Kalman Filter Tuning for State Estimation of Lithium-Ion Batteries by Multi-Objective Optimization via Hyperspace Exploration

Patrick Mößle, Tobias Tietze, and Michael A. Danzer*

For the estimation of the state of charge of lithium-ion batteries Kalman filters are the state of the art. To ensure precise and reliable estimations these filters use covariance matrices, which need to be tuned correctly by the developer. This process is time-consuming and depends largely on the experience and skill of the developer. Hence, filter tuning is not reproducible and not optimal with regard to goals as accuracy and convergence speed. Herein a multiobjective optimization framework called hyperspace exploration is used for the first time to automate the filter tuning procedure for an extended Kalman filter and two versions of adaptive extended Kalman filters. Four key performance indicators, including the maximum error in the estimation of the state of charge and the according root mean square error, are used to describe, validate, and compare the filter performance. This automated process enables optimal usage of the degrees of freedom in filter tuning and no longer requires manual tuning while the whole hyperspace, including different use cases and validation scenarios, is considered in the optimization. Furthermore, the proposed approach yields a novel method for the evaluation of filter parameters and their influence on the estimation behavior.

1. Introduction

The user of a product powered by a lithium-ion battery (LIB) has the expectation of being able to retrieve information about the current state of charge (*SOC*) and thus fundamental information about the remaining usage time or driving range at any time. In contrast to physical quantities such as voltage or current, the *SOC* is a state that cannot be measured directly, but must be estimated using calculation methods.[1] Hence, a considerable computational effort is required to obtain precise information about the remaining battery capacity. A variety of approaches has been developed to estimate the *SOC*.[2] For a detailed review on different battery state estimation approaches, refer to Wang et al.[3] According to the current state of the art, the estimates of the states are realized by mathematical descriptions of the system behavior in combination with filter algorithms. Based on measurable input data, the states can thus be determined via recursive equations. During operation of the physical system, these calculation steps are repeated continuously. Thus, current information about the desired system state is available at any time.[4] The Kalman filter (KF) or KF variants are most commonly used for this purpose.[3,5–8] Shrivastava et al.[9] show a combined *SOC* and state of energy estimation using the dual forgetting factor (FF) adaptive extended Kalman filter (AEKF). Additionally, the authors establish a new sliding window-approximate weighted total least square algorithm for the estimation of the maximum available capacity and maximum available energy estimation. Other approaches like from Zhang et al.[10] utilize a gated recurrent unit neural network to determine the *SOC*. Wang and Chen[11] propose an unscented particle filter for *SOC* estimation and compare their results to an extended KF (EKF).

The accuracy of the estimation by the KF depends on the choice of its filter parameters (covariances). The covariance matrices of the error, process noise, and measurement noise determine the performance of the KF. Modifications of these parameters have a large influence on the convergence speed and precision of the estimation. Adjusting the three mentioned quantities is called filter tuning. The determination of these parameters usually turns out to be difficult. On the one hand, they can vary in a very wide range. A comparison of different values from the literature shows how strongly the parameters for filter tuning can deviate. While in ref. [6] the covariance matrix is initialized with values down to 10^{-12}, in ref. [12] this

P. Mößle, M. A. Danzer
Chair of Electrical Energy Systems (EES)
University of Bayreuth
Universitätsstraße 30, 95447 Bayreuth, Germany
E-mail: patrick.moessle@uni-bayreuth.de

P. Mößle, T. Tietze, M. A. Danzer
Bavarian Center for Battery Technology (BayBatt)
Universitätsstraße 30, 95447 Bayreuth, Germany

T. Tietze
Chair of Systems Engineering for Electrical Energy Storage (SysEE)
University of Bayreuth
Weiherstraße 26, 95448 Bayreuth, Germany

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/ente.202300796.

DOI: 10.1002/ente.202300796

Table 1. Kalman filter steps.

Initialization	$\tilde{\boldsymbol{x}}_0 = \mathbb{E}[\boldsymbol{x}_0]$; $\tilde{\boldsymbol{P}}_0 = \mathbb{E}(\boldsymbol{x}_0 - \tilde{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \tilde{\boldsymbol{x}}_0)^{\mathrm{T}}$
State estimation	$\hat{\boldsymbol{x}}_{k+1} = \boldsymbol{A}_k \times \boldsymbol{x}_k + \boldsymbol{b}_k \times u_k$
Covariance estimation	$\hat{\boldsymbol{P}}_{k+1} = \boldsymbol{A}_k \tilde{\boldsymbol{P}}_k \boldsymbol{A}_k^{\mathrm{T}} + \boldsymbol{Q}$
Kalman gain	$\boldsymbol{L}_{k+1} = \hat{\boldsymbol{P}}_{k+1} \boldsymbol{C}_k^{\mathrm{T}} \left[\boldsymbol{C}_k \hat{\boldsymbol{P}}_{k+1} \boldsymbol{C}_k^{\mathrm{T}} + r\right]^{-1}$
State update	$\tilde{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1} + \boldsymbol{L}_{k+1}[y_k - \boldsymbol{C}_k \hat{\boldsymbol{x}}_{k+1} - \boldsymbol{C}_k u_k]$
Covariance update	$\tilde{\boldsymbol{P}}_{k+1} = (\boldsymbol{I} - \boldsymbol{L}_{k+1} \boldsymbol{C}_k) \hat{\boldsymbol{P}}_{k+1}$

is done with values up to 100. On the other hand, there is no known method for direct calculation of the parameters for use in battery state estimation. Often, the determination of the filter parameters is carried out by a trial-and-error method and the success depends on the experience of the developer.[13,14]

An exact determination of the covariances of the process noise and the measurement noise (see Equation (6) and **Table 1**) required for the KF is one of the major challenges for the design of a KF. According to Abbeel et al.,[15] the noise parameters usually result from a variety of different effects: system and measurement dynamics are not accurately represented, hidden states exist in the environment that are not considered in the model used by the KF, discretization errors of the measurement, and approximation errors, e.g., in the Taylor series approximation for the linearization in an EKF.

Ting et al.[16] propose an evolutionary algorithm to determine these parameters. Atukalp and Kink[17] have developed an analytical tuning method that calculates the noise parameters dynamically by identifying the main contributors to those uncertainties formulating them as probability density functions and combining them to a single variance. The contributing parts to those uncertainties are then varied using a Monte Carlo method. The authors verify their results by comparing them with Coulomb counting of highly precise measurements. Their *SOC* estimation shows a root mean square error (*RMSE*) of under 1.5% and a maximum error below 2.8%. Theiler et al.[13] present a genetic algorithm for filter parameter tuning. They use dynamic load profiles and compare simulated states with estimated states using data with added noise by predefined uncertainties. The covariance matrices are subject to a multi-objective optimization (MOO) by applying a genetic algorithm based on the survival of the fittest principle. The authors show the feasibility of the proposed method. Wang et al.[18] also show the applicability of genetic algorithms for this purpose by implementing a distributed evolutionary algorithm in Python with considerable success.

In practice, however, the matrices are usually manually adjusted to the desired conditions by matching them with reference data for the states to be estimated. The reference data comes from precise measurements from which the estimated states can be determined and is only available at this level of accuracy during the development phase. The developer then adjusts the tuning parameters until the KF reproduces the measured values with the desired behavior. In such an approach, the experience of the user plays a considerable role.[14,15] Campestrini[14] formulates certain rules of thumb that can guide the programmer: 1) if the diagonal elements of the covariance matrix of the process noise have value 0, the filter assumes a perfect model and does not correct the states. If, in contrast, the values are high, the estimation errors increase because the filter assumes that at each time point uncertainties affect the states. Here, a trade-off must be made between the ability to correct incorrect values and the magnitude of the estimation error. 2) The convergence behavior of the filter is controlled by the error covariance matrix.[19,20] When its diagonal elements are zero, the filter assumes exact initial values and adjusts incorrect initial values only slowly. Too high values can cause unstable filter behavior. Here, a compromise between convergence speed and stability must be found. 3) For low values of the measurement noise variance, the filter assumes high measurement accuracy, and it calculates states based on the measured values rather than the model. At high values, the Kalman gain decreases because the measured values are not trusted enough, and the estimate follows the model without noticeable correction. Again, a trade-off must be made.

In this article, two approaches are tested and compared: 1) the already established adaptive KF (AKF), which adjusts the filter parameters during operation based on the deviations between estimation and measurement. 2) A framework for system design, the hyperspace exploration (HSE), is adapted to automate the process of adjusting the filter parameters. The HSE is applicable in this context, since it was developed for analyzing and solving complex optimization problems.[21]

It is an evolution of the MOO methods introduced by the groups mentioned before and presents a holistic approach with high potential for automation. With this automated process, the filter covariance parameters no longer require manual tuning and the method presents a unique way for the evaluation of filter parameters and their influence on the estimation results. The HSE finds the best covariance values for the specific application in regard to the desired filter behavior. The framework also takes into account different use-case scenarios, allowing for the identification of use-case specific individual trade-offs between target indicators. In addition to that, it makes the evaluation of trade-offs accessible through the utilization of a surrogate model.[21]

To the best of the authors' knowledge, this work is the first that employs the HSE framework to the optimal tuning of a KF.

