A comparative study and validation of state estimation algorithms for Li-ion batteries in battery management systems

Joaquín Klee Barillas *, Jiahao Li, Clemens Günther, Michael A. Danzer

Zentrum für Sonnenenergie- und Wasserstoff-Forschung Baden-Württemberg (ZSW), Germany

HIGHLIGHTS

• Description of state observers for estimating the battery’s SOC.
• Implementation of four estimation algorithms in a BMS.
• Reliability and performance study of BMS regarding the estimation algorithms.
• Analysis of the robustness and code properties of the estimation approaches.
• Guide to evaluate estimation algorithms to improve the BMS performance.

ARTICLE INFO

Article history:
Received 27 February 2015
Received in revised form 29 May 2015
Accepted 29 May 2015
Available online 24 June 2015

Keywords:
Lithium-ion battery
Battery management system
State of charge estimation
Robustness analysis
Sliding-mode observer
Kalman-based SOC estimation

ABSTRACT

To increase lifetime, safety, and energy usage battery management systems (BMS) for Li-ion batteries have to be capable of estimating the state of charge (SOC) of the battery cells with a very low estimation error. The accurate SOC estimation and the real time reliability are critical issues for a BMS. In general an increasing complexity of the estimation methods leads to higher accuracy. On the other hand it also leads to a higher computational load and may exceed the BMS limitations or increase its costs.

An approach to evaluate and verify estimation algorithms is presented as a requisite prior the release of the battery system. The approach consists of an analysis concerning the SOC estimation accuracy, the code properties, complexity, the computation time, and the memory usage. Furthermore, a study for estimation methods is proposed for their evaluation and validation with respect to convergence behavior, parameter sensitivity, initialization error, and performance.

In this work, the introduced analysis is demonstrated with four of the most published model-based estimation algorithms including Luenberger observer, sliding-mode observer, Extended Kalman Filter and Sigma-point Kalman Filter.

The experiments under dynamic current conditions are used to verify the real time functionality of the BMS.

The results show that a simple estimation method like the sliding-mode observer can compete with the Kalman-based methods presenting less computational time and memory usage. Depending on the battery system’s application the estimation algorithm has to be selected to fulfill the specific requirements of the BMS.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Lithium-ion battery technologies are a very promising technology for both stationary energy storage and electro-mobility. The optimal operation to improve the performance, prolong the lifetime, and to prevent damage of the battery are key factors that have to be achieved by the battery management system (BMS).

Therefore, a precise real time reliable, accurate estimation of battery parameters and internal states are needed. Some of these states cannot be measured directly by sensors, which means that they have to be estimated making use of available measured variables such as cell voltages, current load, and cell temperatures. One of these internal states with a particular interest is the battery state of charge (SOC). The SOC represents the available battery capacity that can be withdrawn from the battery. On the other hand the SOC is used to prevent battery over-discharge or over-charge as well as to operate the battery in such a manner that aging effects are
reduced. One of the most implemented methods to estimate the SOC is the Coulomb counting method [1,2]. Although simple to carry out, an accurate estimation of the SOC cannot be achieved, mainly because of the initialization and accumulated measurement errors. The initialization error can be solved by having knowledge of the relation between SOC and the open circuit voltage (OCV) that can be obtained by characterization measurements [3–5] or by OCV estimation algorithms [6]. However, this requires long resting periods.

To solve these problems, many different model-based approaches have been developed and published in recent years, presenting high SOC estimation accuracy and reliability [7–9]. Although very precise, some of these methods compute complex equations that have to be processed by a microcontroller.

Besides the estimation of the battery state, the BMS has to fulfill other functions like protecting the battery from operating outside its safe operating area, collecting and reporting data, controlling its environment and cell balancing. These operations have to be executed within some few milliseconds to ensure the BMS functionality.

The focus of this work lies on the application of selected estimation algorithms and the study of their reliability regarding external disturbances and erroneous parametrization as expected in a real battery system environment. Furthermore this paper proposes a study of some fundamental topics besides the estimation accuracy of the state estimation algorithms that have to be considered for the implementation in a BMS, mainly the BMS cycle time, code complexity and code memory usage which are different for every algorithm.

To address the issue of comparability of observer design implementations the following points are taken into account:

- The programming is carried out by the same programmer.
- Same workflow and compiler is employed.
- Same battery module, BMS and battery model is used.
- Same parameterization procedure is followed.

