaker criterion. Lastly, distributional robust optimization involves fitting a model to data using Gaussian distributions and evaluating the distance between the empirical distribution and the set of "true" distributions.
For testing and validation, application profiles should represent real-world situations as uncertainties, variations, and noise. However, encapsulating all possible scenarios in one cycle is challenging. Therefore, it's important to test standardized cycles for comparability under expected ambient conditions and in marginal cases.
Algorithm testing for accuracy and robustness can be time-consuming due to the high randomness in real-world applications. A pragmatic approach could involve prioritizing application scenarios based on occurrence frequencies or probabilities. Stochastic scenario generation, using tools like Markov chains models, can infuse randomness and variety into driving profiles.
The expanding array of added-value applications in the automotive sector, like vehicle-to-home and vehicle-to-grid, necessitates further nuanced consideration. Finally, the application decisively defines the required performance of BMS state estimators as accuracy and robustness.

Weighting of the criteria for the selection of the algorithms

Customer and application demands shape BMS algorithm requirements, limited by factors such as cost and hardware specification. Conversely, algorithm development might dictate hardware requirements, based on needed measurement accuracy and computational power. Both sides shape

algorithm selection. Interviewees revealed varying criteria for algorithm choice, with robustness, accuracy, and real-time execution emerging as decisive factors.
There is a balance between robustness and accuracy: a non-robust yet accurate algorithm is as ineffective as a robust, inaccurate one. Both traits must be considered for state estimators, with a worst-case scenario focus. Linked to robustness is plausibility—if the algorithm produces implausible estimates, it's unsuitable. Consistency without errors (e.g., negative SOC) and predictability are key.
Algorithms must also be implementable in real-time on target systems, with low resource use. In resource-restricted scenarios, simpler, efficient algorithms might be required. Efficiency becomes paramount when estimating states for many cells, as in large-scale applications e.g., battery-electric trains.
Moreover, an algorithm's generalizability is important, aiming for easy calibration and applicability across different applications and battery chemistries. Extensive testing is needed to ensure this, and it raises another criterion: testability. Estimation reproducibility and optimality verification are vital for example, the Kalman filter under some scenarios can be proven optimal, since it provides confidence bounds.
While academic researchers often add complexity for better accuracy, industry practitioners prioritize robustness, speed, and accuracy for quick, reliable estimates.

Simulation environment for validation of BMS algorithms

After outlining the requirements and selecting a suitable algorithm, the test scenarios for validating the algorithm must be defined and created accordingly. To handle the complexity of test scenario selection for validating BMS algorithm for electric vehicles, this paper presents a methodology for creating test scenarios and a validation example regarding accuracy using heat maps.
For validation, a Model-in-the-Loop (MiL) environment is appropriate to save costs and time by simulating the algorithms on a virtual BMS. However, care must be taken here to generate signals that are as close to reality as possible and finally to perform verification at the hardware level. The estimation outputs of the BMS algorithms as well as its performance are compared with the requirements. The workflow of this validation process is presented in Figure 2.

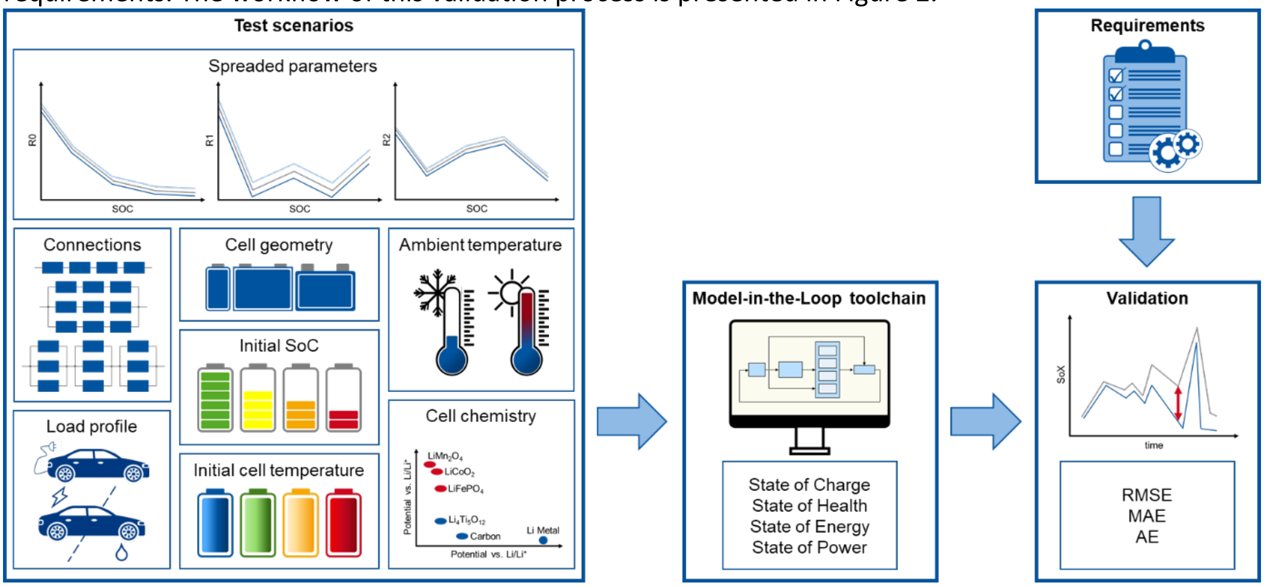

Figure 2: Workflow of BMS algorithm validation process from creation of test scenarios to validation in a Model-in-the-Loop environment considering the specified requirements. The variation of load profiles, initial cell, and ambient temperatures, initial SoC, cell chemistries, connections of cells and cell parameters, and their combinations form a large number of test scenarios. The Model-in-the-Loop environment enables the simulation for the different test cases and state estimators. To validate the algorithms, evaluation parameters such as root mean square error, mean absolute error and absolute error are calculated.

Example of validation process used for SoC estimators

In this paper, three different SoC estimators, based on the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) with online parameter updates, are validated for a selection of test scenarios, summarized in Table 1.
The focus of this paper is not on algorithm optimization, but to give an example of validation of a SoC estimator. For this reason, the algorithms have the same initialization values, e.g., for the covariances, to guarantee comparability and to highlight the influence of the battery cell, application, drive cycle, temperature, and sensor inaccuracy on the performance of the algorithms. For simplification, the accuracy of the battery models, R1RC and R2RC, does not impact the validation in this paper.
In the heat maps in Figure 3 and Figure S3-S5, the results are either for the same cell or the same profile to enable the investigation of the adaptability of the algorithms. The initial SoC variation is the same for all three cells, only the upper initial SoC depends on the application. For the LTO cell it is 80% initial SoC and for the NMC and NCA cell 100% due to the differences in HEV and BEV operational characteristics.
For every map, the same scaling of the colour means the same error, for RMSE and max. absolute error (AE) respectively. An upper error limit restricts the two different error types due to the requirements on the estimators: Outliers are not acceptable in real application. The threshold error value for the RMSE is 3 percentage points SoC difference and for the max. AE 5 percentage points SoC difference. These values are set by the authors. The errors above this upper threshold are illustrated by the same colour. This allows a quick overview for which scenarios the single algorithms are within the tolerance band.
In Figure 3, the comparison for the sporty profile and the different cells, NCA, NMC and LTO, respectively, shows better results for the EKF-R2RC and UKF-R2RC for the LTO and the NCA cells than for the EKF-R1RC. For this estimator based on R1RC battery models, not a single max. AE value is below 5 percentage points for the LTO cell and only one value for the NCA. Whereas for the NMC cell, the EKF-R1RC performs better compared to the two estimators based on a R2RC battery model.
Considering different temperatures, the two EKF perform better for temperatures above 0°C. This trend is not that pronounced for the UKF. For the UKF, the accuracy is higher with the LTO cell when the initial SoC is set to 80%, regardless of temperature. If sensor noise is added to the current and voltage signal, the EKF-R1RC can handle this for the NMC cell. Whereas the other two filters can only handle noise on the current sensor, in terms of accuracy, but not noise on the voltage sensor.
Not only for the different settings as temperature and initial SOC the performance of the estimators varies, also for different drive cycles. Therefore, the investigation for the three cell chemistries and different driving cycles are presented in Figure S3 to Figure S5 in Supporting Information.