2. State Estimation and Filter Parameter Determination

Battery states must be calculated from measurable quantities about which information is available at any time. In addition to the *SOC*, the available power is also of interest to the battery management system (BMS), as is the state of health (SOH) for predicting the remaining service life, which also requires an estimate of the battery capacity. This changes with the lifetime of the cell due to aging effects.[22]

To calculate these states, the cell parameters must be determined from a suitable battery model and the measured variables current *I*, voltage *U*, and temperature *T*. For more details about the interaction of the individual estimated states within a BMS see Waag.[22] In the following, two commonly used estimation approaches are briefly discussed.

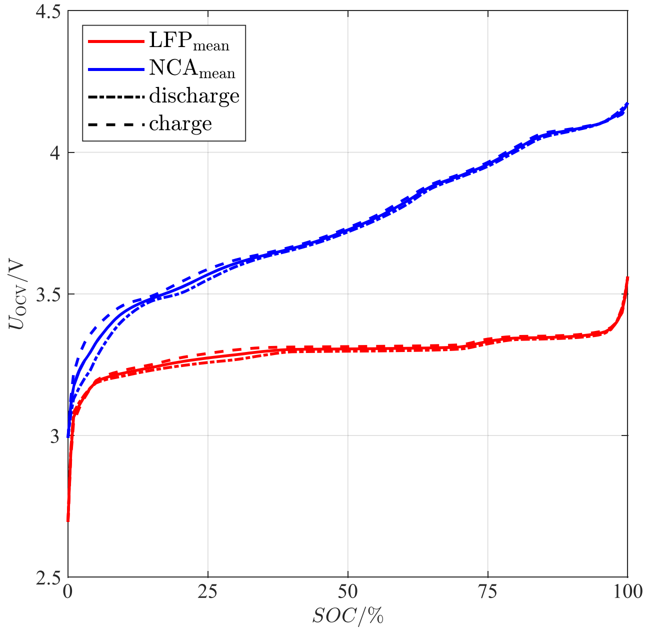

Figure 1. Dependency of the *SOC* on the U_{OCV} for an NCA and a lithium iron phosphate cell.

2.1. *SOC* Estimation

The open-circuit voltage (U_{OCV}) method uses the dependence of U_{OCV} on the *SOC* (see **Figure 1**). A look-up table (LUT) can be generated from measured U_{OCV} data, from which the *SOC* can be determined for the respective voltage value. In this procedure, the U_{OCV} curve is formed from the mean value of the charge and discharge curves to account for hysteresis effects.[23] For cell chemistries with a low slope for U_{OCV}, such as lithium iron phosphate, voltage deviations of only a few can cause an *SOC* error of 10%.[24] This method is not suitable as the sole *SOC* determination method for an LIB to which a load is applied, because it does not take into account dynamic losses. Thus, it is only useful for applications with long relaxation periods, and even then it is not always guaranteed that the cell is completely at rest.[25,26] The most common method for determining the *SOC* is the so-called Coulomb counting. It calculates the amount of charge added and removed from a cell based on an integration of the measured charge and discharge current over time. From this amount of charge, the

$$SOC = SOC_0 - \int_0^t \frac{\eta_C \times I(\tau)}{C_n} d\tau \tag{1}$$

is calculated.

For this, the start value SOC_0 at the beginning of the current measurement must be known as accurately as possible (see Equation (1)). Inaccuracies in the current measurement accumulate through integration, so that the estimated value deviates further and further from the real value. A reset on known events can increase accuracy. By inserting the Coulomb efficiency η_C, the current losses are taken into account. Coulomb counting is a simple method that is easy to implement with high potential for inaccuracies, especially if the current measurement is not precise enough.[25,26]

Model-based estimation combines the two aforementioned methods by not deriving the *SOC* exclusively from one measurable variable (voltage for the U_{OCV} method, current for Coulomb counting), but combining both information. Based on a model of the cell, this approach calculates the internal and non-measurable states of the system from the directly measurable quantities of current, terminal voltage, and temperature.[22] This is then used to calculate the theoretical terminal voltage. A comparison of the calculated terminal voltage with the voltage actually measured at the real cell represents a criterion for the accuracy of the estimate. However, the difficulty with model-based approaches is to properly evaluate the voltage deviation. It can not only be due to estimation errors that should be corrected, but also measurement noise and/or modeling errors.[25]

2.2. Battery Model

Depending on the requirements, the behavior of a battery cell can be modeled with varying degrees of accuracy. In case of the BMS, methods are preferable that represent the physical behavior with sufficient accuracy and moderate computational effort.[27] According to Campestrini,[14] black box models, which do not require knowledge of the exact physical processes, are one solution for BMS applications. Their parameters are no longer physically interpretable, which is why this method is used for diagnostic purposes, rather than for accurate mapping of the cell's input and output behavior. It is advantageous that the determination of the parameters can be fully automated.[28] Black box models include neural networks,[29,30] fuzzy logic,[31] and stochastic models.[32,33]

Equivalent circuit models (ECM) are a popular alternative to more complex models. Almost all state-of-the-art BMS make use of them. This method of describing terminal behavior is particularly suitable because of its simplicity and robustness, and it represents the cell properties sufficiently accurately for the state estimates.[34] The basic idea of these models is to imitate the real cell behavior by an electrical circuit. The ECM can only represent the input and output behavior, but not cell-internal electrochemical processes. In this method, the model parameters are fitted to empirical data in the time domain. For load profiles that deviate strongly from the parameterization profile, extrapolation must be performed, resulting in deviations between predicted and real cell behavior.[25]

The Thevenin model is often used to describe a battery cell. It is a simple ECM consisting of a voltage source, a series resistor and an resistor–capacitor (RC) element connected in series. This model can be extended by any number of RC elements.[35] The corresponding ECM is shown in **Figure 2**. In the present work, the direction of the current *I* is defined that the cell is charged when the sign is positive and discharged when the sign is negative. The mathematical description of the ECM can be expressed in form of a discrete state space representation using the Euler method. For this purpose, it is first necessary to define the input and output variables as well as the states that are to describe the system. The cell current *I* serves as the input variable and the cell

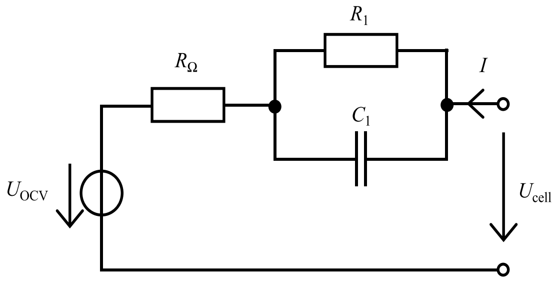

Figure 2. Equivalent circuit model (ECM) with one RC element.

voltage U_{cell} as the output variable. Both are directly measurable, which allows later a comparison between calculated and measured cell voltage. States that are not directly measurable are chosen as state variables. On the one hand, the voltage across the RC element must be taken into account to calculate the total cell voltage. On the other hand, states can be implemented that are to be additionally estimated such as the SOC of the cell. Thus, the state vector

$$\boldsymbol{x}_k = \left[U_{RC,k}, SOC_k\right]^{\mathrm{T}} \tag{2}$$

is defined. With the difference equation of the RC element

$$U_{RC,k+1} = \left(1 - \frac{\Delta t}{RC}\right) \times U_k + \frac{\Delta t}{C} \times I_k \tag{3}$$

and after discretization of the SOC in Equation (1) the state space representation

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \begin{bmatrix} U_{RC,k+1} \\ SOC_{k+1} \end{bmatrix} = \boldsymbol{A}_k \times \boldsymbol{x}_k + \boldsymbol{b}_k \times u_k \\ &= \begin{bmatrix} \left(1 - \frac{\Delta t}{RC}\right) & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} U_{RC,k} \\ SOC_k \end{bmatrix} + \begin{bmatrix} \frac{\Delta t}{C} \\ \frac{\eta_{\mathrm{C}} \times \Delta t}{C_{\mathrm{n}}} \end{bmatrix} \times I_k \end{aligned} \tag{4}$$

follows. From the state vector, the input current and the U_{OCV} corresponding to the SOC at time $k \times \Delta t$, the response equation for the state space representation of the battery model

$$y_k = U_{\mathrm{cell},k} = U_{\mathrm{OCV}}(SOC_k) + U_{RC,k} + I_k \times R_\Omega \tag{5}$$

can be derived. In this case, the term y_k is equal to the cell voltage.

