Contents lists available at ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

Embedded real-time state observer implementation for lithium-ion cells using an electrochemical model and extended Kalman filter

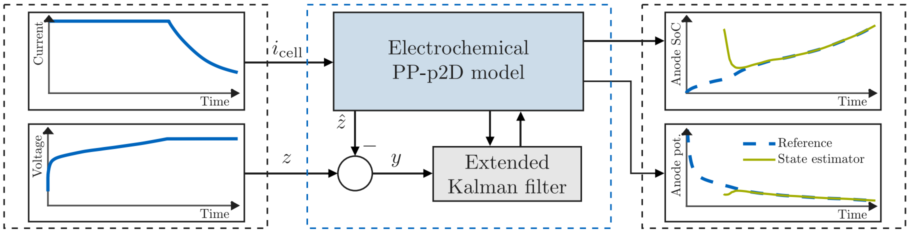
F.F. Oehler [a,b,*], K. Nürnberger [b], J. Sturm [a], A. Jossen [a]

[a] Technical University of Munich (TUM), Institute for Electrical Energy Storage Technology (EES), Arcisstr. 21, 80333 Munich, Germany
[b] Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany

HIGHLIGHTS

- Full microcontroller implementation of an electrochemical model-based state observer.
- Significant real-time capability on an application-oriented microcontroller.
- Robust estimation of global and model internal local states.
- Optimized numerical solution schemes yield performance gains and memory savings.
- Electrochemical models in battery management systems are applicable.

GRAPHICAL ABSTRACT



Cell monitoring using onboard physical sensors

Closed-loop observer running on BMS microcontroller AURIX TriCore

Estimation of internal local states

ARTICLE INFO

Keywords:
Pseudo-two-dimensional model
Extended Kalman filter
Battery management system
Lithium-ion battery
State estimation
Microcontroller

ABSTRACT

Accurate knowledge of the current state of lithium-ion battery cells is crucial for enhanced operational management of battery electric vehicles. Electrochemistry-based models provide the unique ability to predict cell internal states for advanced state monitoring, such as the anode potential, being of great interest for preventing lithium plating during fast charging. Furthermore, they are extendable by electrochemistry-based descriptions of cell degradation mechanisms. However, unlike the widely used equivalent circuit models, they are more computationally expensive and therefore have been scarcely used in embedded systems for online applications. This work addresses an efficient implementation of the popular pseudo-two-dimensional (p2D) electrochemical model in combination with a nonlinear filtering algorithm on a state-of-the-art microcontroller for automotive applications. For the first time, it is shown that with a suitable discretization scheme for the finite-difference approximation, a state observer convergence duration below 40 ms can be achieved for constant current, constant voltage charging at a current rate of 2C, with fast recovery of even local states. The mean relative error of the model internal states is less than 2% compared to an about 15 times slower implementation. This works therefore contributes valuably to demonstrating fast and robust performance of online state estimation using electrochemical models.

1. Introduction

Lithium-ion batteries represent the state-of-the-art technology for storing electrical energy in battery electric vehicles (BEVs) and are equipped with a battery management system (BMS). The latter provides key functionalities such as state monitoring, charge control, fault detection, and thermal management [1]. Battery cells are monitored by voltage measurement [2]. Current sensing is performed at pack level [3]. In addition, temperature sensors installed at specific positions contribute significantly to the accuracy of state estimation and the associated operating strategy [2].

Currently, various methods can be used to estimate the main state variables state of charge (SoC) [4] and state of health (SoH) [5] for

[*] Corresponding author at: Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany.
E-mail addresses: fabian.oehler@tum.de, fabian.oehler@infineon.com (F.F. Oehler).

https://doi.org/10.1016/j.jpowsour.2022.231018
Received 2 November 2021; Received in revised form 6 January 2022; Accepted 11 January 2022
Available online 10 February 2022

real-time application in BMSs. They can be grouped into three main categories: direct [6], data-driven [1] and model-based methods [7]. SoC calculation by coulomb counting [8] or by open-circuit voltage measurement [9] are examples of direct methods. Data-driven approaches [10,11], commonly presented as supervised machine learning methods, are receiving increasing attention, especially for SoH estimation [12,13]. Electrochemical models (EMs) and equivalent circuit models (ECMs) together form the third group known as model-based methods. ECMs require little processing power from microcontroller units (MCUs) and are widely used in BMSs due to their low parameterization complexity [14]. However, the lumped elements usually have limited physical meaning and their validity is limited to the range of their initial parameterization [15]. Moreover, conventional ECMs often cannot describe internal physical processes [16,17]. Extended models, called transmission line models, try to overcome this limitation [18–20].

In contrast, the Doyle–Fuller–Newman (DFN) model [21,22], a first-principle and pseudo-two-dimensional (p2D) model type, directly incorporates local concentrations and potentials of immediate physical significance. In theory, this enables a very accurate description of cell behavior but is associated with the drawback of a comparatively complex parameter identification process [16]. Nevertheless, previous research has shown that successful parameterization is possible using almost exclusively experimental techniques [23–27]. Moreover, the sensitivity of the model output is not the same for all parameters [28], which means the identification process can be further streamlined. Finally, the required parameters are predominantly design parameters that are important during cell development and therefore likely to be accessible to original equipment manufacturers in real-world use cases for BEVs.

The most commonly described limitation that currently prevents the use of p2D models in BMS is the high computational complexity [14,29,30], e.g., in comparison to ECMs, hampering real-time capability. Efforts have been made to mitigate this, either by model simplifications such as in the single particle model (SPM) [31–33] or by making some moderate assumptions, e.g., on the diffusion of lithium ions in the solid domain. This is often approximated by the parabolic profile (PP) method described by Subramanian et al. [34,35]. The validity of this method has been confirmed for high energy cells [36]. Other model reduction techniques, such as linearization of the Butler–Volmer equation [37,38] or Gelerkin's approximation [39,40] for the liquid phase lithium-ion concentration, have been reported.

Few numerical solution methods that improve the suitability of p2D models for real-time applications are available in the literature [41–43]. Unfortunately, performance analyzes were only performed using computers with powerful central processing units (CPUs).