The results show that the validation and selection of algorithms is complex, and the performance of different Kalman filter implementations can vary greatly, compared to each other, and for different influence factors as temperatures and measurement inaccuracies. Therefore, using various test scenarios is indispensable for validation and to ensure comparability. In order to minimize testing efforts, it is necessary to make a representative selection of scenarios.
For this reason, not all test cases and influencing factors are considered in the presented heatmaps, but only a selection. The algorithms are validated regarding accuracy but not related to robustness and real-time capability or the further requirements from the requirements section. Furthermore, the initialization has a great influence on the performance of the Kalman Filter. Cell dependent initial values are neglected in the example.
The test case at 25°C with 100% initial SoC and without sensor noise is a good example that not only one error validation parameter as RMSE is sufficient. For this scenario the RMSE of EKF-R2RC is for the NMC cell 1.13 percentage points, but the max. AE is above 5 percentage points. According to the predefined thresholds, the estimator's performance is not accurate enough, even the RMSE is low, and this error is more sensitive to outliners than the max. MAE for example.

For sporty profile and different cells (NCA||Graphite, NMC||Graphite and NMC||LTO)

EKF-R1RC		Error	RMSE			max. AE		
		Cell	NCA || C	NMC || C	LTO || C	NCA || C	NMC || C	LTO || C
Temp.	Init. SoC	Sensor						
0°C	20%	No noise	2.71	>3	>3	4.55	>5	>5
	50%	No noise	>3	>3	>3	>5	4.95	>5
		U & I noise	>3	>3	>3	>5	>5	>5
	80%	No noise	-	-	2.41	-	-	>5
	100%	No noise	>3	1.31	-	>5	2.36	-
25°C	20%	No noise	2.93	0.19	>3	>5	0.83	>5
	50%	No noise	>3	0.03	>3	>5	0.24	>5
		I noise	>3	0.07	>3	>5	1.4	>5
		U noise	>3	0.64	>3	>5	1.22	>5
		U & I noise	>3	0.61	>3	>5	1.35	>5
	80%	No noise	-	-	>3	-	-	>5
	100%	No noise	>3	0.13	-	>5	0.55	-
40°C	20%	No noise	>3	1.48	>3	>5	2.94	>5
	50%	No noise	>3	0.47	>3	>5	0.69	>5
		U & I noise	>3	0.28	>3	>5	0.87	>5
	80%	No noise	-	-	>3	-	-	>5
	100%	No noise	>3	0.6	-	>5	0.92	-

EKF-R2RC		Error	RMSE			max. AE		
		Cell	NCA || C	NMC || C	LTO || C	NCA || C	NMC || C	LTO || C
Temp.	Init. SoC	Sensor						
0°C	20%	No noise	1.71	>3	1.68	>5	>5	>5
	50%	No noise	2.11	>3	2.26	>5	>5	3.73
		U & I noise	>3	>3	>3	>5	>5	>5
	80%	No noise	-	-	1.87	-	-	>5
	100%	No noise	>3	>3	-	>5	>5	-
25°C	20%	No noise	1.76	0.41	2.42	>5	3.27	>5
	50%	No noise	1.88	0.7	1.81	4.07	3.87	4.98
		I noise	1.89	0.51	1.57	>5	>5	4.74
		U noise	>3	>3	>3	>5	>5	>5
		U & I noise	>3	>3	>3	>5	>5	>5
	80%	No noise	-	-	1.94	-	-	4.23
	100%	No noise	1.74	1.13	-	>5	>5	-
40°C	20%	No noise	1.91	1.22	>3	>5	3.23	>5
	50%	No noise	2.15	0.47	>3	3.91	3.54	4.67
		U & I noise	>3	>3	>3	>5	>5	>5
	80%	No noise	-	-	2.18	-	-	>5
	100%	No noise	1.24	1.07	-	>5	>5	-

UKF-R2RC		Error	RMSE			max. AE		
		Cell	NCA || C	NMC || C	LTO || C	NCA || C	NMC || C	LTO || C
Temp.	Init. SoC	Sensor						
0°C	20%	No noise	0.98	>3	1.29	1.45	>5	>5
	50%	No noise	1.26	>3	>3	1.66	4.12	>5
		U & I noise	>3	>3	>3	>5	>5	>5
	80%	No noise	-	-	0.53	-	-	3.42
	100%	No noise	1	1.28	-	3.22	2.03	-
25°C	20%	No noise	1.17	0.21	2.08	1.7	0.47	>5
	50%	No noise	0.88	0.5	1.57	1.35	1.66	3.59
		I noise	1.55	0.63	1.2	4.75	4.93	3.13
		U noise	>3	>3	>3	>5	>5	>5
		U & I noise	>3	>3	>3	>5	>5	>5
	80%	No noise	-	-	0.37	-	-	0.78
	100%	No noise	0.72	0.49	-	2.44	2.69	-
40°C	20%	No noise	1.27	1.14	2.36	1.69	1.48	3.58
	50%	No noise	0.91	0.3	>3	1.53	0.75	>5
		U & I noise	>3	>3	>3	>5	>5	>5
	80%	No noise	-	-	0.7	-	-	1.16
	100%	No noise	0.87	0.76	-	2.59	2.52	-

Figure 3: Heatmap of simulation results of NCA, NMC and LTO for sporty profile. The heatmap contains the results for the EKF-R1RC, EKF-R2RC and UKF-R2RC for the three different cell chemistries, different temperatures, initial SoC values and sensor noise, based on the same driving profile to compare the performance of the Kalman Filters for different cell chemistries and the same driving profile.