2.3. Extended Kalman Filter

The measured values often cannot be exactly reproduced by the model. These differences are unavoidable due to measurement inaccuracies, incorrect model parameters, and the equations for the state itself.[12] Another possibility in addition to the measures already mentioned in Section 2.1 to minimize the error is the use of state observers. They are built on the concept of model-based estimation. Based on the state space model, the state observer continuously updates and corrects the state Equation (4), in this case online. Different algorithms can be used for this purpose. The Luenberger observer is mainly used for linear control systems, while the sliding-mode observer is used for nonlinear behavior. A third possibility is the so-called KF.[36] The KF offers the advantage that it works recursively and, in contrast to filters that average over an interval of measured values, it requires only the last and the current value of the measurement. This makes its use suitable for estimation in real-time applications.[25] The KF relies on the difference between Equation (4) and the output Equation (5) of the model to predict the state of a physical process.[14] The deviation from the measured system output adjusts the estimates. Accordingly, it is based on minimizing the mean-squared error.[25] For efficient computation by a BMS, it is convenient to use the discrete model.[4] The state equation of a KF is generally extended by the process noise $\boldsymbol{w}_k$

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} \left(1 - \frac{\Delta t}{RC}\right) & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} U_{RC,k} \\ SOC_k \end{bmatrix} + \begin{bmatrix} \frac{\Delta t}{C} \\ \frac{\eta_{\mathrm{C}} \times \Delta t}{C_n} \end{bmatrix} \times I_k + \begin{bmatrix} \boldsymbol{w}_{1,k} \\ \boldsymbol{w}_{2,k} \end{bmatrix} \tag{6}$$

and the system response by the measurement noise $\boldsymbol{v}_k$

$$y_k = U_{\mathrm{cell},k} = U_{\mathrm{OCV}}(SOC_k) + U_{RC,k} + I_k \times R_\Omega + v_k \tag{7}$$

Process noise is a stochastic quantity and models nonmeasurable external conditions that affect the states. Measurement noise accounts for uncertainties in the measurement system. Both variables are considered as uncorrelated white Gaussian noise with known covariance matrices. Generally, both noise terms do not exhibit these properties in reality, although most authors agree that this method is nevertheless considered practical.[4] The covariance is a measure of the probability that the calculated value corresponds to the actual value in the physical system. The higher the values of the covariance, the more uncertain is the estimate of the associated states. The covariance matrix $\boldsymbol{P}$ of the state vector is calculated according to

$$\tilde{\boldsymbol{P}}_k = \mathbb{E}[(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k)(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k)^{\mathrm{T}}] \tag{8}$$

where $\mathbb{E}[\cdot]$ defines the statistical operator for the expected value.[4]

For the recursive determination of the states, the KF resorts to the principle of Gaussian sequential probabilistic deduction, which consists of six steps. Thus, at the end of each iteration, the probable value $\tilde{\boldsymbol{x}}_k$ for the state can be calculated together with the corresponding covariance matrix $\tilde{\boldsymbol{P}}_k$. The updated estimates at the end of the previous iteration are the starting point for the estimate at the new measurement point. The feedback creates a loop in the algorithm that continuously corrects the calculation.[25]

The loop and the six steps are shown in **Figure 3**. Before the calculation starts, the filter must first be initialized. The initial values $\tilde{\boldsymbol{x}}_0$ and $\tilde{\boldsymbol{P}}_0$ should be chosen as well as possible based on the available information.[4,14]

First, the update for $\hat{\boldsymbol{x}}_k$ and $\hat{\boldsymbol{P}}_k$ is performed. Then, the algorithm calculates the system response and the gain matrix $\boldsymbol{L}_k$ to correct the quantities in the time update. The equations for the KF associated with each step can be obtained from Table 1. The matrices $\boldsymbol{A}_k$, $\boldsymbol{B}_k$, and $\boldsymbol{C}_k$ can be time-varying and are taken from the state space model. The matrix $\boldsymbol{Q}$ and the scalar r are the covariances of $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$. These two covariances for the noise terms cannot be computed, but must be fixed a priori.[4]

The filter has already been proven in a wide variety of forms for battery state estimation in numerous studies.[3,12,36–39] The

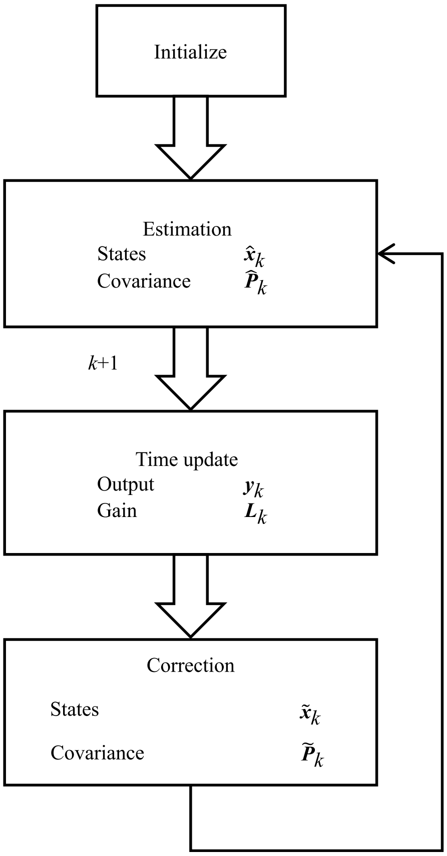

Figure 3. Procedure of the recursive state estimator. After initialization, the states and the covariance are calculated with current measured values. Thereupon, the error adjustment takes place and the values are corrected accordingly.

variant used in this work is the EKF. It is often applied to nonlinear systems (compare refs. [4,6,40–43]) and linearizes the model at each measurement point by forming the derivative.[44]

This is mostly done by a first-order Taylor series approximation.[14] Applying a linear KF to this system results in an EKF for an originally nonlinear system.[4]

For an application of the EKF, a nonlinear state space model is needed. The equation of the states then takes the form

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, u_k) \tag{9}$$

and the system response corresponds to

$$y_k = g(\boldsymbol{x}_k, u_k) \tag{10}$$

Because of the nonlinearity of the equations, the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ for the EKF must be replaced by the partial derivatives of the functions f in Equation (4) and g in Equation (5) as well as in Table 1. Otherwise, the procedure of an EKF equals that of an ordinary KF.[6]

The recursive algorithm of the KF makes it possible to avoid the time-consuming storage of multiple measurement data, as is the case with a moving average filter.

2.4. Adaptive Extended Kalman Filter

To solve the problem of inaccurate estimations due to insufficient covariance values, an AEKF can be used. Here, the entries in the covariance matrices are adjusted during the operation of the EKF with the aim of limiting the estimation error. Several methods exist to achieve this goal. Mehra[45] classified them into four categories: Bayesian estimation, maximum likelihood methods, correlation methods, and covariance matching methods. In this work, two variants of the covariance matching methods are used. The goal of these methods is to keep the deviations between the measured and the predicted values of the output variable consistent with the theoretical covariances of the EKF[45] In the following, the moving window average (MWA) and FF approaches used in this work are explained in more detail. In the approach with an MWA, the mean value

$$R_k^\sigma = \frac{1}{\sigma} \sum_{m=k-\sigma+1}^{k} \left(y_m - y_{m,\text{measured}}\right)\left(y_m - y_{m,\text{measured}}\right)^\mathrm{T} \tag{11}$$

is determined for the sum over the deviations between calculated and measured output variables during the last σ iterations of the KF (see Equation (11)).[6] Using this average, the covariance matrices of the system noise

$$\boldsymbol{Q}_{k+1} = \boldsymbol{L}_k R_k^\sigma \boldsymbol{L}_k^\mathrm{T} \tag{12}$$

and measurement noise

$$r_{k+1} = R_k^\sigma + \boldsymbol{C}_k \boldsymbol{P}_{k+1} \boldsymbol{C}_k^\mathrm{T} \tag{13}$$

are then fitted.[6]

A second version of the AEKF uses an FF instead of an MWA to adjust the covariances during operation of the KF. The FF can be of any value between 0 and 1. Smaller values lead to a higher reliance on the current deviations between the estimation of the output and the measured value, higher values lead to a focus on the history of these deviations. For FF → 1, the AEKF turns into a regular EKF. The adjustment of the covariances

$$\begin{aligned}\boldsymbol{Q}_{k+1} = {} & (1-d_k)\boldsymbol{Q}_k + d_k \\ & \left[\boldsymbol{L}_k\left(y_k - y_{k,\text{measured}}\right)\left(y_k - y_{k,\text{measured}}\right)^\mathrm{T}\boldsymbol{L}_k + \hat{\boldsymbol{P}}_{k+1} - \boldsymbol{A}_k\tilde{\boldsymbol{P}}_k\boldsymbol{A}_k^\mathrm{T}\right]\end{aligned} \tag{14}$$

and

$$\begin{aligned}r_{k+1} = {} & (1-d_k)r_k \\ & + d_k\left[\left(y_k - y_{k,\text{measured}}\right)\left(y_k - y_{k,\text{measured}}\right)^\mathrm{T} - \boldsymbol{C}_k\hat{\boldsymbol{P}}_{k+1}\boldsymbol{C}_k^\mathrm{T}\right]\end{aligned} \tag{15}$$

is done with the FF according to ref. [46]. The parameter d_k is defined as $\frac{1-\mathrm{FF}}{1-\mathrm{FF}^k}$.[46] In this work, the AEKF is used as a benchmark to have a comparison value for the results of the EKF with the tuned parameters.