Four estimation algorithms including Luenberger observer, sliding-mode observer, Extended Kalman Filter and Sigma-point Kalman Filter, are analyzed and compared in this work. At first the model-based estimation method theory as well as the approaches under study are presented. The results of validation tests regarding accuracy, robustness and computational time are demonstrated and discussed.

This work serves as a guide for analyzing, testing, and validating state estimation algorithms to assure the BMS reliability and performance for an optimal operation of lithium-ion batteries.

2. Cell model and model-based state estimation algorithms

2.1. Cell model

Model-based state estimation algorithms are very promising approaches for reliable battery monitoring [10,11]. These sorts of algorithms make use of an equivalent circuit, e.g. depicted in Fig. 1 which describes the battery’s dynamic, electrical behavior with a constant nonlinear voltage source for the OCV, a cell ohmic resistance and a RC-circuit connected in series.

For the presented equivalent circuit the current is defined positive for charging and negative for discharging. The relationship between the input values, current I, ambient temperature T, and the voltage output \( U_{\text{cell}} \) of the model leads to the estimation of internal cell states, e.g. SOC. Thinking of the system having a state vector that outlines the effect of past inputs on the system the cell model can be expressed in the following state space representation

\[
x_{k+1} = f(x_k, u_k, d_{uk}),
\]

\[
y_k = g(x_k, u_k, d_{uk}).
\]

where \( k \) is the discrete-time index, \( x_k \) the system state vector as a function of the system input \( u_k \) and the input disturbances \( d_{uk} \).

The system output \( y_k \) depends on the state vector, the system input and the output disturbances \( d_{yk} \). In this work the SOC and the voltage drop \( U_1 \) at the RC-circuit from Fig. 1 are the selected system states

\[
x_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} = \begin{bmatrix} \text{SOC}_k \\ \frac{U_{1,k}}{C_1} \end{bmatrix},
\]

and are defined as the discrete current integration

\[
\text{SOC}_{k+1} = \text{SOC}_k + \frac{\Delta t}{C_N} I_k,
\]

and as the voltage at the capacitance

\[
U_{1,k+1} = \left(1 - \frac{\Delta t}{R_1 C_1} \right) U_{1,k} + \frac{\Delta t}{C_1} I_k,
\]

where \( \Delta t \) is the time increment, and \( C_N \) the nominal cell capacity. In steady state (no current flow) the cell behavior is characterized by the OCV of the equivalent circuit. The dynamic response due to a change in current is described by the parameters \( R_0, R_1 \) and \( C_1 \). These values identified from characterization tests can be stored in look-up tables [3]. Hence the system output is defined as the cell voltage

\[
y_k = U_{\text{cell},k}.
\]

where \( U_{\text{cell},k} \) is calculated using the Kirchoff’s second law as

\[
U_{\text{cell},k} = \text{OCV} (\text{SOC}_k) + R_0 \cdot I_k + U_{1,k}.
\]

Then the battery’s dynamic and electrical behavior is given through the linear state Eqs. (4) and (5) and the nonlinear system’s output Eq. (7).

2.2. Model based state estimation principle

Since SOC is a non-measurable variable it has to be estimated by an algorithm that is capable to fully reconstruct the internal system states using a state observer. As seen in the structure of the SOC estimation in Fig. 2, the observer calculates the cell voltage \( \hat{U}_{\text{cell},k} \) and compares it with the measured voltage \( U_{\text{cell},k} \). The difference \( e_k \) between modeled and measured voltage is fed back through an observer gain into the model to correct the estimated system states and output. The variable \( \theta_k \) contains all the internal cell parameters used in the model.

Beside the input current \( I_k \) and temperature \( T_k \) the disturbances \( d_k \) affect the estimation of the internal states. In this approach the disturbances are seen as sensor uncertainties. Their influence to the estimated SOC is analyzed further in this work.
2.3. Proposed state estimation algorithms

(a) Luemberger observer

One of the most implemented estimation methods in linear control systems is the Luemberger observer (LO). The algorithm is simple to achieve and delivers good results in linear dynamic systems. Although the battery model is a nonlinear system, the LO can be applied because of the linearity of the state equations presented in (4) and (5). It uses a constant feedback gain so that the system state correction is proportional to the model error

\[ e_k = y_k - \hat{y}_k, \]

where \( y_k \) is the estimated output from the observer. The observed state-vector is then given by

\[ \hat{x}_{k+1} = f(\hat{x}_k, u_k) + L \cdot e_k, \]

with \( L \) as the feedback gain [12,13]. The observer gain \( L \) is chosen to make the continuous-time error dynamics converge to zero asymptotically and can be determined through pole placement. The poles can be chosen to reach a fast system response leading to greater noise susceptibility and estimation error. In this work the poles were selected to achieve low estimation error and system stability due to a high dynamic input signal which leads to a slower system response. This will be discussed in the results.