3. Conclusion and outlook

Meaningful validation of BMS algorithms and the lack of discussing the test scenarios in existing literature is challenging. This paper fills the gap and gives an overview regarding requirements and aspects of algorithm validation in automotive application using insights from interviews with national and international researchers specializing in LIBs and BMS-related fields. This paper focuses on SoC, SoH and SoP estimation. The various requirements, as accuracy and robustness, are closely linked and can differ for different states. Additionally, this paper has attempted to provide guidance in the selection of algorithms. Both sides, the given requirements by costumer and application as robustness, real-time capability and accuracy, and the ones given by the algorithms on hardware should be considered.
The given validation example for three different SoC estimators, considering various temperatures, driving cycles, initial SoC and measurement inaccuracies, underlines the necessity of a comprehensive validation of BMS algorithms, thereby the investigated test cases represent only a fraction of all potential test scenarios. Furthermore, the results indicate that relying on a single validation metric for a meaningful evaluation is not sufficient.
Validation of BMS algorithms based on realistic scenarios is relevant to test the algorithm for real driving behaviour, therefor a pool of driving cycles is essential. There are different possibilities to generate new driving cycles based on recorded driving data.
The influence of the battery parameter estimation is not negligible as the parameters are inputs of the filter. Additionally, optimizing the performance of the Kalman filter by initializing the algorithm for each cell chemistry is the next step. A systematic approach would be desirable here instead of the trial-and-error principle. To validate the performance more comprehensively, further validation criteria besides accuracy from Figure 1 and additional evaluation criteria beyond RMSE and maximum error should be incorporated.

4. Methods

Expert interviews

To provide an overview of the requirements and aspects of algorithm validation, this paper contains knowledge gathered through interviews mainly with researchers and engineers from the industry. The interviewees come from Germany, United States of America, China, Cuba, and Sweden, among other countries and have different technical backgrounds mostly in engineering science as mechanical, mechatronics, electrical and computer engineering. As they work on different topics around LIBs and BMS as ageing studies, electrochemical impedance spectroscopy, thermal control, and safety management, BMS functional development, battery modelling, numerical simulation of LIBs, machine learning, control, and optimization algorithms, they have a lot of experience in this field, partly from more than 20 years. In doing so, the majority focuses on software development and work just on interfaces with hardware. However, the scope comprises from cell level upwards to online electrochemical impedance spectroscopy measurement on pack level.

Due to the strong industrial reference of the topic, the researchers work relates to various commercial automotive applications, mainly on BEVs, HEVs and heavy-duty trucks. Besides of that, electric scooters, battery electric locomotives and golf carts are more uncommon applications, but also demand specific requirements on BMS algorithms. Off-road applications as in the aviation, underwater and marine sector are together with stationary grid scale and microgrid storages further applications for battery algorithms. Furthermore, second life applications of vehicle LIBs and vehicle grid integration are interfaces between automotive and other sectors. The guideline for the interviews with the researchers is found in Supporting Information.

Limitations of the method

The interviewees covering different technical backgrounds and country specific aspects, are mainly academic researchers with experiences in industry projects as engineers in industry are restricted in disclosing information due to non-disclosure agreements. The published requirements are a summarize of gathered personal opinions of the several researchers. Even if there was a big consensus, the opinion of individuals may differ. This paper tries to give an extensive overview considering the main aspects, but it is not possible to consider and mention every aspect and requirement. Furthermore, BMS algorithms and hardware are still under development, therefore an ongoing change in this topic is expected.

Creation of test scenarios

As shown in Figure 2 cell chemistries and geometries can differ. Different chemistries challenge the algorithm development due to their different characteristics and behaviours. In general, temperature has an impact on these characteristics. Single cells have initial settings as initial SoC and initial temperature, these both as well as the cell parameters can differ from cell to cell. [2] On module level, the single cells form a module by parallel and serial connections depending on the required power and energy. The cell or module is charged and discharged with different current levels and current durations. These are specified as power profiles and depend on the application.

Test cases for the validation of algorithms arise from the combination of the individual scenarios. Thereby, the combination possibilities rise exponentially, and create a large number of test cases.

In this paper, three different load profiles (for BEV and HEV with varied initial SoC) are applied for three different cells at three different temperatures. For simplification and comparability, the temperature of the environment and the cells are the same and are assumed to be constant. These selected scenarios represent just a small set of scenarios, but sufficient to demonstrate the complexity of validation. Conducting a comprehensive validation is not the aim of this paper.

To validate the influence of measurement errors, different sensor errors as offset, gain and noise for current and voltage measurement, respectively, are added and the results are compared with the simulations without sensor inaccuracies. The different parameters for the test scenarios are summarized in Table 1, thereby different scenarios were determined and selected for an exemplary validation.

Table 1: Variation of the parameters for the creation of the test scenarios

Test scenario	Parameter
Cell chemistries	NCA||C+Si (NCA)
	NMC||C (NMC)
	NMC||LTO (LTO)
Application	BEV
	HEV
Drive cycles	Restrained profile
	Sporty profile
Initial SoC	20 %
	50 %
	80 % (HEV) / 100 % (BEV)
Temperature	0 °C
	25 °C
	40 °C

Measurement error	Current noise, offset and gain	
	Voltage noise, offset and gain	
	Current and voltage noise, offset and gain	
Algorithms for SoC estimation	EKF	R1RC
		R2RC
	UKF	R2RC

In Figure S1, exemplary, the SoC curves for the different scenarios are plotted: The two different drive cycles differ in length and dynamic. The restrained and sporty speed profiles are extracted from Tewiele [5]. They are synthetically generated based on real collected driving data with a sporty and restrained driving style, respectively. Sporty means, that the SoC changes are bigger, including more break and acceleration phases than the restrained profile. In the graph, the SoC curves are presented for the LTO cell at 25 °C and initial SoC of 50 %.
The MiL environment requires a power profile as input. The simulated profiles are generated by a vehicle simulation tool developed at Institute for Power Electronics and Electrical Drives (ISEA, RWTH Aachen University). This tool contains models for BEV and HEV, respectively and uses velocity profiles as input to generate power profiles depending on vehicle type and battery configuration. Other settings that can be specified or selected include initial SoC, ambient temperature, electric motor characteristics, and HVAC conditions.
Depending on the cell chemistry and design, the resulting current, voltage and the SoC curve differ. As NMC||LTO chemistry is designed for high power applications, an application profile of a mild hybrid electric vehicle (MHEV) is used, i.e., there is no external charging, but the battery is charged only by recuperation, and the SoC oscillates in the middle SoC range from 43% to 69% at 25 °C and 50% initial SoC. For the other two cells, NMC||Graphite and NCA||Graphite, a BEV application profile is used; therefore, the SoC is decreasing during the trip. For the sporty profile, the NCA||Graphite cells discharges faster than the NMC||Graphite cell due to different cell capacities. The exemplary curves are shown for the sporty profile at 25 °C and starting at SoC of 50 %.
Not only the application and the driving cycles can vary, also the initial SoC is depending on the driver behaviour and the operational strategy. For the simulation of the MHEV with the NMC||LTO cell, the simulation starts with initial SoC of 20 %, 50 % and 80 %, as shown in Figure for the sporty profile at 25 °C, simulated with the LTO cell, but ends in the middle SoC range of 55 % to 81 % SoC. For the BEV simulation, the high initial SoC is set to 100 % due to the possibility of external full charge.
For considering the influence of temperature on the performance of the algorithms, three different temperatures, 0 °C, 25 °C and 40 °C, are chosen; exemplary the time series data of the sporty profile at 25 °C and starting at SoC of 50 % for the LTO cell are shown. With increasing temperature, the amount of usable capacity is increasing. At low temperatures, the SoC strokes are smaller due to limited charge and discharge currents. In the simulations presented in this paper, the cell and ambient temperature are both the same and constant over time to simplify and reduce the simulation effort without having to rely on a thermal model.
Additional to these test scenarios, considering the influence of measurement errors is necessary for the validation of BMS algorithms. Therefore, gain error, constant offset and Gaussian noise is considered on the voltage and current measurements, respectively. A detail of the current and voltage profile of the measurements with and without sensor inaccuracy is given for the sporty profile and the NMC||LTO cell at 25 °C with an initial SoC of 50 %. To reduce the scope of this paper, only the combination of all three kinds of errors is simulated, separately for voltage and current and for the combination of voltage and current inaccuracies. Additionally, the combination of current and voltage

sensor errors is investigated only for 50% initial SoC, and the single sensor accuracy influences only for 25°C and 50% initial SoC.
The sensor errors for the current and voltage measurement are generated as in Figure S2, respectively. Adding a sensor offset, a random number to generate noise and an error by constant gain to the simulated base measurement signal results in a noisy signal. The used error settings for noisy signal generation in this paper are given in Table S2.