3. Experimental Section

3.1. Key Performance Indices

To compare and to evaluate the performance of different filtering approaches at different operating states, metrics are required. For this purpose, four key performance indicators (KPIs) are established and used in this paper. **Figure 4** shows an overview of the four KPIs illustrated by an exemplary course of the error for the estimation of the *SOC* during an SLC A load profile. One of the most commonly used metrics is the *RMSE*

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(y_k - y_{k,\text{measured}}\right)^2}. \tag{16}$$

Here, the squared deviation between a calculated value y at step k and the corresponding measured value at the same time step y_{measured} is accumulated for a certain time period and normalized. Since the *RMSE* averages over the complete course of the error, large deviations can occur for a short time without being significantly reflected in the KPI. Therefore, the maximum absolute deviation

$$\max(|\Delta SOC|) = \max\left(|y_k - y_{k,\text{measured}}|\right) \tag{17}$$

is additionally introduced as a KPI.

Furthermore, the drift of the estimated value compared to the measured value is considered. The estimation drift may be negligible for short load profiles but can lead to larger deviations of the estimation from the reference due to accumulating errors. The gradient a of the regression line of the deviation between estimated and measured value is used as part of a KPI

$$\Delta SOC_{\text{drift}} = y_k - y_{k,\text{measured}} \approx a \times t + c \tag{18}$$

for the drift behavior. The unit of the gradient a is $\frac{\%}{h}$.[47]

To evaluate the transient response to an incorrect initial value, the estimation error after 10% of the total duration of the respective load profile

$$\Delta SOC_{\text{trans}} = \varepsilon_{10\%} = y_{10\%} - y_{10\%,\text{measured}} \tag{19}$$

is used. The transient behavior of the state estimation algorithms may change for different load dynamics.[47]

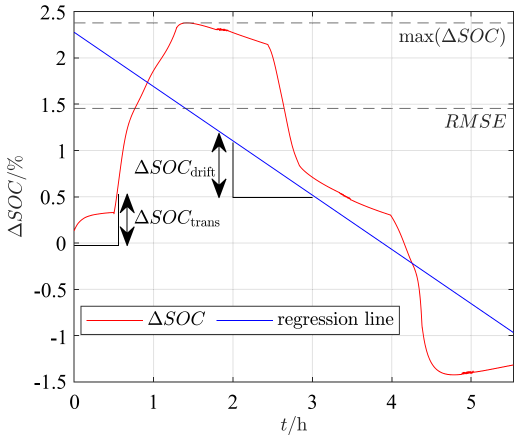

Figure 4. Overview of the definitions of the key performance indicators (KPIs) illustrated by an exemplary profile of the estimation error.

3.2. Cells and Load Profiles

The optimization is carried out for a Samsung INR 18 650-25 R lithium–nickel–cobalt-aluminum-oxide (NCA) cell. For this purpose, pulse measurements are executed at different *SOC*s to parameterize the ECM. The U_{OCV} curves are generated with pseudo OCV measurements with $\frac{\text{C}}{20}$. The parameters and voltage curves are given in Appendix (see Figure A1). To show the capabilities of the observers the mean model parameters are used (see **Table 2**) and only the U_{OCV} is implemented as an LUT and, therefore, considered *SOC* dependent in this model. Afterward, HSE is trained with the measured voltage and *SOC* curves for the two generic standard load cycles (SLC) profiles (see **Figure 5**) introduced by Campestrini.[47] To validate the approach, EKFs tuned with the resulting filter parameters then estimate the *SOC* of the cells for measurements based on the Urban Dynamometer Driving Schedule (UDDS) by the United States Environmental Protection Agency. All measurements were performed with a BaSyTec system. The simulations of the models and state estimation algorithms run with a sample time Δt of 0.4. This value results from a compromise between simulation time of the HSE and accuracy of the estimations. Additionally, AEKFs tuned by hand with the trial-and-error method are used for estimating the *SOC* of the mentioned load profiles. These estimations serve as benchmark for the EKFs tuned with the HSE approach. The tuning by hand is carried out by the authors and their subjective experience in filter tuning and reflects the estimations with the smallest deviations that could be reached after extensive testing.

3.3. Robustness Analysis

Errors in the current or voltage measurement, incorrect parameters of the elements in the ECM, or a bad initial estimation of the state can lead to unstable behavior or inaccurate estimations of the KF and its variants.[36]

To investigate the resilience of the tuned filters against those errors, several failure cases are defined. In the first case, a random noise signal with an amplitude of 0.1% of the 1C current is added to the regular current signal. The second case consists of a random variation of the parameters in the ECM by up to 10%. This indicates how the filter behaves in case of ECMs that are not well parameterized. For the third case, the KFs are initialized with an incorrect value for the *SOC*. Therefore, the value SOC_0 is set 10% below the correct value to gain information about the convergence behavior.

Table 2. Used model parameters for simulation.

Chemistry	R_{Ω}	R	C	η_{C}
NCA	26.5	6.5	3.75	99.5%

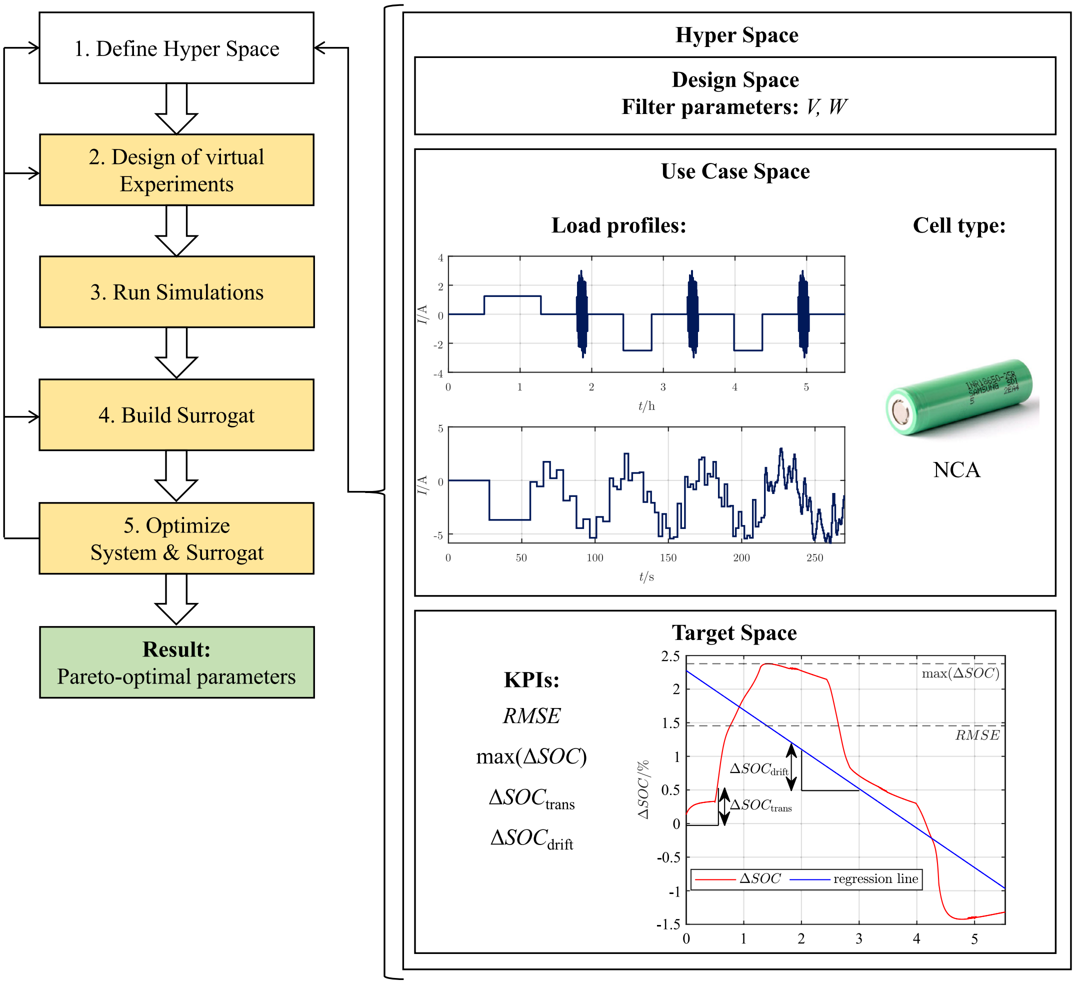

Figure 5. Overview of the hyperspace exploration (HSE) principle based on Palm,[21] as well as the subspaces of the hyperspace.