Sturm et al. [44] implemented the p2D model with different discretization and approximation schemes on a microcontroller, focusing on a close comparison between MATLAB/COMSOL and the microcontroller implementations, and between different approximation schemes. Optimization of the C-code to improve execution time and reduce memory requirements is not included. A very simplified reduced-order model [45] was implemented by Verma et al. [46] on automotive microcontrollers. Unfortunately, little information on the implementation is presented, limiting the ability to evaluate and reproduce the results.

For use in BMSs, the p2D model or reduced versions thereof must be integrated into a state observer structure to adapt the model to an unknown state of the cell. Such closed-loop approaches are already widely used in conjunction with ECMs (e.g., [47–50]). Few studies have focused on the use of EMs integrated into state observers such as Kalman filters [51–54], e.g., using a state space representation of a full-order p2D model and an extended Kalman filter (EKF) [53]. EKFs are the extension of traditional linear Kalman filters for nonlinear systems [55]. Neither publication [53,54] includes an investigation of the real-time capability and model complexity because the implementations were done in MATLAB.

From this condensed literature review, it can be concluded that no study has as yet implemented a state observer in combination with an EM on an application-oriented microcontroller, and in-depth performance analyzes are not yet fully available. Therefore, this work aims (i) to implement and analyze a real-time state estimator using an EM subject to the constraints of a state-of-the-art automotive microcontroller and (ii) to apply methods for performance optimization in terms of computation time, memory requirements and observer stability. To the best of the authors' knowledge, such an investigation has not yet been reported in the literature.

It will be shown that an EKF-based state observer incorporating a full-order DFN model with PP approximation for solid diffusion no longer requires significantly more resources than with a reasonable ECM. Concurrently, superior real-time capabilities are achievable, while the EM offers the possibility of robust and accurate estimation of global and internal physical state variables of the lithium-ion cell. The latter are particularly relevant, e.g., for future anode potential controlled fast charging strategies. The results may contribute to a revision of the prevalent general assessment of the applicability of EMs in state estimators for BMSs within the community and thus motivate further research.

2. Methods

2.1. p2D model and approximations

In this study, a single full-order DFN model is used to describe an individual cell, as schematically shown in Fig. 1. Based on concentrated solution and porous electrode theory, it models the lithium-ion concentrations, electric potentials, and one-dimensional lithium-ion transport in the electrolyte within the two porous electrodes and the separator domain, shown in the lower part of Fig. 1. The intermediate space is filled with liquid electrolyte, likewise the separator region in the center. The electrode materials are typically graphite for the anode and, e.g., lithium nickel manganese cobalt oxide (NMC) for the cathode.

All calculations assume that the particles in the DFN model are spherical with uniform radius, which may differ for anode and cathode. The concentration profile of lithium in the particles can thus be simulated in the radial direction without a complex geometric description of the particles. This direction is referred to as the pseudo-dimension of the p2D model. The current collectors, copper at the anode and aluminum at the cathode, define the boundary conditions. Since the DFN model has no spatial resolution for the current collectors, the input is the current density i_{cell}, which is obtained by dividing the applied current by the active electrode area A_{act}. Inhomogeneous current densities leading to non-uniform utilization of individual cell areas are therefore not taken into account.

Discretization along the electrodes and separator is required to solve the model equations, which are not described in detail here due to the frequent use of the DFN model in scientific literature. This is performed based on a control volume formulation for the finite difference method [53], exemplified in the lower part of Fig. 1.

To solve the microscale diffusion partial differential equation in the spherical electrode particles, the PP approximation of Subramanian et al. [34] is applied. This reduces the problem to a set of three differential–algebraic equations (DAEs), but could cause deviations in the surface concentrations. This is the only difference between this work and the original description taken from Doyle et al. [21], who used superposition for lithium insertion via Duhamel's integral. However, this does not allow a recursive implementation [53].

The full state vector of the PP-p2D model for discrete node j and time step k can be written as

$$\mathbf{x}_{j,k}(x_j, t_k) = \begin{bmatrix} c_{\text{l}} & \Phi_{\text{l}} & c_{\text{ss}} & \Phi_{\text{s}} & i_{\text{l}} & j_{\text{n}} \end{bmatrix}^{\top}_{j,k}. \tag{1}$$

It consists of the lithium-ion concentration in the liquid phase c_{l}, the electrolyte potential Φ_{l}, the lithium-ion surface concentration of the