(b) Sliding-mode observer

Unlike the LO the sliding mode observer (SMO) uses nonlinear high-gain feedback to drive estimated states to a hypersurface where there is no difference between measured and estimated output. The nonlinear gain used in the SMO is typically implemented with the signum function \( \text{sgn}(\cdot) \) of the model error as a scaled switching function. Due to this high-gain switching feedback, the observer trajectories slide along a curve where the estimated output matches the measured output exactly. The equation for the state vector is given by

\[ \dot{x}_{k+1} = f(\hat{x}_k, u_k) + H \cdot \text{sgn}(h(e_k, \theta)), \]

where \( H \) is the feedback gain and \( \theta \) a vector of cell parameters [13,14]. The feedback gain \( H \) can be obtained with Lyapunov functions.

(c) Extended Kalman Filter

For nonlinear systems, as in the case of a battery system, the Kalman Filter can be extended through a linearization procedure resulting in the Extended-Kalman Filter (EKF). The EKF linearizes the state-space model every time step \( k \) considering the last estimated values. In the case of the EKF it is considered that the disturbances \( d_{uk} \) and \( d_{yk} \) are zero-mean white Gaussian stochastic processes with known covariance matrices \( \Sigma_{du} \) and \( \Sigma_{dy} \) respectively [15,16]. The following equations describe the algorithm’s procedure:

1. State estimate update

\[ \hat{x}_k = f(\hat{x}_{k-1}, u_{k-1}, d_{uk-1}), \]

\[ \Sigma_{x,k} = A_k \cdot \Sigma_{x,k-1} \cdot A_k^T + \Sigma_{du}. \]

2. Calculation of the Kalman gain

\[ L_k = \Sigma_{x,k} \cdot C_k \cdot \Sigma_{x,k} \cdot C_k^T + \Sigma_{dy} \]

3. Measurement update

\[ \hat{x}_k = \hat{x}_k + L_k \cdot [y_k - g(\hat{x}_k, u_k)], \]

\[ \Sigma_{x,k} = (I - L_k C_k) \Sigma_{x,k}. \]

The advantage of the Kalman filtering can be observed in (13), where the feedback-gain \( L_k \) is calculated in every time-step forcing the system to converge faster.

(d) Sigma-point Kalman Filter

The linearization approach in the EKF may lead to a loss in estimation accuracy and unstable filter algorithms while calculating the derivative in (16) and (17). Rather than differentiating the nonlinear system the Sigma-point Kalman Filter (SPKF) uses a set of points (sigma-points). Their mean and covariance matches the mean and covariance of the variable being modeled [12,17]. Like the EKF the SPKF can be described as a set of steps:

1. Determination of the sigma-points

\[ \chi_{k-1}^{0} = x_{k-1}, \]

\[ \chi_{k-1}^{j} = x_{k-1} + (\gamma \cdot \sqrt{n \cdot \Sigma_{x}}), \quad j = 1, 2, \ldots, n, \]

\[ \chi_{k-1}^{j} = x_{k-1} - (\gamma \cdot \sqrt{n \cdot \Sigma_{x}}), \quad j = n + 1, n + 2, \ldots, 2n, \]

where \( n \) is the dimension of \( \chi \), \( \gamma \) a weighting constant and \( \chi \) the vector of sigma-points.

2. State estimate update

\[ X_k = f(\chi_{k-1}, u_{k-1}, d_{uk-1}), \]

\[ \hat{x}_k = \sum_{j=0}^{2n} \chi_{k-1}^{j} \cdot \left( X_k^j - \hat{x}_k \right) \left( X_k^j - \hat{x}_k \right)^T, \]

\[ \Sigma_{x,k} = \sum_{j=0}^{2n} \chi_{k-1}^{j} \left( X_k^j - \hat{x}_k \right) \left( X_k^j - \hat{x}_k \right)^T. \]
where $X_k$ is the a priori estimation of the sigma-points vector, $x_m$ and $z$ are constant weighting factor vectors and $\Sigma$, the covariance error of the state estimation.  