4. MOO Approach

4.1. Hyperspace Exploration

The HSE represents an approach to system design for complex problems. It combines methods for virtual prototyping with methods for the design of virtual experiments (DovE). The generic HSE workflow includes five steps. The results of these steps are several sets of pareto-optimal design parameters for the considered system. Figure 5 provides an overview of the steps in the workflow of the HSE as well as an overview of the different subspaces of the hyperspace.[21]

In the first step, the hyperspace is spanned as the Cartesian product of the design space $\boldsymbol{D}$, the use case space $\boldsymbol{U}$, and the target space $\boldsymbol{T}$. The design space contains all possible design variants for the system that are considered for optimization. The use case space includes the possible use cases, in which the system to be optimized can operate, while the target space consists of the desired target variables that are part of the objective functions of the system. In steps 2–5 of the HSE, the surrogate optimization of computationally expensive multiobjective problems (SOCEMO) algorithm (see the yellow boxes in Figure 5) is employed. The initial DovE, the second step of the HSE, is conducted using the Latin hypercube sampling in the SOCEMO algorithm, here with a symmetric Latin hypercube.[48,49] More details about the SOCEMO algorithm are provided by Müller in ref. [48]. After obtaining the experiment plan from the DovE, computationally expensive evaluations of the model are performed during the third step for each element of the experiment plan. The number of evaluations is defined previously during the DovE step. From the results of the evaluations a surrogate model of the system is constructed. The surrogate

$$\hat{s} = f_\alpha(d, u) \approx t(d, u) \tag{20}$$

is a function family that approximates the relation between the elements d and u of the design and use case space to the elements t of the target space.[21]

The SOCEMO algorithm uses radial basis functions (RBFs) to build the surrogate. For each objective inside the target space, a separate RBF with the general form

$$f_\alpha = \sum_{k=1}^{n} \alpha_i \phi(\|d - d_i\|_2) + P(d) \tag{21}$$

is calculated. In Equation (21), α denotes the scalar coefficients linking the vectors from the different spaces, ϕ denotes the respective real-valued function, in this case, the cubic function, and p is the linear polynomial tail.[48] During the optimization step, the SOCEMO algorithm employs different search strategies aiming at finding well-distributed, pareto-optimal sets of parameters from the design space. For more information about HSE, refer to Palm and Holzman.[21] In their work, they also present two examples for HSE applications in the automotive sector. The first example examines whether electric vehicles (EV) benefit from shiftable gear boxes and compares different topologies for the gear box via HSE. The second example evaluates the benefit of 2-wheel versus 4-wheel drive trains for EVs using the HSE.

4.2. Adaption for KF Tuning

To implement the process of tuning the KF parameters into the HSE, the design space encompasses the covariances of the noise terms $\boldsymbol{Q}$ and r as well as the initial values of the covariance matrix $\boldsymbol{P}$. Since the covariance matrices are diagonal and the model includes two observed states and one output variable, the design space consists of a total of five design parameters. The upper boundary for the elements of the design space is chosen to be 10 and the lower boundary is chosen to be 10^{-12} for each of the parameters. The use case space consists of the different applications of the system. In this work, the considered use cases are two different load profiles with different dynamic characteristics as well as a Li-ion cell with NCA chemistry. The load profile SLC A is used as a profile with low dynamics and the load profile SLC B as a profile with high dynamics.[47] The NCA cell mentioned in Section 3.2 is used for the experiments. Additionally, different cases like aged cells or temperature variations can be included in this step, but are beyond the scope of this work.

With the method of Latin hypercube sampling, the DovE is created, where the experiments containing the variations of the load profiles and different values for the covariance entries are planned. Then, the simulations of the EKF for these different scenarios according to the DovE are executed. From the results of the evaluations, a surrogate model of the system is constructed. The surrogate model maps the functional influence of the different variables in the design space and use case space on the target space, which comprises the four KPIs established in Section 3.1. A physical interpretation of the surrogate model like in an ECM is no longer possible. The actual optimization of the covariance entries is conducted with the surrogate model, because it is computationally inexpensive and can be executed fast. The algorithm runs until a certain number of complete evaluations of the EKF has been reached and provides the pareto-optimal values for the filter parameters in regard to the KPIs.

4.3. Combined Method

To further optimize the filter behavior, an MOO via HSE is trained for an AEKF with 2000 iterations at the two SLC load profiles mentioned before. Thereafter, the same validation procedure as for the EKF follows with the newly calculated filter parameters and the AEKF algorithm. The aim is to combine the step of initial tuning via the HSE and the adaption of the parameters during operation by the AEKF. This should result in a better convergence behavior.

5. Results and Validation

The run time is largely dependent on the number of expensive evaluations of the KF as well as of the used sample times Δt and the duration of the load profiles. The actual number of evaluations performed is slightly higher than the specified maximum number of evaluations due to the positioning of the checkpoints within the code. Further increasing the maximum number of evaluations only shows a slight increase in the performance of the filter with the obtained parameters. In **Table 3**, the corresponding run times and the number of actual evaluations are mentioned for different predefined numbers of evaluations.

To be able to compare the estimation behavior of the filters using the parameters obtained via HSE, two state of the art AEKFs are tuned with the commonly used technique of trial-and-error. Therefore, the two variants with an MWA and an FF are tuned by hand for the SLC A profile and adjusted to also perform sufficiently well for the SLC B profile. This shows one of the major advantages of the HSE concept. By including several profiles in the use case space it is possible to optimize the parameters for different load profiles at once. In the conventional method the filter is typically optimized for one profile only. The resulting filter parameters can be seen in **Table 4**. For comparability reasons, the MWA and FF variants tuned by hand are run with similar parameters. The HSE results show far smaller values for the values of $\boldsymbol{P}$.

5.1. HSE Tuning for EKF

The resulting estimations of the tuned filters for the two training profiles and the UDDS validation profile are depicted in **Figure 6**.

Table 3. Number of predefined and actual evaluations and run time of the HSE.

Defined number of evaluations	Actual number of evaluations	Run time
250	328	2546
500	536	4171
1000	1832	14 625
2000	2979	22 912

Table 4. Filter parameters for tuning. MWA and FF by trial and error. HSE EKF and HSE AEKF via HSE.

Parameter	MWA	FF	HSE EKF	HSE AEKF
ν	9×10^{-9}	9×10^{-9}	8.7×10^{-5}	1×10^{-12}
w_{11}	2×10^{-10}	2×10^{-10}	1	4.8×10^{-8}
w_{22}	2×10^{-7}	2×10^{-7}	1×10^{-12}	1
$\boldsymbol{P}_{11}$	1×10	1×10	9.9×10^{-1}	2×10^{-12}
$\boldsymbol{P}_{22}$	1×10	1×10	6.1×10^{-11}	1.7×10^{-12}
σ	1×10^{2}	–	–	–

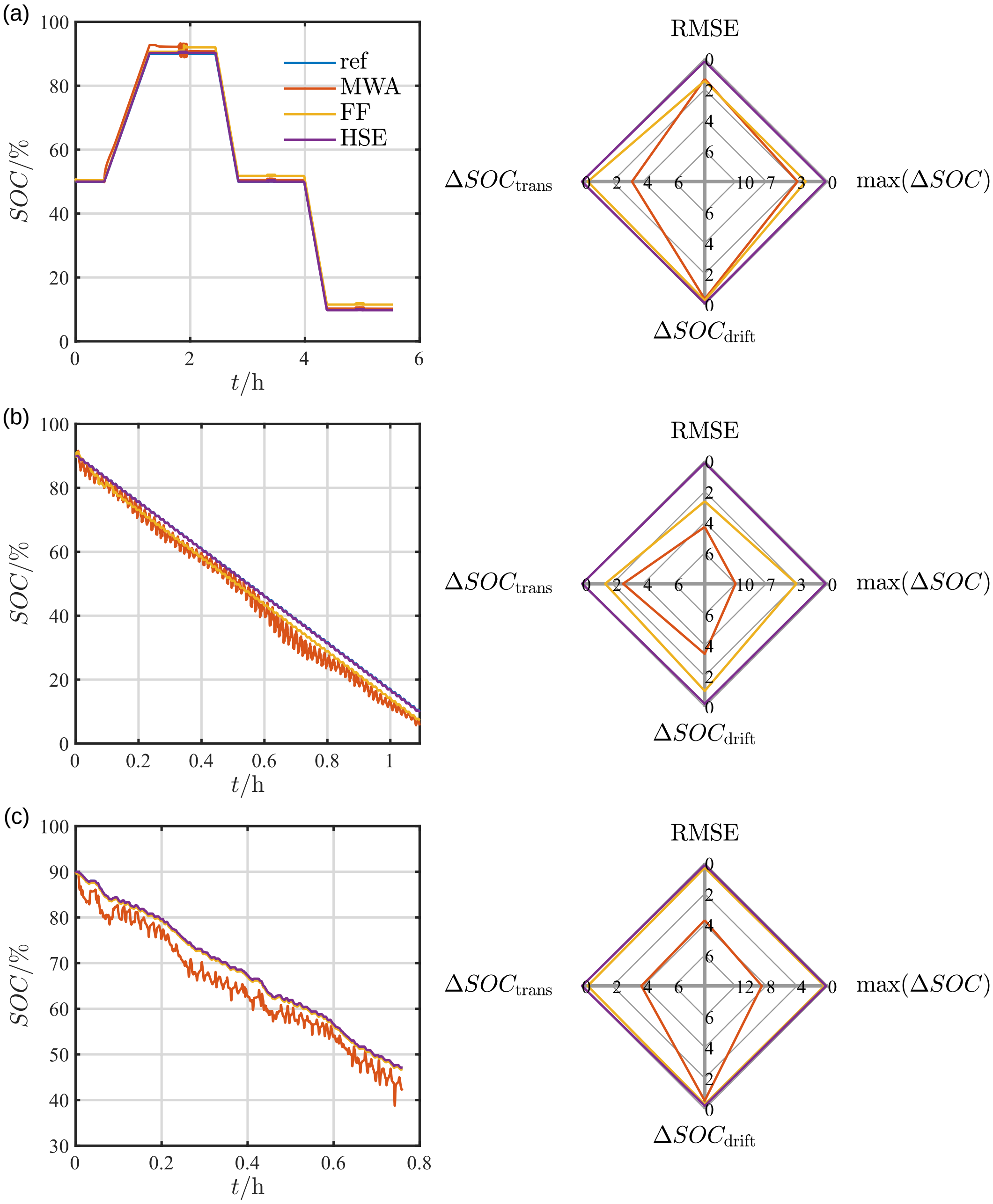

Figure 6. Estimation results for a) the SLC A, b) the SLC B, and c) the Urban Dynamometer Driving Schedule (UDDS) profiles. Compared are manually tuned moving window average (MWA) and forgetting factor (FF) adaptive extended Kalman filters (AEKFs) and an extended Kalman filter (EKF) tuned by the HSE.