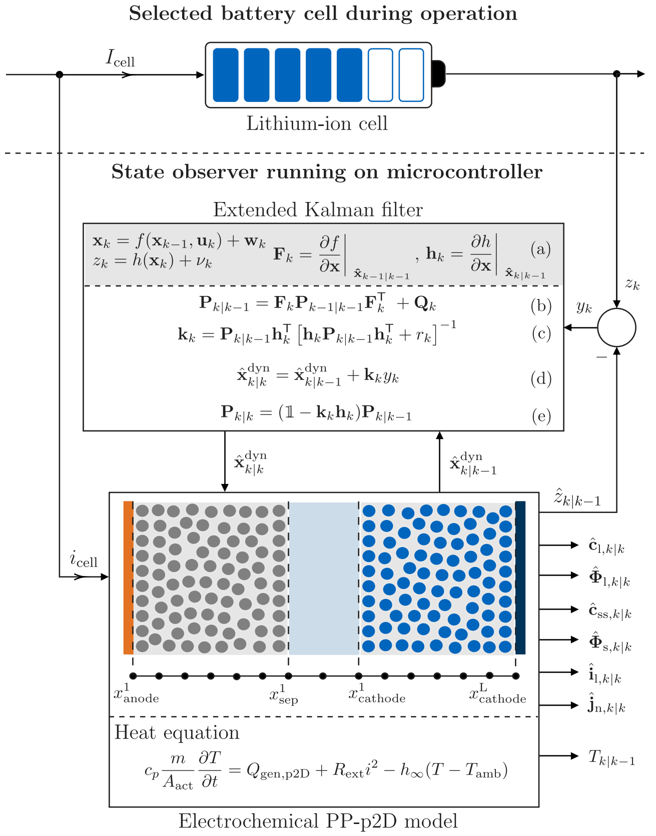

Selected battery cell during operation
I_cell
Lithium-ion cell
y_k
State observer running on microcontroller
Extended Kalman filter
$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k$
$z_k = h(\mathbf{x}_k) + \nu_k$
$\mathbf{F}_k = \frac{\partial f}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}}_{k-1|k-1}}$, $\mathbf{h}_k = \frac{\partial h}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}}_{k|k-1}}$ (a)
$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\text{T} + \mathbf{Q}_k$ (b)
$\mathbf{k}_k = \mathbf{P}_{k|k-1} \mathbf{h}_k^\text{T} \left[\mathbf{h}_k \mathbf{P}_{k|k-1} \mathbf{h}_k^\text{T} + r_k\right]^{-1}$ (c)
$\hat{\mathbf{x}}_{k|k}^\text{dyn} = \hat{\mathbf{x}}_{k|k-1}^\text{dyn} + \mathbf{k}_k y_k$ (d)
$\mathbf{P}_{k|k} = (\mathbb{1} - \mathbf{k}_k \mathbf{h}_k) \mathbf{P}_{k|k-1}$ (e)
z_k
y_k
$-$
$\hat{\mathbf{x}}_{k|k}^\text{dyn}$
$\hat{\mathbf{x}}_{k|k-1}^\text{dyn}$
$\hat{z}_{k|k-1}$
i_cell
$\hat{\mathbf{c}}_{1,k|k}$
$\hat{\mathbf{\Phi}}_{1,k|k}$
$\hat{\mathbf{c}}_{\text{ss},k|k}$
$\hat{\mathbf{\Phi}}_{\text{s},k|k}$
$\hat{\mathbf{i}}_{1,k|k}$
$\hat{\mathbf{j}}_{\text{n},k|k}$
x_anode^1
x_sep^1
x_cathode^1
$x_\text{cathode}^\text{L}$
Heat equation
$c_p \frac{m}{A_\text{act}} \frac{\partial T}{\partial t} = Q_\text{gen,p2D} + R_\text{ext} i^2 - h_\infty (T - T_\text{amb})$
$T_{k|k-1}$
Electrochemical PP-p2D model

Fig. 1. Concept using the PP-p2D model as an electrochemical model integrated into a state observer for estimation of the local internal states of a lithium-ion battery cell. The upper part schematically shows a real battery cell with an applied current as input and a voltage response as output. The lower part presents the EKF and the PP-p2D model.

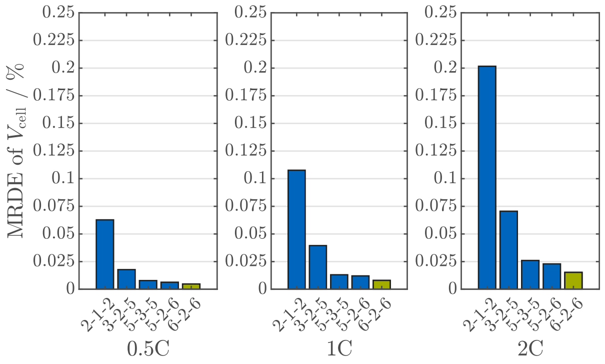


Fig. 2. Dependence of the MRDE of the cell voltage V_cell during CC/CV charging with 0.5C (left), 1C (middle) and 2C (right) on the number of discretization elements relative to a 10-5-10 discretization using MATLAB. This benchmark has been experimentally validated in previous work [44]. The abbreviation $n_\text{a} - n_\text{s} - n_\text{c}$ denotes the number of nodes in the anode n_a, separator n_s, and cathode n_c domain for different model discretizations.

active material c_ss, the solid potential Φ_s, the liquid current density i_l, and the molar flux j_n. The transition from the previous time step $k-1$ to k is performed via a Newton–Raphson approach:

$$\mathbf{x}_k^{(n)} = \mathbf{x}_k^{(n-1)} - \mathbf{J}_g^{-1}\left(\mathbf{x}_k^{(n-1)}\right) \cdot \mathbf{g}\left(\mathbf{x}_k^{(n-1)}\right) . \tag{2}$$

Eq. (2) is called recursively until the solution $\mathbf{x}_k^{(n)}$ converges, measured with absolute $\epsilon_{\text{abs},i} \le |x_{i,k}^{(n)} - x_{i,k}^{(n-1)}|$ and relative $\epsilon_{\text{rel},i} \le |(x_{i,k}^{(n)} - x_{i,k}^{(n-1)})|/|x_{i,k}^{(n)}|$ termination criteria for the entries x_i in $\mathbf{x}$. According to previous work [53], $\epsilon_{\text{abs},i} = 10^{-10}$ and $\epsilon_{\text{rel},i} = 10^{-4}$ are initially used for all state variables. Here, $\epsilon_{\text{abs},i}$ has the unit of the respective state variable, omitted for readability. If either of the two criteria is satisfied, the loop terminates after n iterations. The optimal choice of convergence criteria is a key aspects and is discussed in detail in Section 4. An important property of the Newton–Raphson method for minimizing the number of computationally expensive recalculations of the inverse Jacobian $\mathbf{J}_g^{-1}$ of the model equations $\mathbf{g}$ is the quadratic convergence rate given a suitable choice of initial values [56].

In addition to the state vector of the PP-p2D model, a heat equation is introduced to describe the temperature dependence of the charge transfer kinetics defined in the Butler–Volmer equation and the electrolyte potential. Further indirect temperature effects result from the open circuit potentials (OCPs) of the electrode materials and the electrolyte parameters. Hence, a time-dependent average temperature is calculated according to [27,57]. This neglects temperature inhomogeneities, which occur in particular at higher charging currents. In a previous work [27], the error caused by this was analyzed in detail with respect to a 3D electrochemical–thermal model. Since, as described in Section 2.1, only a single PP-p2D model is implemented in this work, the use of a 0D thermal model seems reasonable. The heat equation is shown at the bottom of Fig. 1 with specific heat capacity c_p and mass density mA_act^{-1}. Reaction heat q_r and ohmic heat caused by the electrolyte q_l and by the solid q_s potential gradient are considered as generation terms Q_gen for the p2D model [27,58]. Additional heat may be introduced by external resistances summarized in R_ext, e.g., at the contacts or current collector foils. Passive heat transfer to the ambient at constant temperature T_amb is approximated by the transfer coefficient h_∞, which simplistically adds together the contributions of heat convection, heat radiation and heat conduction [27,44].