Model-in-the-Loop environment
A model-based validation allows the comparison of diagnostic algorithms under same and repeatable but freely definable conditions and profiles. The flexibility of this simulation environment enables a comprehensive investigation of BMS algorithms for a variation of test scenarios.
The environment used for this work consists of several submodels: *Application*, *Reference Battery Model*, *BMS Model* and *Evaluation*.
The submodel *Application* uses the power of the selected application profile and scales it depending on the power of the simulated battery system. The resulting power demand is forwarded to the *Reference Battery Model*, based on a physical-chemically motivated ECM. The estimated SoC values of the algorithms are compared to the SoC generated by this battery model. These reference model derives from fitting cell characterization measurements, done on a test bench, to a complex ECM.
Different test scenarios can be simulated by changing:

- the power profile for different applications with different initial SoC,
- the system topology for different cell chemistries, designs, and interconnections,
- the initial cell and ambient temperature,
- and spread parameter spread of the single cell ECMs.

The simulated signals of voltage and current are forwarded to the *BMS Model* divided in hardware emulation and software. This model covers the electronics in a battery pack to map real application conditions as sensor inaccuracies. The adapted measurement signals go into the *Diagnostic Algorithms*, which is a submodel of the *BMS Software Model*. It includes the different state estimators, as well as the parameter identification. In this case, different Kalman filters and lookup tables for parameter estimation are used. The final step is the evaluation of the estimated states: comparing the respective signals from the BMS algorithms and the reference states.
In the simulations presented in this paper, the cell and ambient temperature are both the same and constant over time to simplify and reduce the simulation effort and a thermal model is not necessary. Further details about the MiL simulation environment can be found in [23].
Within the BMS framework, estimators for various states such as SoC, SoH, and SoP are integral components. Nonetheless, this study primarily delves into the simulation of SoC estimators, illustrating the methodology for constructing test scenarios and validating diverse algorithms. While the outlined procedure can be applied to other states, certain modifications might be required to accommodate the different estimation criteria used by each state.

The considered estimators are based on Kalman filter: the EKF and a sigma-point Kalman filter (SPKF), the UKF, EKF using a R1RC and R2RC battery model and UKF a R2RC. The initialization of the filter tuning parameters is the same for every test scenario, based on the same application profile for the NMC||LTO cell at 25 °C, even for every tested cell chemistry to validate the adaptability and the influence of initialization on the algorithms' performance. The parameter identification method for the R1RC and R2RC battery model is the recursive least square method. The development of the Kalman Filter and recursive least square method is not part of this work, and the implementation is used for the validation of the example. Further details about the principle of KF can be found in literature: In [3,16,24] Plett et al. proposes a method based on EKF and in [25,26] based on SPKF. The investigation of the implementation of 18 different KF [7] enables a comparative study and comparison.
Validation parameters

To evaluate and contrast the various algorithms, it's imperative to establish benchmark metrics. This study examines two specific error metrics: the RMSE and the AE. Both serve as indicators of a model's accuracy.
The RMSE compares the n values of an estimation $X_{Est,n}$ with its reference $X_{Ref,n}$ [27]:

$$RMSE = \sqrt{\frac{1}{n} \sum \left(x_{Est,n} - x_{Ref,n}\right)^2} \quad (1)$$

Compared to the mean absolute error, it is more sensitive to outliers because of the squaring of the error deviations. [28] Since the unit of the RMSE is the same as that of the model value, it is often preferred to the mean square error. For these reasons, the RMSE is used to validate the accuracy in this paper.
Additionally, the absolute error AE is calculated to compare the performance of the algorithms for different test scenarios. This error uses the deviation of the values of an estimation $X_{Est,n}$ with its reference $X_{Ref,n}$ in every estimation step n:

$$AE = \left|x_{Est,n} - x_{Ref,n}\right| \quad (2)$$

In addition to a mean error value, the AE is useful to directly detect outliers and to assess whether each estimate is within a tolerance band. Thereby, the maximum AE is the maximum value of all AE over the cycle. Both measures, RMSE and max. AE, are used in this paper to evaluate accuracy of the different Kalman filters for various test cases.
The exemplary validation results for accuracy are presented in a heat map for visualizing. The coloured illustration is intended to help compare the errors. On the y-axis, the different test cases as temperature, initial SoC and noise errors are plotted, on the x-axis the test cases as cells chemistries and driving cycles.

Acknowledgments:
First, the authors want to thank all interviewees: Javier Muñoz Alvarez, Alexander Blömeke, Wenjiong Cao, Weihan Li, Nico Manteuffel, Scott Moura, Simona Onori, Gregory Plett, Christopher Rahn, David Rosewater, Stefan Waldhör and Changfou Zou for their essential contribution. The authors also thank Matthias Faber, Stephan Bihn, and Katharina Quade for providing example cell parameters and Ziheng Wu for supporting algorithm development.

Supplementary Information is available for this paper.

Correspondence and requests for materials should be addressed to Franziska Berger and Philipp Dechent.

Disclaimer on the use of AI writing aids:

During the preparation of the manuscript the authors used GPT4/OpenAI in order to improve readability and language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

ORCID:
Franziska Berger: 0000-0002-3677-4564
Dominik Jöst: 0000-0002-5731-5666
Elias Barbers: 0000-0002-5836-2210
Ziheng Wu:
Katharina Quade: 0000-0001-6236-4115

Dirk Uwe Sauer: 0000-0002-5622-3591
Philipp Dechent: 0000-0003-3041-1436

Author contribution
F.B.: conceptualization, software, data curation, writing – original draft, visualization.
D.J: conceptualization, software, methodology, writing – review and editing.
E.B.: software, visualization.
K.Q.: software, writing – review and editing.
Z.W.: software, data curation.
D.U.S.: supervision, funding acquisition.
P.D.: conceptualisation, software, writing – original draft.