Figure 6, 7 and 8 compare the manually tuned KF variants and the KF variants optimized by HSE with a reference case. The reference curves are measurements on the battery cell at the mentioned load profiles. The corresponding KPIs are represented in the spider diagrams next to the profiles. In case of the SLC A profile with less dynamics, the MWA AEKF overall performs with a slightly higher precision (*RMSE* of 1.3%) compared to the FF variant, which reaches an *RMSE* of 1.4%. However, the MWA AEKF shows higher deviations from the reference curve at the beginning, but manages to converge during the course of the profile. This results in a considerably larger max($|\Delta SOC|$) compared to the FF AEKF (3.3–2.4%). This trend consequently manifests itself in the KPI for the transient behavior $\Delta SOC_{\text{trans}}$, where the FF variant is far superior with a value of 0.4% compared to the 3.3% of the MWA AEKF. The drift KPI is rather low for both variants and suggests that this negative influence can be neglected in this case. The best performance in all categories for this profile shows the EKF tuned by the HSE. The *RMSE* is reduced to 0.2% and max($|\Delta SOC|$) shows a value of 0.3%. Therefore, a properly tuned EKF is able to outperform

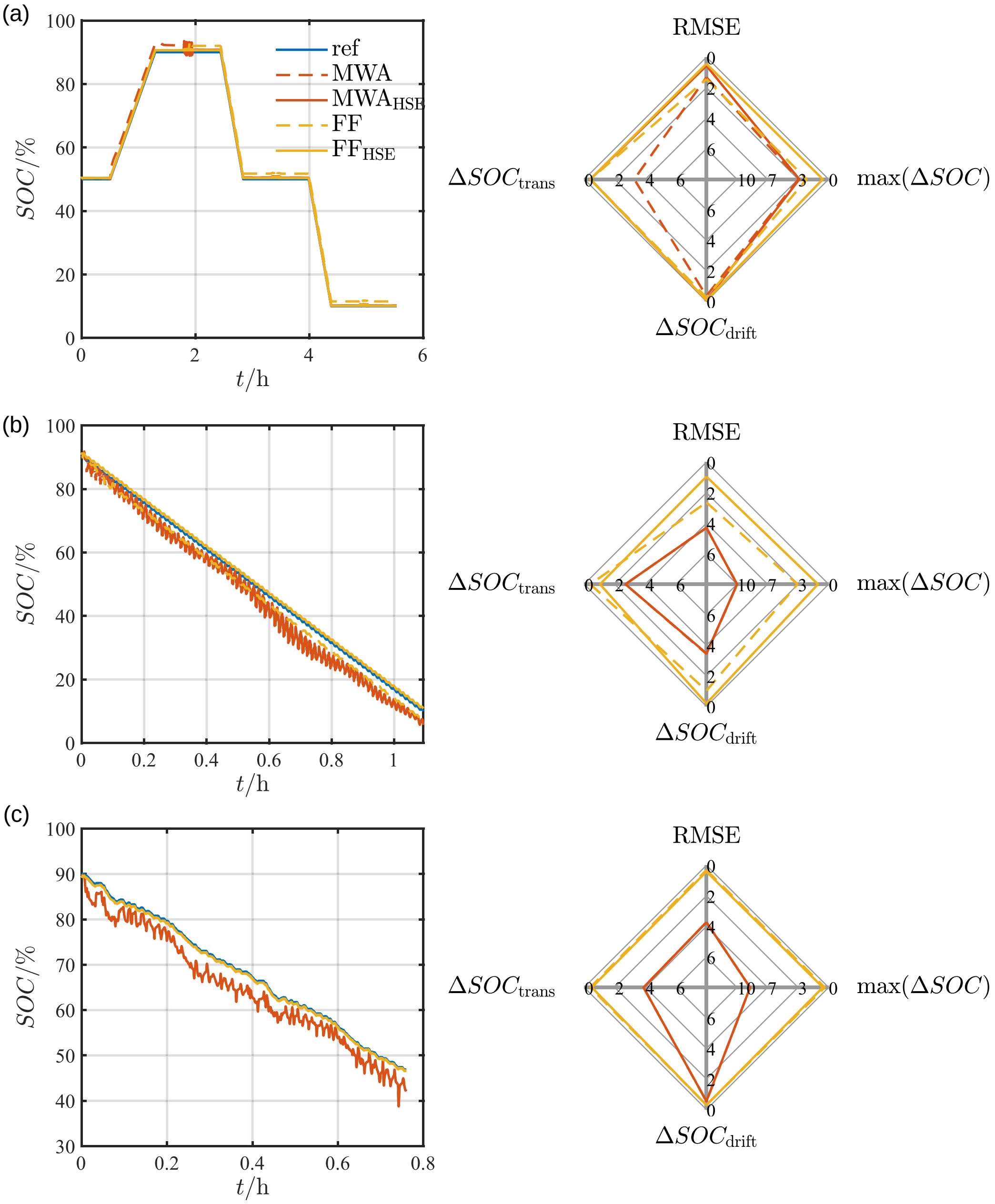

Figure 7. Estimation results for a) the SLC A, b) the SLC B, and c) the UDDS profiles. Compared are manually tuned MWA and FF AEKFs and an AEKF tuned by the HSE.

the state-of-the-art AEKFs that are manually tuned. The same trend is observable for the SLC B profile with higher dynamics. The FF AEKF yields lower values in all 4 KPIs compared to the MWA AEKF. Additionally, the MWA AEKF leads to high fluctuations in its estimation, which results in an *RMSE* of 4.3% and a maximum deviation of 9.9%. The drift for both variants is considerably higher due to the lack of static behavior in this profile. The worse performance for the SLC B profile can be explained by the filter tuning process. The AEKFs have first been tuned for estimations with very high precision for the SLC A profile and then the parameters were adjusted for the dynamic profile for acceptable estimations. In case of the HSE tuned EKF, no KPI surpasses a value of 0.2%. A comparison of the estimations for the UDDS profile, which has not been part of the training data set, shows the filter behavior for profiles that have not been part of the tuning process. The FF AEKF and HSE EKF derive estimations on a similar level regarding all KPIs, although the values for the later are lower in every aspect and do not surpass 0.3%. Again, the MWA AEKF performs the worst of the three variants, with higher fluctuations of the estimations, an *RMSE* of 3.7% and a maximum deviation of 8.6%. By considering both training sets at once in the MOO by the HSE, the filter