The PP-p2D model is parameterized for a commercial cylindrical 3.35 Ah NMC-811/SiG cell (INR18650-MJ1, LG Chem, South Korea), which was extensively characterized in [27]. This high-energy cell has a nickel-rich cathode and a silicon–graphite composite anode (SiG). Cylindrical cells remain of great interest for future automotive applications as shown, e.g., by Tesla's proposed large format 4680 cylindrical cell [59].

2.2. Number of discretization elements

A preparatory evaluation was performed using the MATLAB code of the PP-p2D model from [53] to determine a suitable number of discretization nodes for implementation on an MCU. The abbreviation $n_\text{a} - n_\text{s} - n_\text{c}$ denotes the number of nodes in the anode n_a, separator n_s, and cathode n_c domain. The mean relative discretization error (MRDE) for different configurations is shown in Fig. 2 in reference to a 10-5-10 discretization in MATLAB. A qualitatively similar trend with decreasing relative error for finer discretization is observed for different charge rates. The error increases with increasing charge currents due to increasing inhomogeneities in electrode utilization. Incrementally increasing the discretization reduces the MRDE slightly more if it is done in either electrode, as shown by comparing the 5-3-5 with the 5-2-6 discretization. Choosing $n_\text{a} = n_\text{c} = 6$ and $n_\text{s} = 2$ yields an MRDE of about 0.015% for 2C. The number of DAEs to be solved in this configuration is only 60.4% compared to the 10-5-10 benchmark. Obviously, other combinations are conceivable, but 6-2-6 is found to be a good trade-off between complexity and fidelity and is chosen for the analyzes presented in this paper.

2.3. State observer framework

A discrete-time state-space representation of the PP-p2D model has been reported in previous work [53] and implemented and investigated in MATLAB with an EKF algorithm. The concept of using the PP-p2D model as part of a state observer framework is illustrated in Fig. 1. Schematically depicted at the top is a *selected* battery cell in operation, indicating that the cell may be part of a larger battery module or pack. The overall goal of the observer is to match the electrochemical model as closely as possible to the current state of the real cell during initial start-up and to accurately track state changes during operation. Both can be achieved by continuously correcting the state of the observer framework to avoid increasing deviations caused, e.g., by inaccuracies in current measurement and limited sampling frequency, as can occur in coulomb counting [8]. The local state variables of the PP-p2D model in $\hat{\mathbf{x}}_{j,k}$ are estimates for local internal states of the real cell. For a given current density i_{cell} at time step k, a solution of the PP-p2D model is computed based on the previous state. The cell voltage is given by the difference between the solid potentials at the current collectors

$$\hat{z}_{k|k-1} = \hat{\Phi}_{\text{s}}(x^{\text{L}+1}_{\text{cathode}}) - \hat{\Phi}_{\text{s}}(x^{0}_{\text{anode}}) - R_{\text{f}}\, i_{\text{cell}}, \tag{3}$$

where R_{f} represents additional ohmic resistance due to the current collector foils. The residual voltage $y_k = z_k - \hat{z}_{k|k-1}$ is derived from the current noisy voltage measurement z_k, and serves as the input to the EKF. In addition, a dynamic state vector

$$\hat{\mathbf{x}}^{\text{dyn}}_{j,k|k-1} = \begin{bmatrix} c_{\text{l}} & \bar{c}_{\text{s}} & \bar{q}_{\text{s}} \end{bmatrix}^{\mathsf{T}}_{j,k|k-1} \tag{4}$$

is passed to the EKF, consisting of c_{l}, volume-averaged solid concentration $\bar{c}_{\text{s}}$, and flux $\bar{q}_{\text{s}}$.

The update of $\bar{c}_{\text{s}}$ and $\bar{q}_{\text{s}}$ for time step k is derived from the PP approximation [53]. For the lithium-ion concentration in the liquid c_{l}, the mass balance equation discretized in space and time can be written in matrix form [53]. Overall, the full transition matrix $\mathbf{F}_k$ is computed for the three dynamic state variables (see (a) in Fig. 1). Linearizing the expression for the cell voltage [53] yields the observation matrix $\mathbf{H}_k$, which henceforth is written as vector $\mathbf{h}_k$ because z_k is the only available sensor information.

The EKF algorithm can be executed using the state-space representation for the current time step k. Referring to Fig. 1, this comprises the computation of the *a priori* state estimation error covariance matrix $\mathbf{P}_{k|k-1}$ (b) as well as the Kalman gain $\mathbf{k}_k$ (c). When calculating $\mathbf{k}_k$, the innovation covariance matrix in square brackets must be inverted, which can be avoided in this work because only one measurand z_k is used and the expression returns a single number. Thereafter, the Kalman gain is used to compute the updated *a posteriori* dynamic state vector $\hat{\mathbf{x}}^{\text{dyn}}_{k|k}$ (d) as well as the estimation error covariance $\mathbf{P}_{k|k}$ (e). The modified dynamic state vector is fed back into the PP-p2D model, and a new consistent solution $\hat{z}_{k|k}$ is computed along with an updated average cell temperature $T_{k|k-1}$.