References

1. Wang, Z., Feng, G., Zhen, D., Gu, F. & Ball, A. A review on online state of charge and state of health estimation for lithium-ion batteries in electric vehicles. *Energy Reports* **7**, 5141–5161 (2021).
2. Klee Barillas, J., Li, J., Günther, C. & Danzer, M. A. A comparative study and validation of state estimation algorithms for Li-ion batteries in battery management systems. *Applied Energy* **155**, 455–462 (2015).
3. Plett, G. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 1. Background. *Journal of Power Sources* (2004) doi:10.1016/S0378-7753(04)00359-3.
4. Campestrini, C. *et al.* Validation and benchmark methods for battery management system functionalities: State of charge estimation algorithms. *Journal of Energy Storage* **7**, 38–51 (2016).
5. Tewiele, S. Development of Representative Driving and Load Cycles Based on Real-World Driving Data of Battery Electric Vehicles. (Universität Duisburg-Essen).
6. Zheng, Y., Ouyang, M., Han, X., Lu, L. & Li, J. Investigating the error sources of the online state of charge estimation methods for lithium-ion batteries in electric vehicles. *Journal of Power Sources* **377**, 161–188 (2018).
7. Campestrini, C., Heil, T., Kosch, S. & Jossen, A. A comparative study and review of different Kalman filters by applying an enhanced validation method. *Journal of Energy Storage* **8**, 142–159 (2016).

8. Neupert, S. & Kowal, J. Model-Based State-of-Charge and State-of-Health Estimation Algorithms Utilizing a New Free Lithium-Ion Battery Cell Dataset for Benchmarking Purposes. *Batteries* **9**, 364 (2023).
9. Farmann, A., Waag, W., Marongiu, A. & Sauer, D. U. Critical review of on-board capacity estimation techniques for lithium-ion batteries in electric and hybrid electric vehicles. *Journal of Power Sources* **281**, 114–130 (2015).
10. Galeotti, M., Cinà, L., Giammanco, C., Cordiner, S. & Di Carlo, A. Performance analysis and SOH (state of health) evaluation of lithium polymer batteries through electrochemical impedance spectroscopy. *Energy* **89**, 678–686 (2015).
11. Yuan, H.-F. & Dung, L.-R. Offline State-of-Health Estimation for High-Power Lithium-Ion Batteries Using Three-Point Impedance Extraction Method. *IEEE Trans. Veh. Technol.* **66**, 2019–2032 (2017).
12. Chaoui, H., Golbon, N., Hmouz, I., Souissi, R. & Tahar, S. Lyapunov-Based Adaptive State of Charge and State of Health Estimation for Lithium-Ion Batteries. *IEEE Trans. Ind. Electron.* **62**, 1610–1618 (2015).
13. Noura, N., Boulon, L. & Jemeï, S. A Review of Battery State of Health Estimation Methods: Hybrid Electric Vehicle Challenges. *World Electric Vehicle Journal* **11**, 66 (2020).
14. Dubarry, M., Baure, G. & Anseán, D. Perspective on State-of-Health Determination in Lithium-Ion Batteries. *Journal of Electrochemical Energy Conversion and Storage* **17**, (2020).
15. Waag, W., Fleischer, C. & Sauer, D. U. Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles. *Journal of Power Sources* **258**, 321–339 (2014).
16. Plett, G. L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 2. Modeling and identification. *Journal of Power Sources* **134**, 262–276 (2004).
17. Rubenbauer, H. & Henninger, S. Definitions and reference values for battery systems in electrical power grids. *Journal of Energy Storage* **12**, 87–107 (2017).

18. Zheng, L., Zhu, J., Wang, G., He, T. & Wei, Y. Novel methods for estimating lithium-ion battery state of energy and maximum available energy. *Applied Energy* **178**, 1–8 (2016).
19. Chaoui, H. & Mandalapu, S. Comparative Study of Online Open Circuit Voltage Estimation Techniques for State of Charge Estimation of Lithium-Ion Batteries. *Batteries* **3**, 12 (2017).
20. Marongiu, A., Nlandi, N., Rong, Y. & Sauer, D. U. On-board capacity estimation of lithium iron phosphate batteries by means of half-cell curves. *Journal of Power Sources* **324**, 158–169 (2016).
21. Groot, J., Swierczynski, M., Stan, A. I. & Kær, S. K. On the complex ageing characteristics of high-power LiFePO4/graphite battery cells cycled with high charge and discharge currents. *Journal of Power Sources* **286**, 475–487 (2015).
22. Lelie, M. *et al.* Battery Management System Hardware Concepts: An Overview. *Applied Sciences* **8**, 534 (2018).
23. Jöst, D., Zappen, H. & Sauer, D. U. METHODOLOGY FOR MODEL-BASED BATTERY MANAGEMENT ALGORITHM VALIDATION, BENCHMARKING AND APPLICATION-ORIENTED SELECTION. 5 (2019).
24. Plett, G. L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 3. State and parameter estimation. *Journal of Power Sources* **134**, 277–292 (2004).
25. Plett, G. L. Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs. *Journal of Power Sources* **161**, 1356–1368 (2006).
26. Plett, G. L. Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs. *Journal of Power Sources* **161**, 1369–1384 (2006).
27. Kuipers, M. *et al.* An Algorithm for an Online Electrochemical Impedance Spectroscopy and Battery Parameter Estimation: Development, Verification and Validation. *Journal of Energy Storage* **30**, 101517 (2020).
28. Berecibar, M. *et al.* Online state of health estimation on NMC cells based on predictive analytics. *Journal of Power Sources* **320**, 239–250 (2016).

Supporting Information:

Definitions of key parameters

Accuracy

In general, accuracy can be understood as a measure of how close a particular measurement comes to a specific quantity. It should not be confused with precision, which pertains to the dispersion of multiple measurements from a true value. In the context of BMS algorithms, accuracy is assessed by comparing the prediction or estimate to reality or the 'true' value. Importantly, in this paper, we distinguish between 'estimation' (using past and present data to infer the instantaneous value) and 'prediction' (using past and present data to forecast a future value).

The 'true' value used for comparison with the estimated value can never be known with absolute certainty, but assumptions can be made. It can be defined under laboratory conditions that return the battery to a reproducible state. For SoC estimation, the true value could be tracked on a test bench using Ampere-hour counting with laboratory equipment to minimize error sources. In industry, validation of the SoC estimator's accuracy involves determining the SoC for different application profiles, starting SoCs, and a specified number of cycles (usually five to ten). The deviation is then measured as the difference between the estimated SoC and the test bench SoC.

This process can be repeated for various test scenarios, resulting in a distribution of errors around the predicted value. Alternatively, driving profiles can be categorized into sections such as high and low dynamics, with the mean value calculated in each. This provides more information about dependency on the load profile and dynamics.

Different errors can be considered when evaluating accuracy. The error values can be averaged over a time window or driving profile, or the absolute value at each time point can be determined – a more complex implementation. In some cases, the worst case from several different cycles is used to evaluate unacceptable outliers due to safety risk. Accuracy validation of algorithms can also involve setting a threshold which the maximum absolute error or mean absolute error should not exceed, offering a measure of robustness accuracy. Accuracy errors are usually expressed in percentage terms.

Robustness

Robustness refers to a system's ability to remain stable when subjected to unexpected or non-constrained deviations. In control theory, stability indicates that a system does not blow up within a range of uncertainty. When applied to BMS algorithms, stability robustness is typically not a concern, but performance robustness is. For instance, if SoC and SoH estimators are run simultaneously, and the SoC estimator is continually updated with new ageing parameters, the SoH estimator must provide reliable estimates.