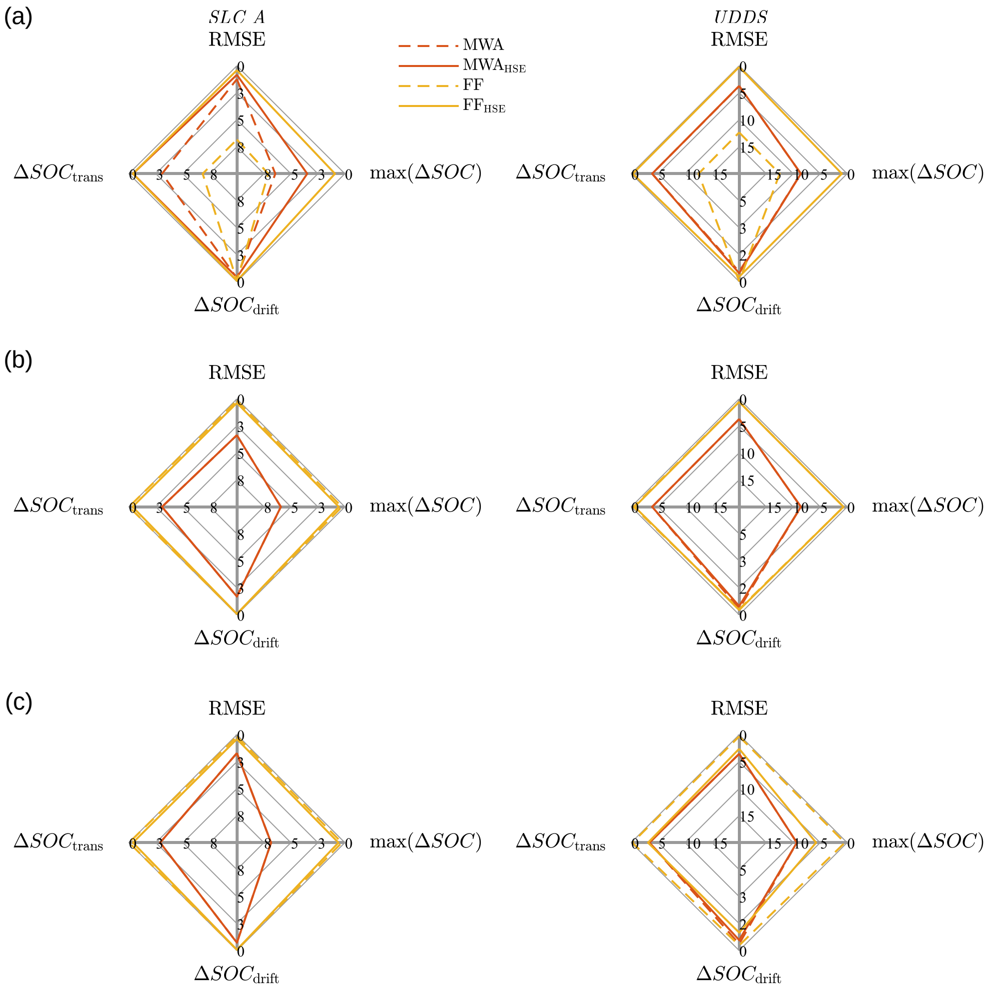

Figure 8. Estimation results for a) initialization with $SOC_0 - 10\%$ (state of charge [*SOC*]), b) additional noise on the input signal, and c) random variation of the ECM parameters by 10%. The SLC A profile is pictured on the left and the UDDS on the right.

parameters found are suitable to a wider range of applications compared to the traditional, manual tuning process, which mainly focuses on one profile.

5.2. HSE Tuning for AEKF

To illustrate the abilities of deriving the filter parameters via HSE, in the following, optimized versions of both AEKF variants are compared to their manually tuned counter parts (see Figure 7). The respective filter parameters are stated in Table 4. The MWA AEKF estimation behavior is only improved by the HSE in case of the SLC A profile, while no significant changes can be identified for the other two use cases. This indicates that suitable filter parameters have already been determined by the manual tuning process. However, for the first mentioned profile, the *RMSE* is reduced from 1.3% to 0.6% and also the transient behavior is improved significantly as is shown by a reduction of $\Delta SOC_{\mathrm{trans}}$ from 3.3% to 0.4%. In case of the FF AEKF, an improvement of the estimations for the two SLC profiles is observable, while maintaining the overall strong results for the UDDS profile. For the first load profile, the course of the estimation is approximated by the HSE closer toward the reference curve as can be deduced from the fact that the *RMSE* is reduced from 1.4% to 0.4% and the maximum deviation from 2.4% down to 0.6%. Similar behavior is illustrated in Figure 7b for the SLC profile with higher dynamics. Here, the *RMSE* decreases from 2.6% to 0.9% and $\max(|\Delta SOC|)$ from 3.4% to 1.2%. Thus, an improvement is achieved by automated filter tuning with the HSE framework compared to the time intensive tuning by the user.

5.3. Robustness Against Disturbances

In the following, the results of the robustness analysis from Section 3.3 are shown for an example with relatively low dynamics (SLC A) and the UDDS profile as an example with higher dynamics. Again, a comparison between the two manually tuned AEKF variants and the ones tuned by HSE is conducted. All mentioned fault cases have not been included in the tuning process. Therefore, this study shows the robustness of the found solutions against disturbances. Figure 8a depicts the estimations in case of incorrect initialization of the filters. Thus, the values of SOC_0 have been chosen 10% lower than the correct value. The influence of this disturbance on the estimation process can best be seen in the KPIs for the transient behavior $\Delta SOC_{\text{trans}}$ and for the maximum deviation $\max(|\Delta SOC|)$. In case for the manually tuned AEKFs, the difference between the reference SOC and the estimation after 10% of the total time $\Delta SOC_{\text{trans}}$ amounts to 3.3% for the MWA variant and 7.3% for the FF AEKF at the SLC A profile. The respective values for the HSE tuned filters do not exceed 0.4% which means a significant increase in convergence speed. Also $\max(|\Delta SOC|)$ is decreased from 6.9% to 3.7% (MWA AEKF) and from 7.6% to 0.9% (FF AEKF). These values give an impression on the adjustment of the initial deviation by the filters. Similar to the results in Figure 7, the MWA AEKF is hardly influenced by the HSE for dynamic profiles. However, with the HSE parameters and the UDDS profile $\Delta SOC_{\text{trans}}$ sinks from 12.5% to 0.3% for the FF AEKF and the maximum deviation can be reduced from 12.5% to 1%. Therefore, convergence speed is increased by usage of the automated filter tuning process.

Figure 8b shows the estimations with additional noise on the input signal I (see Figure 2). The FF AEKF is not affected by the additional noise and all values for the KPIs do not exceed values of 0.6% for both load profiles and both filter tuning variants. The MWA variant is also not decisively changed by this disturbance in case of the dynamic profile. However, the $RMSE$ and the KPI for the drift behavior are influenced at the SLC A profile and increase to values of 3.6% and 3.3%. Additionally, the maximum deviation reaches values of 6.3%. Tuning the MWA filter with the HSE has no influence on the behavior of the estimation with noisy system input.

The random variation of the ECM parameters by 10% in Figure 8c simulates how the filters react, when the ECM is not parameterized well or the cell ages over time and, therefore, the cell parameters have changed with time. For the profile with low dynamics redundantly, small differences between the manually tuned and HSE optimized filter parameters can be observed. The FF AEKF performs overall very well again for this profile and the KPIs do not exceed values of 0.7%. The MWA variant shows a maximum deviation of 7.3% and an $RMSE$ of 1.8%. The other KPIs are similar to the values for the case without disturbances (see Figure 7a). However, the manually tuned FF AEKF performs significantly better for the UDDS profile compared to the HSE tuned parameters. An explanation for this behavior is that the HSE with its 1000 iterations specializes on the ECM with the correct parameters and is therefore more sensitive to changes in that regard. The manually tuned variant allows more leeway in that regard.

6. Conclusion and Future Work

As stated in the introduction, KFs are commonly used to account for uncertain current measurements or inaccurate model parameters for state estimation, e.g., SOC, in battery diagnosis and in BMSs. The main drawback of this algorithm is the dependency on manually tuned meta parameters, the covariances of the filter. Therefore, the results are strongly influenced by the skill and experience of the respective developer. The filter can be tuned either for convergence speed or for estimation accuracy, and usually the developer has to find the compromise that works best for the specific application. In this article, the time extensive tuning process of KFs to estimate the SOC of LIBs is automated for the first time by applying MOO via the holistic framework of the hyperspace exploration. With this algorithm, it is no longer necessary to manually search decent values for the covariance matrices, which is time-consuming and not a reproducible way of determining these parameters. Furthermore, tuning by HSE ensures a way to compare different filter applications, because the estimation behavior is no longer dependent on the skill of the user to find suitable values for the covariances. This framework also allows implementing different objective functions in the optimization and, e.g., tunes the filter behavior simultaneously for precision and convergence speed. Furthermore, different load profiles can be considered during the tuning process and improve the estimation for varying applications. This is especially of interest for hybrid and battery EVs. The hyperspace exploration is a useful tool, as the framework allows to include different driving cycles in the use case space and differently aged cells or temperature variations in the design space. These scenarios are then considered in the overall optimization by the algorithm. Different KPIs are established in this work to evaluate and compare the performance of the filter variants. It could be shown that the filter parameters found by the automated process perform at least as well as the manually tuned filters and in case of an AKF with an FF or an extended KF, the estimation results could even be improved in terms of precision and convergence behavior for load profiles with high and low dynamics. However, if the parameters of the ECMs deviate strongly from the values of the real system, inaccuracies occur. One of the advantages of the HSE is that a variation of the model parameters can be integrated in the use case space. Therefore, this source of inaccuracies is then integrated in the optimization process. This will be part of the authors' future work regarding filter tuning by MOO.