The choice of the process $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ and the measurement noise $v_k \sim \mathcal{N}(0, r_k)$, described as Gaussian white noise processes, is crucial for the performance of the EKF. The precise specification of these stochastic processes is often difficult [60]. Some papers lack comprehensive information on specific numerical values (e.g., [54]). Nevertheless, for the readers' convenience and reproducibility of results, the parameters used in this study are briefly described. The measurement noise covariance r_k is a scalar and set to 10^{-4} V^2 [51,53]. The matrix $\mathbf{Q}_k$ contains the process noise information of each dynamic state variable and is assumed to be uncorrelated, hence all off-diagonal elements are zero. The entries for c_{l} and $\bar{q}_{\text{s}}$ are chosen analogously to those of previous work [53]. For $\bar{c}_s$, studies in this work have revealed that an adjustment depending on the charge/discharge current interval improves the EKF stability. In Eq. (5), the standard deviation $\sigma_{Q,\bar{c}_{\text{s}}}$ [53] is divided by an empirical factor of 15 for currents $I(t_k)$ greater than and equal 0.1C, while otherwise it is multiplied by 3.

$$\sigma_{Q,\bar{c}_{\text{s}}} = \begin{cases} \frac{1}{15} \cdot \frac{3\Delta t_k}{R_{\text{p,a}} F} \cdot I(t_k), & \text{for } I(t_k) \geq 0.1\text{C}, \\ 3 \cdot \frac{3\Delta t_k}{R_{\text{p,a}} F} \cdot I(t_k), & \text{else.} \end{cases} \tag{5}$$

Δt_k denotes the time step size, $R_{\text{p,a}}$ the particle radius of the anode material, and F is Faraday's constant. Small applied currents lead only to small changes in the model state, which can introduce inaccuracies when solving the equations numerically. By increasing the process noise variance for $\bar{c}_{\text{s}}$ for currents smaller than 0.1C, this additional uncertainty can be handled and the ability to correct potentially inaccurate values is improved [47]. Conversely, reducing $\sigma_{Q,\bar{c}_{\text{s}}}$ for larger currents prevents too large an estimation error [47].

3. Microcontroller implementation

3.1. Features of the microcontroller

In contrast to the MCU used in [44], this work uses the state-of-the-art automotive real-time integrated architecture (AURIX) microcontroller of the second generation TC387 [61] by Infineon Technologies AG, Germany. It has four cores with a 32-bit super-scalar TriCore architecture [61], of which only the first (CPU 0) is used for all analyzes performed in this work. The clock frequency was set to the maximum frequency $f_{\text{clk}} = 300$ MHz [61] and remained unchanged. A static random-access memory (SRAM) of 304 kB [61] is available per CPU. The flash memory is 10 MB [61]. Furthermore, there is an integrated floating-point unit that is optimized for operations on 32-bit floating-point numbers. According to IEEE Std 754-2008, this format is denoted as binary32 [62].

3.2. C-code for EKF comprising PP-p2D model

Code was manually developed in C for the p2D model based on the previous work [44] to implement the state observer. The PP-p2D model was supplemented by the EKF, using the MATLAB code [53] as a reference. For the implementation, a spline interpolation [44] is used for the lookup tables with equilibrium half-cell potentials of SiG anode $E_{\text{eq,a}}$ and NMC-811 cathode $E_{\text{eq,c}}$. The same applies to the temperature dependency of the OCPs, i.e., the entropic coefficients $\partial E_{\text{eq,i}}/\partial T$ with $\text{i} = \{\text{a}, \text{c}\}$. The program flow is shown in Fig. 3. After one-time initialization in `InitStateObserver` assuming equilibrium conditions, the state vector $\hat{\mathbf{x}}_{k|k-1}$ is calculated in `RunModel` along with the cell voltage estimate $\hat{z}_{k|k-1}$ in `TerminalVoltage`. In addition, the volume-averaged solid concentration and flux are calculated using PP approximation in `SolidDiffApprox_PP`. For further details, the reader is referred to the original work [34].

After defining $\mathbf{Q}_k$ according to Eq. (5), the linearized state-space representation in `StateSpace_p2D` is calculated for the current time step. Subsequently, the steps of the EKF algorithm explained in Section 2.3 and illustrated in Fig. 1, are executed yielding $\hat{\mathbf{x}}^{\text{dyn}}_{k|k}$. Mass conservation of lithium is taken into account during state correction via the computed Kalman gain $\mathbf{k}_k$ [53]. Then the PP-p2D model is solved for the remaining unaltered four variables. In `EKF_JacobianCalc`, the Jacobian of the model equations for these is computed. The implementation of `SolveBandLES` is discussed in detail in Section 3.4. Lastly, `TemperatureCalc` updates the average temperature.

3.3. Numerical precision and compiler settings

The 32-bit AURIX TC387 can only handle 32-bit floating-point numbers. A significantly slower software emulation can be used for 64-bit [62], also called double-precision. While binary64 has a precision of about 16 digits in decimal format, binary32 has only approximately 7 digits' precision [62]. Thus, numerical errors occur more easily when adding or subtracting numbers of very different magnitudes. Some of the studies mentioned in the introduction developed and tested their models and state estimation algorithms in MATLAB [43]. In the absence of detailed information, it is assumed that the computations were, by default, performed in double-precision [63]. To the best of the authors' knowledge, no work explicitly dealt with a 32-bit implementation of a PP-p2D model and analyzed the influences of reduced numerical precision. Therefore, this is one of the research questions discussed in Section 4.