In a BMS application, it is preferable for algorithms to consistently produce plausible values, even if they are slightly inaccurate, as long as there are not any extreme outliers. This consistent performance without high error fluctuations denotes robustness. Robustness also pertains fit conditions under which a method fails. An algorithm is deemed robust fit can handle a wide array of conditions.

A robust algorithm shows less sensitivity to parameter uncertainties and mismatches, and is more tolerant of sensor noise and measurement inaccuracies. I.e., if the parameter estimation accuracy is low, state estimation has a higher accuracy and robustness requirement.

Robustness in algorithms is also related to dealing with nonsensical or contradictory measurement inputs and sensor noise, i.e. to what extent the algorithm is affected by them.

Robustness tests should consider the signals for current, voltage, and temperature both individually and collectively. Besides of measurement uncertainties that are larger than usual, robust algorithms can handle abnormalities, such as incorrect initial values, and still estimate consistently across conditions. A further aspect of robustness is the algorithm's ability to self-detect errors in its estimations.

In an industrial context, robustness often refers to whether an algorithm meets customer expectations and functions as anticipated. Validating the robustness of BMS algorithms could involve ensuring that a mean absolute error metric remains below a certain threshold over a diverse set of circumstances, rather than for a specific case. This set would include different application profiles, starting SoC values, initial errors, measurement accuracies, and environmental conditions like temperature.

Real-time capability

Real-time capability is the ability of a system or algorithm to execute tasks online within a specified cycle or at a designated frequency, with minimal acceptable deviation or jitter. This means an algorithm must be able to determine desired values simultaneously with operations and depending on how quickly the state variables change, rather than saving and processing data later.

Additionally, in context of signal processing, it is based on the Nyquist criterion: Running the system or algorithm at least twice as fast as the highest frequency component of the underlying process.

Real-time capability is divided into 'hard' and 'soft'. In hard real-time, jitter is minimal, and tasks must execute periodically. If an algorithm does not complete its task within the allotted time, the system's reliability is questioned. In soft real-time, however, jitter is less critical, and the algorithm has flexibility in execution time. A gradation between these two classifications is the handling of no provided reliable estimates over a longer time.

Real-time capability is crucial for embedded/on-board systems, where response time is critical. In contrast, cloud applications have different real-time requirements as more advanced hardware is available, but also the focus for on-board systems has shifted from raw real-time capability to computational efficiency. Factors such as the number of states and sampling rate play a role in real-time considerations, for instance, scaling from one cell to 100-1000 cells with complex estimators may not be practical.

When testing algorithms for real-time performance in a lab, it's essential to use the exact hardware intended for application to replicate true operating conditions. In simulations, computations must ideally be faster than the actual timeline; i.e., a one-second model computation should be completed in less than one second. This underscores the importance of code optimization over hardware capabilities in this context.

Further Aspects

Comparing algorithms related to their complexity, e.g., the number of model or algorithm input variables is an indicator. In research, algorithm complexity is incidental, whereas it is more important in industrial applications where cost and hardware requirements are paramount. Higher number of inputs can increase the implementation effort and possible error sources. Faulty inputs as estimated capacity, which is used, e.g., for an Ampere-hour counter, and noisy measurements inputs can influence the performance of the algorithm and increase its estimation error. In addition to noisy inaccuracies, further possible impacts by sensors are failure over a few cycles, offset and scaling effects.

When developing algorithms from research perspective, there are no real hardware constraints, because of the available hardware in the lab as high-accuracy sensors and high-performance processors. In real applications, there are limitations by hardware, but on the other side the algorithms depending on the implementation make demands on the hardware, such as due to CPU utilization, RAM and ROM memory requirements, request rate at which the algorithm is executed and number of the interfaces the algorithm has. In addition, considering the iterations steps is important because of possible hardware limitations, so that the states of only one or a few cells are estimated instead of all cells in a module or pack and the duration of an iteration step in the actual system is depended on the computing power of the hardware.

Scalability, e.g., from cell to pack is a further aspect, because in vehicle application we usually have modules consisting of several cells and especially in use of heavy-duty long hauls, there are even many packs. In this case, scalable could mean that the computation and sensing requirements scale sub-linearly with the number of cells. A second aspect due to higher numbers of cells is the dynamic

(re)configuration capability that means the possibility to skip cells immediately, when they are damaged without disassembling the whole pack or to select optimal cells, e.g., less aged cells. This would be a chance to increase the performance of battery packs in operation but at the same time places high demands on the algorithms.
The next requirement is related to the section on accuracy. There can be initial errors on the inputs. Among other, for some cell chemistries as LFP cells, it is difficult to determine the OCV value during an insufficient relaxation or by sensor inaccuracies due to the flat OCV curve and thus the initial value for e.g., SOC estimation value is faulty. To validate the duration until convergence, i.e., how quickly do the algorithm get back to the correct value in case of incorrect initialization, and the adaptability of the algorithms, the initial value is set wrong, and behaviour of aged cells is considered. This results in an answer to the question in which time frame we can expect an output from the algorithm.
As battery cells behave differently depending on the chemistry, e.g., NMC and LFP based cells, they are also very different in accuracy and robustness. Making the same demands on these two cell chemistries can be a challenge. Testing the developed algorithms for each cell chemistry separately is indispensable.
As already mentioned in context with robustness, a confidence factor indicates the algorithm itself and the user how much trustworthy an estimate is and how big the deviation could be. Furthermore, it is about whether the estimator basically trust itself and to what extent would it pretend to trust the superior control unit. Kalman filters, for example, automatically produce a confidence value for the estimate. In addition to the robust performance of algorithms, robust optimization is applicable in this regard. This kind of optimization means to taking the worst-case scenario and optimizing the method for this case. Compared to a strict optimization, the robust optimization has better outcome because of planning for the worst-case outcome. Optimizing to a specified portion of distribution is known as risk averse optimization in control theory. I.e., if you have a Gaussian distribution and you want to optimize to the e.g., ninetieth percentile, you set up your random variables in the optimization to drive the outcome to some value at or greater than that 90% outcome. This offers flexibility because any distribution is possible but is concurrent data-driven and depending.
Each research project may have its own unique requirements, but in general, the research objectives are SMART: Specific, Measured, Achievable, Realistic/Relevant and Time-Bound. These are universally applicable and not algorithm specific. Before developing a method or model, the first crucial step is writing down the objective, which can be SMART. The idea behind that SMART goals is to give criteria for giving better results while focusing the efforts.

In general, the requirements differ according to the algorithms for different state estimation. This paper focuses on the specific requirements on SoC, SoH and SoP estimation, however, the first two being the more discussed ones. After defining the requirements as accuracy, robustness, and real-time capability, they are discussed more in detail.