3. Measurement update

$$Y_k = g(X_k, u_{k-1}, d_{u,k-1}),$$

$$\hat{y}_k = \sum_{j=0}^{2n} z_{ij} Y_k,$$

$$\hat{x}_k = \hat{x}_k^n + L_k (y_k - \hat{y}_k),$$

$$\Sigma_k = \Sigma_k^n - L_k \Sigma_k^n L_k^T,$$

where the Kalman gain $L$ is defined as

$$L_k = \Sigma_y^{-1} \Sigma_{y,k},$$

with the coefficients

$$\Sigma_y = \sum_{j=0}^{2n} z_{ij} (Y_k - \hat{y}_k) (Y_k - \hat{y}_k)^T,$$

and

$$\Sigma_{y,k} = \sum_{j=0}^{2n} z_{ij} (X_k - \hat{x}_k) (Y_k - \hat{y}_k)^T.$$  

As in the EKF the Kalman gain $L_k$ from (28) is adapted in every time-step reducing the estimation error effectively and fast.  

3. Simulation and code generation

To run the simulated algorithms on a BMS an executable machine code has to be created. Starting at the simulation framework the simulated algorithms have to be converted into a runnable C-code with code-generation and compiler tools. The generated C-code of the estimation algorithm is then inserted into the code of a real time operating system which communicates with the BMS components, e.g. voltage, temperature and current sensors. An internal interface sends the collected data from the operating system to the estimation algorithm that calculates the SOC. The combined code (operating system and estimation algorithm) is then compiled to create a runnable machine code applicable to the battery module designed for large transport machines such as trucks, vans and trains. This profile, depicted in Fig. 3(a), consists of long charge/discharge phases, due to the long-distance driving at a constant speed, with a maximum charge and discharge current of 1 C. The limitation of the current is applied to the battery module designed for large transport machines (e.g. trucks, vans and trains). This profile, depicted in Fig. 3(a), consists of long charge/discharge phases, due to the long-distance driving at a constant speed, with a maximum charge and discharge current of 1 C. The limitation of the current is applied to prevent battery damage and battery aging. All tests are executed at a controlled ambient temperature of 25 °C. The battery voltage response is plotted in Fig. 3(b).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RAM (kB)</th>
<th>Stack-size (kB)</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luenberger observer</td>
<td>2.16</td>
<td>0.05</td>
<td>5</td>
</tr>
<tr>
<td>Sliding-mode observer</td>
<td>2.30</td>
<td>0.05</td>
<td>13</td>
</tr>
<tr>
<td>Extended Kalman Filter</td>
<td>3.54</td>
<td>1.24</td>
<td>16</td>
</tr>
<tr>
<td>Sigma-point Kalman Filter</td>
<td>3.67</td>
<td>1.25</td>
<td>38</td>
</tr>
</tbody>
</table>

4. Experimental set-up

4.1. Measurement equipment

The battery system used for the validation tests consist of an NMC/graphite module with 12 cells connected in series with a capacity of 46 A h. The management of the system is controlled by a BMS of the company Sensor-Technik Wiedemann GmbH with a 32 bit main controller and a current range of ±300 A with an accuracy of ±0.2 A. The BMS is connected to a Digitron Power Electronics test bench (0–60 V, ±100 A) with a voltage accuracy of ±30 mV and a current accuracy of ±50 mA. The battery module and the BMS are placed in an ESPEC CORP PU3KP temperature chamber (−40 to +100 °C) during the validation tests. All measured values are logged using CANalyzer from the company Vector Informatik GmbH.

4.2. Validation test

To validate the presented algorithms a dynamic load profile is applied to the battery module designed for large transport machines such as trucks, vans and trains. This profile, depicted in Fig. 3(a), consists of long charge/discharge phases, due to the long-distance driving at a constant speed, with a maximum charge and discharge current of 1 C. The limitation of the current is applied to prevent battery damage and battery aging. All tests are executed at a controlled ambient temperature of 25 °C. The battery voltage response is plotted in Fig. 3(b).

5. Results and discussion

5.1. SOC and cell voltage validation

For the measurements discussed in this chapter the parameters for the estimation algorithms are determined at a defined operating point. Here a compromise between the system’s dynamic, stability and estimation accuracy is made. The parameters are
obtained, seeking for the fastest convergence behavior at steady state, wherein the SOC estimation error during current flow does not exceed 5% [21].