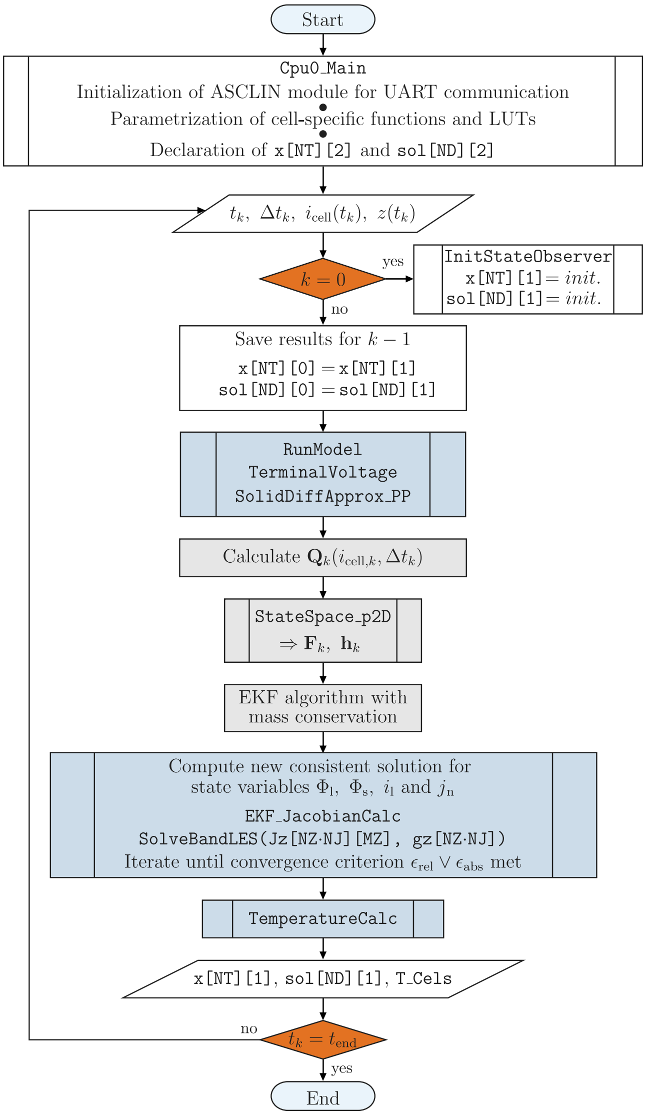

Start
Cpu0_Main
Initialization of ASCLIN module for UART communication
Parametrization of cell-specific functions and LUTs
Declaration of x[NT][2] and sol[ND][2]
$t_k,\ \Delta t_k,\ i_\mathrm{cell}(t_k),\ z(t_k)$
$k=0$
yes
no
InitStateObserver
x[NT][1]= init.
sol[ND][1]= init.
Save results for $k-1$
x[NT][0] = x[NT][1]
sol[ND][0] = sol[ND][1]
RunModel
TerminalVoltage
SolidDiffApprox_PP
Calculate $\mathbf{Q}_k(i_{\mathrm{cell},k},\Delta t_k)$
StateSpace_p2D
$\Rightarrow \mathbf{F}_k,\ \mathbf{h}_k$
EKF algorithm with mass conservation
Compute new consistent solution for state variables $\Phi_\mathrm{l},\ \Phi_\mathrm{s},\ i_\mathrm{l}$ and j_n
EKF_JacobianCalc
SolveBandLES(Jz[NZ·NJ][MZ], gz[NZ·NJ])
Iterate until convergence criterion $\epsilon_\mathrm{rel} \vee \epsilon_\mathrm{abs}$ met
TemperatureCalc
x[NT][1], sol[ND][1], T_Cels
no
$t_k = t_\mathrm{end}$
yes
End

Fig. 3. Program flowchart illustrating the full state observer implementation consisting of PP-p2D model (in blue) and EKF (in gray). Also included is the initialization of the asynchronous synchronous interface (ASCLIN) module for universal asynchronous receiver transmitter (UART) communication from the MCU to the personal computer as well as the model parameterization including LUTs and the declaration of selected variables within the `Cpu0_Main.c` file. The two-dimensional array `x[NT][2]` refers to the state vector and `sol[ND][2]` contains $\bar{c}_\mathrm{s}$ and $\bar{q}_\mathrm{s}$ evaluated at every node. Both vectors contain the values for the time step $k-1$ and k. The program is executed until t_end, where no new inputs are received. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The compiler settings heavily influence performance analyzes of EMs, but have rarely been reported in this context so far. For this work, the non-commercial TASKING compiler v1.1r4 was used in combination with the integrated development environment *AURIX Development Studio* V1.2.0–1.5.0. The floating-point model chosen is `--fp-model` = 1, which stands for *precise* and uses more accurate library functions [64]. As a starting point and unless otherwise specified, code optimization by the compiler is disabled `--optimize` = 0 (-O0). This deselects features such as *constant propagation*, *constant folding* and *expression simplification*. For further details, the reader is referred to the documentation [64]. The default level for optimization is level 2 (-O2) [64]. In this context, the option `--tradeoff` = 4 is chosen, which defines optimization for code size rather than speed.

Table 1
Mapping of symbols to model equation and the included state variables.

Symbol	Model equation	Included state variables
g_1	Li⁺ mass balance in electrolyte	$c_\mathrm{l},\ i_\mathrm{l}$
g_2	Ohm's law in electrolyte	$c_\mathrm{l},\ \Phi_\mathrm{l},\ i_\mathrm{l}$
g_3	Electrode/charge transfer kinetics	$c_\mathrm{l},\ \Phi_\mathrm{l},\ c_\mathrm{ss},\ \Phi_\mathrm{s},\ j_\mathrm{n}$
g_4	Li⁺ mass balance in the particles	$c_\mathrm{ss},\ j_\mathrm{n}$
g_5	Charge balance	$i_\mathrm{l},\ j_\mathrm{n}$
g_6	Ohm's law in electrode	$\Phi_\mathrm{s},\ i_\mathrm{l}$

3.4. Newton's method with banded Jacobian

Calculation of the inverse Jacobian $\mathbf{J}_g^{-1}$ is required to be able to apply the Newton–Raphson method. A block tridiagonal structure $\mathbf{J}_g \in \mathbb{R}^{(n_x\cdot n_j)\times(n_x\cdot n_j)}$ results from the numerical discretization with three blocks, each having a size of $n_x \times n_x$ [53]. Thus, for $n_x = 6$ and $n_j = 6+2+6+1 = 15$ (see Fig. 1), the size is 90 × 90, consuming 32.4 kB of RAM in binary32. Instead of arranging the blocks for the partial derivatives of $\mathbf{g}$ at nodes $j-1$, j and $j+1$ along the diagonal, they can also be arranged in the columns of a banded matrix $\mathbf{J}_\mathrm{B}$ as shown in Eq. (A.1) in parts for rows $n_x \cdot j$ to $n_x \cdot j+5$ with $j \in [1; n_j-2]$. The columns of the banded Jacobian correspond to the diagonals of $\mathbf{J}_g$ with column $(b-1)/2$ representing the main diagonal. The dimension of $\mathbf{J}_\mathrm{B}$ is $n_x \cdot n_j \times b$ with a bandwidth $b = 2\cdot(2\cdot n_x - 1)+1 = 23$. Thus, the memory footprint is reduced to 8.28 kB, which is only 25.6% of the original value. It is worth noting that the memory dependence on the number of nodes is now only linear $\mathcal{O}(n_j)$ compared to $\mathcal{O}(n_j^2)$ for $\mathbf{J}_g$. This allows the use of significantly more discretization nodes in embedded systems with limited RAM, which is beneficial for MRDE as discussed in Section 2.2.