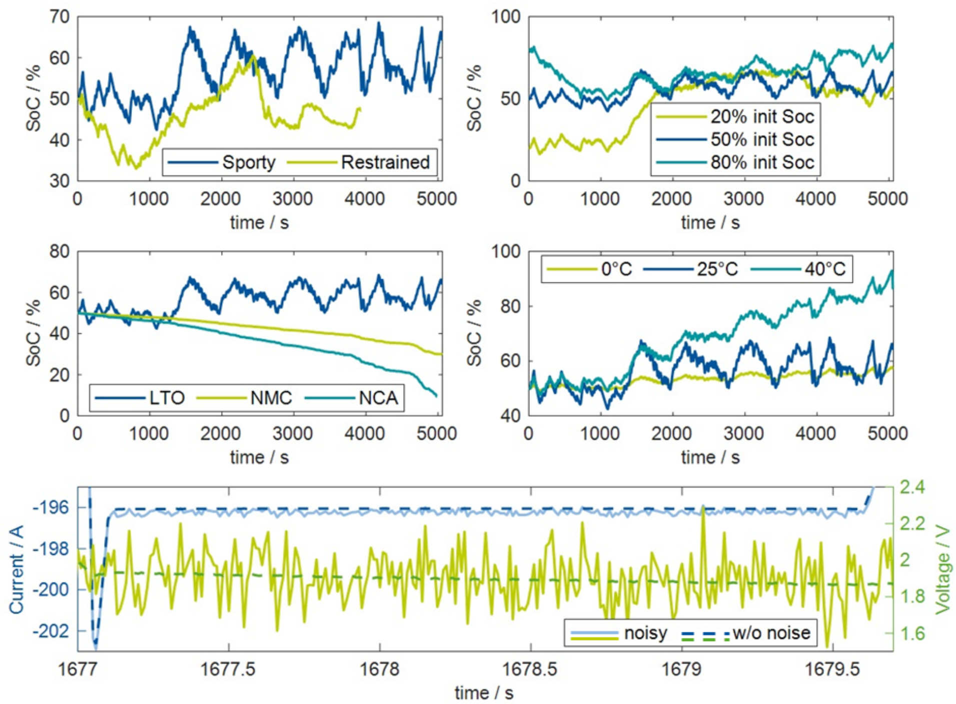

Figure S1: Driving profiles for different cells, initial temperatures, initial SoCs and measurement inaccuracy

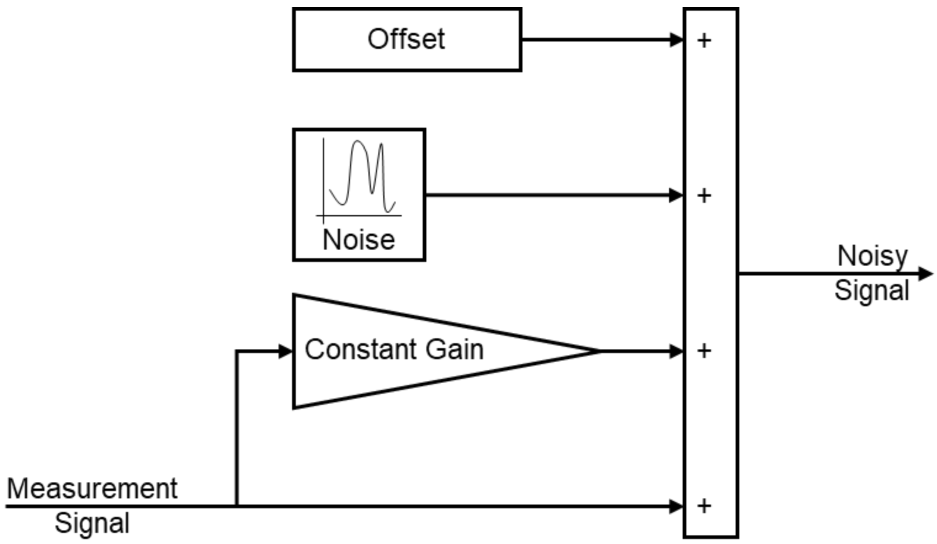

Figure S2: Creation of noisy measurement signal in simulation

Table S2: Error settings for noisy signal generation

Sensor errors		Current	Voltage
Offset		20 mA	3 mV
Constant Gain		0.1%	
Noise (Random Number)	Mean	0	
	Variance	0.012	0.018
	Seed	0	
	Sample Time	10 ms	

For different BEV drive cycles (restrained and sporty) and NCA||Graphite						
EKF-R1RC		Error	RMSE		max. AE	
		Drive cycle	Restrained	Sporty	Restrained	Sporty
Temp.	Init. SoC	Sensor				
0°C	20%	No noise	2.54	2.71	>5	4.55
	50%	No noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	100%	No noise	>3	>3	>5	>5
25°C	20%	No noise	3.00	2.93	4.24	>5
	50%	No noise	>3	>3	>5	>5
		I noise	>3	>3	>5	>5
		U noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	100%	No noise	>3	>3	>5	>5
40°C	20%	No noise	>3	>3	>5	>5
	50%	No noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	100%	No noise	>3	>3	>5	>5

EKF-R2RC		Error	RMSE		max. AE	
		Drive cycle	Restrained	Sporty	Restrained	Sporty
Temp.	Init. SoC	Sensor				
0°C	20%	No noise	>3	1.71	>5	>5
	50%	No noise	1.54	2.11	>5	>5
		U & I noise	>3	>3	>5	>5
	100%	No noise	1.43	>3	>5	>5
25°C	20%	No noise	1.19	1.76	>5	>5
	50%	No noise	0.92	1.88	3.8	4.07
		I noise	1.43	1.89	>5	>5
		U noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	100%	No noise	0.57	1.74	>5	>5
40°C	20%	No noise	1.61	1.91	>5	>5
	50%	No noise	0.78	2.15	3.11	3.91
		U & I noise	>3	>3	>5	>5
	100%	No noise	0.7	1.24	4.52	>5

UKF-R2RC		Error	RMSE		max. AE	
		Drive cycle	Restrained	Sporty	Restrained	Sporty
Temp.	Init. SoC	Sensor				
0°C	20%	No noise	>3	0.98	4.82	1.45
	50%	No noise	2.05	1.26	2.59	1.66
		U & I noise	>3	>3	>5	>5
	100%	No noise	0.61	1	2.52	3.22
25°C	20%	No noise	1.21	1.17	1.96	1.7
	50%	No noise	0.71	0.88	1.25	1.35
		I noise	2.11	1.55	4.61	4.75
		U noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	100%	No noise	0.39	0.72	1.96	2.44
40°C	20%	No noise	1.37	1.27	1.78	1.69
	50%	No noise	0.68	0.91	1	1.53
		U & I noise	>3	>3	>5	>5
	100%	No noise	0.61	0.87	1.97	2.59

Figure S3: Heatmap of simulation results of NCA for different profiles

For different BEV drive cycles (restrained and sporty) and NMC||Graphite

EKF-R1RC		Error	RMSE		max. AE	
		Drive cycle	Restrained	Sporty	Restrained	Sporty
Temp.	Init. SoC	Sensor				
0°C	20%	No noise	>3	>3	>5	>5
	50%	No noise	>3	>3	>5	4.95
		U & I noise	>3	>3	>5	>5
	100%	No noise	1.16	1.31	1.78	2.36
25°C	20%	No noise	0.55	0.19	2.56	0.83
	50%	No noise	0.16	0.03	0.6	0.24
		I noise	0.14	0.07	0.54	1.4
		U noise	0.71	0.64	1.8	1.22
		U & I noise	0.67	0.61	1.75	1.35
	100%	No noise	0.26	0.13	0.65	0.55
40°C	20%	No noise	1.5	1.48	2.88	2.94
	50%	No noise	0.46	0.47	0.6	0.69
		U & I noise	0.39	0.28	1.49	0.87
	100%	No noise	0.67	0.6	1.33	0.92