The resultant SOC for the complete current profile in Fig. 3(a) applied to the BMS with the estimation methods under investigation is shown in Fig. 4(a).

The presented SOC curves represent the mean SOC of the battery cells and are compared with the real battery SOC determined through the integration of the battery current. Before the test is started the battery has been charged to 60% SOC. All applied estimation methods, initialized correctly, present an estimation error lower than 4%. The SPKF presents the highest estimation accuracy with an average error over the entire curve of about 2%. As described above the precision of the state estimation algorithm depends on how good the observer models the cell voltages. To visualize the estimation accuracy of the equivalent model used in the observer, the measured voltage of one cell is plotted against its respective estimated voltages in Fig. 4(b). Very close agreement is observed in the prediction of the cell voltage by the estimation algorithms. The relative error of the modeled voltage lies under ±2% for all the estimation algorithms.

The SOC estimation error as well as the voltage estimation error can be reduced by expanding the equivalent circuit with more RC-circuits. On the other hand the algorithm makes use of more data which increases the RAM of the BMS. Its influence will be discussed later.

5.2. Dynamic convergence

The data generated by the estimation algorithm has to be saved in the internal memory of the BMS every cycle. This data is used to initialize the estimation algorithm and assures a proper functionality every time the application, e.g. electric vehicle, is switched on. In case the BMS is not initialized correctly, i.e. at data loss, the estimation algorithm has to be able to converge quickly to the real SOC value. To test this dynamic behavior the battery SOC is set to 60% and the BMS is initialized at SOC = 50%. The results are presented in Fig. 5.

While the LO needs more than 30 min to reach the real SOC, the other algorithms are able to estimate an accurate SOC value within 8 min. As described before, changing the poles of the LO to converge faster in steady state might lead to a loss in accuracy in the dynamic state. To this purpose the poles of the LO are adjusted to present comparable convergence behavior to the other algorithms. While the system can correct the initialization error within 8 min, shown in Fig. 6(a) and (c), the maximal SOC error during current flow rises to 9%, as seen in Fig. 6(b) and (d). Since a SOC accuracy of 3–5% is required [21], the feedback gain of the LO is chosen to fulfill this requirement achieving a convergence time as short as possible.

Fig. 3. (a) Dynamic current profile for the validation test and (b) the battery voltage response.

Fig. 4. Results of the validation tests for the studied estimation algorithms. The SOC estimations and the respective errors are shown in (a) and (c). The comparison of the modeled cell voltage with the measured voltage and the resulting error is plotted in (b) and (d).

Fig. 5. Dynamic convergence behavior of the SOCs for the validation tests.

Fig. 6. Dynamic response of the SOCs and the respective errors for the validation tests.
This shows that a simple estimation method like the SMO can compete with the Kalman-based algorithms’ fast convergence properties.

5.3. Disturbances

The state estimation algorithms do not only have to present high accuracy. It is important to know how the observer behaves when external disturbances or parameter uncertainties influence the system. In this work disturbances regarding current and voltage measurement and cell parameter uncertainty are studied. In the following, the results and the discussion regarding the robustness are presented.

(a) Voltage measurement noise

It has been already described in the earlier sections that the model based algorithms use the difference between modeled and measured cell voltage to estimate the cell SOC. In order to analyze the response of the state observer to a disturbance in the system output, i.e. the cell voltage, noise is added to the measured signal. The added noise has an amplitude of ±5 mV and is normally distributed.

The estimation errors of the state observers are presented in Fig. 7(a). It can be observed that the Kalman-based algorithms work very robustly under the presence of measurement noise with a maximum SOC estimation error of ~2%. Also the SMO shows good stability against the influence of external disturbances. For the LO, the error increases over time reaching a maximum of 3%.

(b) Current measurement noise

To investigate the response of the observers in case of a disturbed input, noise is added to the current signal. The introduced noise has an amplitude of ±100 mA which is four times the current sensor accuracy and is normally distributed. In Fig. 7(b) the results of the estimation algorithms to a disturbance in the current signal is compared. Again the SPKF achieves the best results, reaching a maximum deviation of 3.8%.

(c) Current sensor drift

Another disturbance type to be considered is the sensor drift, due to an uncalibrated or aged sensor. For this purpose an offset of −100 mA is added to the measured current.