After transforming the matrix into the band structure, the following system of equations has to be solved to apply the Newton–Raphson method:

$$\mathbf{J}_\mathrm{B}\left(\mathbf{x}_k^{(n-1)}\right)\cdot\Delta\mathbf{x}_k^{(n-1)} = \mathbf{g}\left(\mathbf{x}_k^{(n-1)}\right). \tag{6}$$

To minimize memory requirements, in `SolveBandLES` an in-place algorithm is implemented in contrast to previous work [44] to solve Eq. (6) by LU decomposition based on the Fortran code given by Thorson [65]. For the decomposition $\mathbf{J}_\mathrm{B} = \mathbf{L}_B\mathbf{U}_B$ only the memory space marked in Eq. (A.1) is used. The unit diagonal of the lower triangular matrix $\mathbf{L}_\mathrm{B}$ is not explicitly stored [65] and the solution $\Delta\mathbf{x}_k$ is written to the memory previously allocated for $\mathbf{g}$. The LU decomposition of a matrix to solve a system of linear equations is a standard technique. Thus, the reader is referred to available literature [65,66] for a detailed description.

Given that, e.g., partial derivatives of the Butler–Volmer equation $\partial g_3/\partial c_\mathrm{ss}$ are zero in the electrolyte region, a partial pivoting method was used for LU decomposition. It takes the nearest nonzero element in the rows below the main diagonal as the pivot element. Due to the banded matrix approach, there may be insufficient memory after swapping rows resulting from pivoting, usually requiring additional $(b-1)/2$ columns to be allocated [65]. However, by carefully deciding on the equations $g_1, g_2, \ldots, g_6$ and the order of the state variables in $\mathbf{x}_k$, a configuration can be found that does not require this additional memory. This is possible because not all of the model equations depend on all six state variables, and therefore some partial derivatives yield further zero entries in the Jacobian. In the event of model extensions this approach will have to be reviewed. An implemented exception handling routine serves to detect a possible out-of-bounds access. The order of $\mathbf{x}_k$ can be read from Eq. (A.1); for the model equations it is given in Table 1.

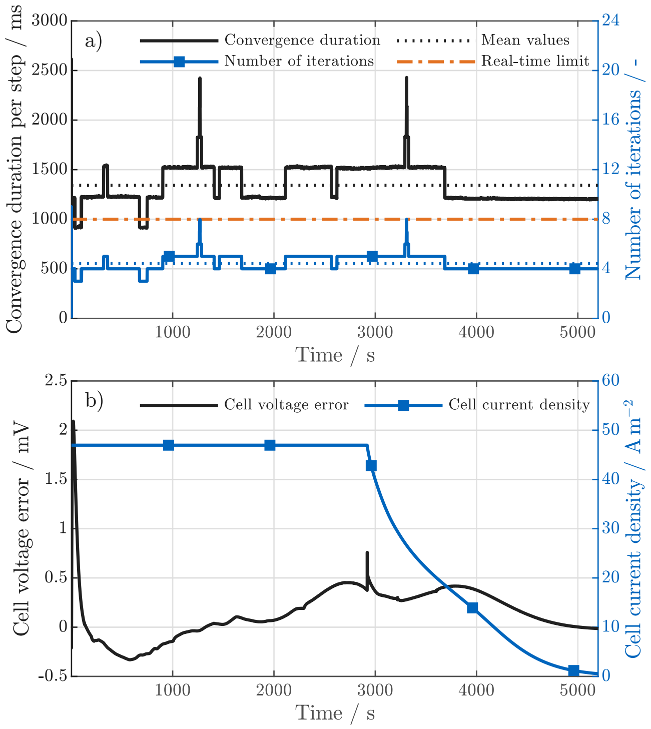


Fig. 4. Results of the 6-2-6 PP-p2D model in binary64 precision. Subplot (a) displays the convergence duration per time step Δt plotted versus time during CC/CV charging with 1C and termination of CV phase at C/100. The number of iterations per time step is also given. The dotted lines indicate the respective average values. The orange line denotes the limit of the real-time capability for the chosen temporal step size of $\Delta t = 1$ s. Subplot (b) shows the absolute cell voltage error compared to the MATLAB code with identical discretization and the charge current density.

4. Results and discussion

4.1. Reference implementation of the PP-p2D model

Starting from other work that has implemented and evaluated electrochemical models in MATLAB or MCUs, the PP-p2D model is first implemented with binary64 precision and using the Gauss–Jordan (GJ) method [44] for matrix inversion on the AURIX TC387 microcontroller.

The results for a 6-2-6 discretization (see Section 2.2) are plotted in Fig. 4. At almost all times during the charging process, the model is not real-time capable, because the convergence duration is almost always greater than the temporal step size of $\Delta t = 1$ s (orange line). The mean value is $\bar{t}_{\mathrm{iter,64}} \approx 1.34$ s. Considering the higher number of discretization nodes and a possibly different implementation of the software emulation for 64-bit floating-point handling, this value is in a reasonable range compared to previous work [44]. On average, the Newton–Raphson approach requires $\bar{n}_{\mathrm{iter,64}} \approx 4.42$ iterations, and a maximum of 8 iterations when Φ_{s} takes very small values, as analyzed in detail in Section 4.3.

Comparing the calculated cell voltage of the binary64 C-code and the MATLAB code V_{err} reveals a mean absolute error of only $\overline{|V_{\mathrm{err}}|} \approx 0.225$ mV, using the built-in matrix inversion function in MATLAB instead of the GJ method. Thus, the total error introduced by the microcontroller implementation is small. Therefore, in the remainder of this section, this implementation will be used as a reference instead of the MATLAB code.

4.2. Single precision code and error analysis

As described in Section 3.3, using binary64 on a 32-bit microcontroller architecture has significant disadvantages. At the same time, possible inaccuracies associated with 32-bit values must be carefully assessed. All floating point data types in the C-code of the PP-p2D model have been changed to binary32.

Table 2
Model accuracy and computational performance of the PP-p2D model implementation in binary32 precision on AURIX microcontroller. Sensitivity of parameters dependent on the choice of relative iteration termination criterion.