EKF-R2RC		Error	RMSE		max. AE	
		Drive cycle	Restrained	Sporty	Restrained	Sporty
Temp.	Init. SoC	Sensor				
0°C	20%	No noise	>3	>3	>5	>5
	50%	No noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	100%	No noise	2.11	>3	>5	>5
25°C	20%	No noise	0.64	0.41	4.77	3.27
	50%	No noise	1.18	0.7	>5	3.87
		I noise	1.07	0.51	>5	>5
		U noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	100%	No noise	1.44	1.13	>5	>5
40°C	20%	No noise	0.78	1.22	2.91	3.23
	50%	No noise	0.97	0.47	>5	3.54
		U & I noise	>3	>3	>5	>5
	100%	No noise	1.19	1.07	4.64	>5

UKF-R2RC		Error	RMSE		max. AE	
		Drive cycle	Restrained	Sporty	Restrained	Sporty
Temp.	Init. SoC	Sensor				
0°C	20%	No noise	>3	>3	>5	>5
	50%	No noise	>3	>3	5	4.12
		U & I noise	>3	>3	>5	>5
	100%	No noise	0.65	1.28	1.97	2.03
25°C	20%	No noise	0.27	0.21	0.89	0.47
	50%	No noise	0.94	0.5	1.92	1.66
		I noise	0.9	0.63	4.93	4.93
		U noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	100%	No noise	1.03	0.49	2.41	2.69
40°C	20%	No noise	0.73	1.14	1.43	1.48
	50%	No noise	0.62	0.3	2.35	0.75
		U & I noise	>3	>3	>5	>5
	100%	No noise	0.87	0.76	2.45	2.52

Figure S4: Heatmap of simulation results of NMC for different profiles

For different HEV drive cycles (restrained and sporty) and NMC||LTO

EKF-R1RC		Error	RMSE		max. AE	
		Drive cycle	Restrained	Sporty	Restrained	Sporty
Temp.	Init. SoC	Sensor				
0°C	20%	No noise	2.36	>3	>5	>5
	50%	No noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	80%	No noise	>3	2.41	>5	>5
25°C	20%	No noise	>3	>3	>5	>5
	50%	No noise	>3	>3	>5	>5
		I noise	>3	>3	>5	>5
		U noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	80%	No noise	>3	>3	>5	>5
40°C	20%	No noise	>3	>3	>5	>5
	50%	No noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	80%	No noise	>3	>3	>5	>5

EKF-R2RC		Error	RMSE		max. AE	
		Drive cycle	Restrained	Sporty	Restrained	Sporty
Temp.	Init. SoC	Sensor				
0°C	20%	No noise	1.92	1.68	>5	>5
	50%	No noise	0.96	2.26	1.66	3.73
		U & I noise	>3	>3	>5	>5
	80%	No noise	>3	1.87	>5	>5
25°C	20%	No noise	2.23	2.42	>5	>5
	50%	No noise	1.06	1.81	2.88	4.98
		I noise	1.27	1.57	3.29	4.74
		U noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	80%	No noise	1.95	1.94	>5	4.23
40°C	20%	No noise	>3	>3	>5	>5
	50%	No noise	2.11	>3	4.35	4.67
		U & I noise	>3	>3	>5	>5
	80%	No noise	>3	2.18	>5	>5

UKF-R2RC		Error	RMSE		max. AE	
		Drive cycle	Restrained	Sporty	Restrained	Sporty
Temp.	Init. SoC	Sensor				
0°C	20%	No noise	0.63	1.29	3.02	>5
	50%	No noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	80%	No noise	1.57	0.53	4.48	3.42
25°C	20%	No noise	2.58	2.08	>5	>5
	50%	No noise	1.41	1.57	2.5	3.59
		I noise	2.13	1.2	4.33	3.13
		U noise	>3	>3	>5	>5
		U & I noise	2.61	>3	>5	>5
	80%	No noise	0.68	0.37	0.97	0.78
40°C	20%	No noise	1.31	2.36	2.31	3.58
	50%	No noise	>3	>3	>5	>5
		U & I noise	>3	>3	>5	>5
	80%	No noise	0.66	0.7	3.1	1.16

Figure S5: Heatmap of simulation results of LTO for different profiles

Interview concerning requirements for BMS Algorithms

- Introduction of the interviewee to understand relation of answers
  - Background (studies, current work, working group, projects)
  - Depth of research / development
  - Hardware or software
  - Applications (e.g. vehicles, stationary storage)
  - Industry reference or research level
- Are the answers related to all three algorithms SOP, SOC and SOH or is one directly excludable?
- **Definitions**: Accuracy, Robustness, Real-time Capability
  - How would you define accuracy, robustness, and real-time capability related to BMS algorithms?
- What are **important parameters** when talking about requirements for algorithms, additional to the three mentioned?
- How **accurate** does an algorithm need to be? Over what period of time is a certain error tolerance acceptable? Does an algorithm need to be **real-time capable** and how is this determined? (Are time windows considered at all?)
  - Real-time capability: differentiation between function of the algorithm (e.g. SOC vs. SOH)
  - Update interval
- (Dependent on the background of the interviewee)

What are the limitations of **hardware** or what influence does hardware have on our requirements, accuracy, evaluation, e.g. measurement accuracy?

  - Influence on evaluation, accuracy, requirements, etc.?
  - E.g. sensors, sampling rate
- Which **sampling rate** is required, i.e. how often must be measured and how often the state determined?
  - Which values are realistic from hardware side / limitations due to hardware?
  - Are there limitations due to memory?
- What **historical data** needs to be available?
  - From the estimator - for the estimator
  - E.g. start values, time windows
  - Memory
- How **robust** does an algorithm need to be, to temperature variations, sensor failure and sensor error (noise, offset), different usage profiles?
  - When must reliable results be delivered? à Safety aspect
  - How big may the environmental influences be, so that the algorithm is still robust?
  - Are unusual usage scenarios taken into account, or does the algorithm have to cover them (e.g. second use)?
- Is **calibration** or under what user effort acceptable?
  - e.g. with full charge at SOC
  - How often is ok?
  - How high may the error be over what time until calibration?
  - Full charge, 15 min. break, active intervention of the user / workshop
- How are **application scenarios** defined, which ones are considered during validation and design?
  - Consideration of load profiles, also "untypical", real vs. synthetic?

  - For which ones were the following questions answered?
  - Large differences of the answers depending on the application (load profile, user, vehicle)
- What **criteria** are used to **select** an algorithm?
  - E.g. Which SOH is the most important and why?
  - Relation to key figures

Optional:

- Do the requirements for an algorithm change over **ageing states**?
  - If yes, what changes
  - Dependence on the considered algorithm

à e.g. performance limitation only required exactly from a certain ageing state onwards

- How **automated** does an algorithm need to adapt to the specific application?
  - Focus: better many very specific / slightly different or universal
  - Fully automatic, only parameterization/calibration, change to the operating principle
  - Specific for one battery chemistry
  - For one application
  - Always self-calibrating

- **Current state, current and future goals?**
  - Current state of research
  - Research and industry objectives (own, in general)
  - What kind of algorithms are considered to be
    - target-oriented for industrial implementation
    - state of the art with potentially highest accuracy