Appendix

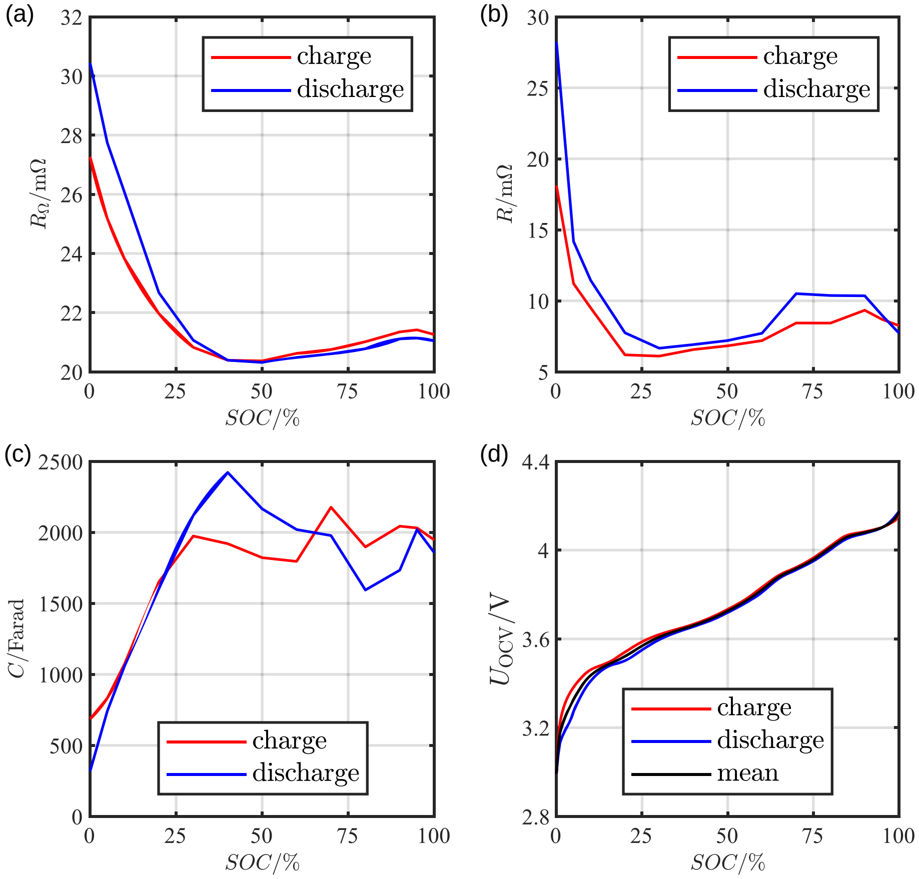

Figure A1. Model parameters for an NCA cell and the model in Figure 2. a) R_Ω, b) R, c) C, and d) the U_{OCV} curve. In this article, only the mean values of the parameters and of U_{OCV} are used.

Acknowledgements

The authors would like to thank Dr. Markus Mühlbauer for the input and for sharing his expertise on hyperspace exploration. Typos in the article have been corrected on 08 December 2023, after initial online publication of the Version of Record.

Open Access funding enabled and organized by Projekt DEAL.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords

Kalman filter, lithium-ion batteries, multi-objective optimization, state estimation

Received: July 13, 2023
Revised: August 21, 2023
Published online: October 2, 2023

[1] T. Kim, Y. Wang, H. Fang, Z. Sahinoglu, T. Wada, S. Hara, W. Qiao, *J. Power Sources* **2015**, *295*, 16.

[2] Z. A. Khan, P. Shrivastava, S. M. Amrr, S. Mekhilef, A. A. Algethami, M. Seyedmahmoudian, A. Stojcevski, *Sustainability* **2022**, *14*, 7412.

[3] Y. Wang, J. Tian, Z. Sun, L. Wang, R. Xu, M. Li, Z. Chen, *Renew. Sustain. Energy Rev.* **2020**, *131*, 110015.

[4] G. L. Plett, *J. Power Sources* **2004**, *134*, 252.

[5] J. Rivera-Barrera, N. Muñoz-Galeano, H. Sarmiento-Maldonado, *Electronics* **2017**, *6*, 102.

[6] C. Campestrini, T. Heil, S. Kosch, A. Jossen, *J. Energy Storage* **2016**, *8*, 142.

[7] S. Xie, X. Zhang, W. Bai, A. Guo, W. Li, R. Wang, *Energy Technol.* **2023**, *11*, 4.

[8] D. Gong, Y. Gao, Y. Kou, *Energy Technol.* **2021**, *9*, 2100235.

[9] P. Shrivastava, T. K. Soon, M. Y. I. B. Idris, S. Mekhilef, S. B. R. S. Adnan, *J. Energy Storage* **2022**, *56*, 106049.

[10] H. Zhang, Y. Bai, S. Yang, C. Li, *Energy Technol.* **2023**, *11*, 6.

[11] Y. Wang, Z. Chen, *Appl. Energy* **2020**, *260*, 114324.
[12] J. Xu, C. C. Mi, B. Cao, J. Cao, *J. Power Sources* **2013**, *233*, 277.
[13] M. Theiler, D. Schneider, C. Endisch, *Batteries* **2022**, *8*, 104.
[14] C. Campestrini, *Dissertation*, Technische Universität, München **2017**.
[15] P. Abbeel, M. Montemerlo, A. Y. Ng, S. Thrun, in *Proc. of Robotics: Science and Systems,* **2005**.
[16] T. O. Ting, K. L. Man, E. G. Lim, M. Leach, *Sci. World J.* **2014**, *2014*, 176052.
[17] D. Atukalp, K. Kink, in *2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, IEEE, Piscataway, NJ **2021**, pp. 01–06.
[18] Z. Wang, D. T. Gladwin, M. J. Smith, S. Haass, *Appl. Energy* **2021**, *294*, 117022.
[19] J. Zhang, S. Ci, H. Sharif, M. Alahmad, in *2010 Twenty-Fifth Annual IEEE Applied Power Electronics Conf. and Exposition (APEC)*, IEEE, Piscataway, NJ **2010**.
[20] P. Zarchan, H. Musoff, *Fundamentals Of Kalman Filtering: A Practical Approach, Volume 232 Of Progress In Astronautics And Aeronautics*, American Inst. of Aeronautics and Astronautics,, Reston, VA **2009**.
[21] H. Palm, J. Holzmann, in *2018 IEEE Int. Systems Engineering Symp. (ISSE)*, IEEE, Piscataway, NJ **2018**, pp. 1–6.
[22] W. Waag, C. Fleischer, D. U. Sauer, *J. Power Sources* **2014**, *258*, 321.
[23] S. Gantenbein, M. Weiss, E. Ivers-Tiffée, *J. Power Sources* **2018**, *379*, 317.
[24] C. Guenther, J. Klee Barillas, S. Stumpp, M. A. Danzer, in *Proc. of 3rd IEEE PES Int. Conf. and Exhibition on Innovative Smart Grid Technologies (ISGT Europe), 2012*, IEEE, Piscataway, NJ **2012**.
[25] G. L. Plett, *Battery Management Systems: Volume II: Equivalent-Circuit Methods*, Vol. *2*, Artech House, Norwood, MA **2016**.
[26] S. Piller, M. Perrin, A. Jossen, *J. Power Sources* **2001**, *96*, 113.
[27] X. Hu, S. Li, H. Peng, *J. Power Sources* **2012**, *198*, 359.
[28] J. P. Schmidt, Dissertation, Institut für Technologie, Karlsruhe **2013**.
[29] Y. Guo, Z. Yang, K. Liu, Y. Zhang, W. Feng, *Energy* **2021**, *219*, 119529.
[30] J. Tian, R. Xiong, W. Shen, J. Lu, *Appl. Energy* **2021**, *291*, 116812.
[31] P. Singh, A. Nallanchakravarthula, in *Proc. of the 13th Int. Conf. on, Intelligent Systems Application to Power Systems*, IEEE, Piscataway, NJ **2005**, pp. 261–267.
[32] C. F. Chiasserini, R. R. Rao, in *Pro. of the 5th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking – MobiCom '99* (Eds: H. Kodesh, V. Bahl, T. Imielinski, M. Steenstrup), ACM Press, New York USA **1999**, pp. 88–95.
[33] M. Mussi, L. Pellegrino, M. Restelli, F. Trovò, *J. Energy Storage* **2021**, *44*, 103309.
[34] G. L. Plett, in *Battery Management Systems: Volume I: Battery Modeling*, Vol. *1*, Artech House, Norwood, MA **2015**.
[35] H. He, R. Xiong, X. Zhang, F. Sun, J. Fan, *IEEE Trans. Veh. Technol.* **2011**, *60*, 1461.
[36] J. Klee Barillas, J. Li, C. Günther, M. A. Danzer, *Appl. Energy* **2015**, *155*, 455.
[37] G. Dong, J. Wei, Z. Chen, H. Sun, X. Yu, *J. Power Sources* **2017**, *364*, 316.
[38] Q. Yu, R. Xiong, C. Lin, *Energy Procedia* **2017**, *105*, 2791.
[39] W. Zhang, W. Shi, Z. Ma, *J. Power Sources* **2015**, *289*, 50.
[40] H. S. Ramadan, M. Becherif, F. Claude, *Int. J. Hydrogen Energy* **2017**, *42*, 29033.
[41] G. L. Plett, *J. Power Sources* **2004**, *134*, 277.
[42] G. L. Plett, *J. Power Sources* **2004**, *134*, 262.
[43] S. Sepasi, R. Ghorbani, B. Y. Liaw, *J. Power Sources* **2014**, *255*, 368.
[44] J. Li, J. Klee Barillas, C. Guenther, M. A. Danzer, *J. Power Sources* **2013**, *230*, 244.
[45] R. Mehra, *IEEE Trans. Autom. Control* **1972**, *17*, 693.
[46] E. Hou, Y. Xu, X. Qiao, G. Liu, Z. Wang, *Energies* **2021**, *14*, 5579.
[47] C. Campestrini, M. F. Horsche, I. Zilberman, T. Heil, T. Zimmermann, A. Jossen, *J. Energy Storage* **2016**, *7*, 38.
[48] J. Müller, *INFORMS J. Comput.* **2017**, *29*, 581.
[49] K. Q. Ye, W. Li, A. Sudjianto, *J. Stat. Plan. Inference* **2000**, *90*, 145.