From the point of view of the observer this means that at steady state a current is flowing through the battery. In case of a simple
Coulomb-counting method the current offset is integrated at every time step leading to an erroneous SOC estimate of the battery. The results of the modeled base algorithms under investigation are presented in Fig. 7(c).

The results demonstrate that the model based estimation algorithms are capable of reaching a good accuracy while having an erroneous input current signal. Although the EKF is closest to the real SOC at the end of the test, the estimation error varies strongly during the experiment reaching a maximum deviation of 5%. Instead, the SPKF error curve is smoother and reaches a maximum SOC deviation of only 3.8%.

(d) Parameter uncertainty

The presented methodology assumes that all cells in the battery have the same characteristics, i.e. the parameters of the cell model are used for all cells. In reality, the cells in the battery module can experience different temperatures and can age at different rates. This leads to a slight differentiation of the cell parameters from each other and a deviation in the estimation of the SOC. To evaluate the behavior of the algorithms in case of parameter uncertainty due to possible aging, a deviation of 30% was set to $R_0$ and $R_1$ for all cells. The results are plotted in Fig. 7(d).

This shows that the estimation algorithms are capable of determining the cells’ SOCs with a good accuracy although the parameters deviate.

6. Discussion

Depending on the application of the battery system and its components, the estimation algorithm has to be selected to fulfill the specified requirements and limitations. The presented analysis regarding estimation accuracy and robustness has to be balanced with the performance of the BMS regarding the code properties in Table 1. These code properties give an estimate on how the computational time of the BMS is going to be. The results regarding the computational time are presented in Table 2.

For the implementation in the application a suitable algorithm has to be selected. The decision has to be based on the results of the presented analysis. Fig. 8 presents the evaluation of the investigated issues scaled from 1 to 5, where 5 is the best approach and 1 the least. The values are normalized to the maximal and minimal reached results in the described analysis. Regarding the complexity, the higher the value in Fig. 8 the less complexity is presented by the algorithm. Through this representation a decision of the suitable algorithm can be made.

As seen in Fig. 8 the SMO presents the best balance of the code properties reaching higher accuracy and faster dynamic convergence than the LO. Regarding the computational time, RAM, code complexity and stack-size the SMO approach presents better results than the Kalman-based algorithms.

As observed, the more RAM an algorithm uses and the higher its complexity the more computational time the BMS needs. In this

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>BMS computational time (ms)</th>
<th>Max. SOC deviation (%)</th>
<th>Robustness analysis: max. SOC deviation (%)</th>
<th>Convergence time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luenberger observer</td>
<td>2.8</td>
<td>3.0</td>
<td>6.4</td>
<td>30</td>
</tr>
<tr>
<td>Sliding-mode observer</td>
<td>3.0</td>
<td>3.2</td>
<td>5.5</td>
<td>8</td>
</tr>
<tr>
<td>Extended Kalman-Filter</td>
<td>4.8</td>
<td>3.1</td>
<td>5.5</td>
<td>8</td>
</tr>
<tr>
<td>Sigma-point Kalman-Filter</td>
<td>9.1</td>
<td>2.1</td>
<td>3.8</td>
<td>3</td>
</tr>
</tbody>
</table>

Fig. 7. Results regarding the robustness due to (a) voltage measurement noise, (b) current measurement noise, (c) current sensor drift and (d) parameter uncertainty.

Table 2
Computational time and estimation results for the implemented estimation algorithms.
work a relative small battery is used to validate the estimation algorithms. A battery of an electric vehicle has considerably more cells connected in series and more SOCs to be estimated. The sequential estimation of all the cell’s SOC at the same time step leads to a higher computational time. The execution of the estimation algorithm as well as other functions has to be finished within the microcontroller’s cycle. For every function a time range is reserved for its execution. If this is not fulfilled a stack overflow and erratic BMS behavior may occur, showing peaks in the SOC curve because of data loss.

7. Conclusion

The analysis in this work shows that to implement an estimation algorithm more than the estimation accuracy has to be considered. Instead, the performance of the BMS regarding the algorithm code properties plays a significant role.

Although the SPKF presents the best accuracy, it may not be adequate for a battery with hundreds of cells. For this, a powerful BMS is needed to compute the cells’ states and manage the large amount of data.

Although a simple method, this study shows that the SMO approach can compete with the Kalman-Filter-based methods presenting less computational time and memory usage. Depending on the application of the battery system and its components the estimation algorithm has to be selected to fulfill the specific requirements.

References