Configuration	PP-p2D model				
	For 1C CC/CV charging, $I_{\mathrm{CV}} \leq 0.01\mathrm{C}$				
Relative limit ϵ_{rel} / –	1×10^{-4}	5×10^{-4}	1×10^{-3}	5×10^{-3}	1×10^{-2}
$\bar{n}_{\mathrm{iter}}$ / –	4.80	3.98	3.84	3.18	2.64
$\bar{t}_{\mathrm{iter,32}}$ / ms	123.37	103.57	99.92	82.78	68.50
Time saving[a] / %	90.79	92.28	92.54	93.82	94.89
$\bar{V}_{\mathrm{err,abs}}$[a] / mV	0.1910	0.1908	0.1901	0.1989	0.2367
$\bar{T}_{\mathrm{err,abs}}$[a] / K	0.0144	0.0116	0.0144	0.0144	0.0153
Relative error of local state variables[b]					
$\bar{c}_{\mathrm{l,err}}$ / %	0.0563	0.0564	0.0589	0.0577	0.0680
$\bar{\Phi}_{\mathrm{l,err}}$ / %	0.3056	0.3055	0.3092	0.2351	0.3242
$\bar{c}_{\mathrm{ss,err}}$ / %	0.0798	0.0798	0.0801	0.0702	0.2874
$\bar{\Phi}_{\mathrm{s,err}}$ / %	0.0102	0.0102	0.0103	0.0100	0.0098
$\bar{i}_{\mathrm{l,err}}$ / %	0.4286	0.4286	0.4276	0.3572	0.5996
$\bar{j}_{\mathrm{n,err}}$ / %	1.6003	1.5952	1.6084	1.4820	2.3548

[a] In comparison to the binary64 implementation with $\bar{t}_{\mathrm{iter,64}} \approx 1.34$ s for $\epsilon_{\mathrm{rel}} = 1 \times 10^{-4}$.
[b] The mean relative errors for the individual state variables x_i were calculated as $|(x_{i,64} - x_{i,32})/x_{i,64}|$. The maximum error of all discrete nodes $j = 0, 1, \dots, n_{j-1}$ is given for each state variable.

To verify that the global model outputs are well represented in binary32 precision, the results of this implementation are presented in Table 2. For the relative termination criterion $\epsilon_{\mathrm{rel}} = 1 \times 10^{-4}$, the mean number of iterations per time step $\bar{n}_{\mathrm{iter}}$ is about 8.6% greater than the binary64 version. This can be attributed to slightly higher numerical errors when solving Eq. (6) in binary32 precision. For the iteration time $\bar{t}_{\mathrm{iter,32}}$, speeding up by a factor greater than 10 is observed in all the evaluated scenarios, which already represents a considerable improvement in the real-time capability. The absolute mean errors for cell voltage $\bar{V}_{\mathrm{err,abs}} \approx 0.20$ mV and temperature $\bar{T}_{\mathrm{err,abs}} \approx 15$ mK are small. A distinct inverse dependence on ϵ_{rel} is observed for the number of iterations. However, for the calculated model errors, a relevant change is observed only for $\epsilon_{\mathrm{rel}} > 5 \times 10^{-3}$, which is why this value is considered a good compromise based on this sensitivity analysis. Here, the mean iteration count $\bar{n}_{\mathrm{iter}} = 3.18$ is 28% less than in binary64 precision.

Similarly, to confirm that the local state variables can also be well described with reduced numerical precision, the introduced error is investigated separately for each of the six state variables, as shown in the lower part of Table 2. Except for the configuration with $\epsilon_{\mathrm{rel}} = 1 \times 10^{-2}$, the errors are again found to be relatively insensitive to the convergence limit selection. For all variables, the relative error even decreases slightly for increasing ϵ_{rel}. Hence, this analysis for 1C constant current (CC) / constant voltage (CV) charging supports setting $\epsilon_{\mathrm{rel}} = 5 \times 10^{-3}$. The largest relative error is observed for the pore-wall molar flux across the interfacial area between electrolyte and solid particles $\bar{j}_{\mathrm{n,err}} = 1.4820\%$. This is of the order of 10^{-5} mol m^{-2} s^{-1} and decreases, especially towards the end of the CV phase, to 1×10^{-7} mol m^{-2} s^{-1}. Given these small numbers, it seems plausible that the introduced error for this variable is comparatively large. The error is less than 0.4% for all other variables.

4.3. Optimization of convergence stability

Along with the number of cycles n required for the convergence of the solution $\mathbf{x}_k^{(n)}$ at one time step using the Newton–Raphson approach, the fluctuation of n_{iter} over different points in time during a charge profile is investigated. The maximum convergence duration generally limits the real-time capability. An analysis of n_{iter} over time reveals that

Table 3
Sensitivity analysis on how the convergence stability depends on the choice of the absolute iteration termination criterion for the potential in solid.

Configuration	PP-p2D model, $\epsilon_{\mathrm{rel}} = 5\times10^{-3}$				
	For 1C CC/CV charging, $I_{\mathrm{CV}} \leq 0.01\mathrm{C}$				
	binary64[a]	binary32			
Limit $\epsilon_{\mathrm{abs}}(\Phi_{\mathrm{s}})$ / –	1×10^{-10}	1×10^{-10}	1×10^{-8}	1×10^{-6}	1×10^{-4}
Constant current (CC) phase					
$\bar{n}_{\mathrm{iter}}$ / –	4.4726	3.1949	3.1949	3.1911	3.1089
$n_{\mathrm{iter,max}}$ / –	8	15	15	5	4
$n_{\mathrm{iter,min}}$ / –	3	1	1	1	1
Constant voltage (CV) phase					
$\bar{n}_{\mathrm{iter}}$ / –	4.3493	3.1690	3.1960	3.1633	3.0000
$n_{\mathrm{iter,max}}$ / –	8	13	13	5	3
$n_{\mathrm{iter,min}}$ / –	4	3	3	3	3

[a] The relative iteration termination criterion of the model implemented using binary64 remained at the original value $\epsilon_{\mathrm{rel}} = 1\times10^{-4}$.

Table 4
Comparison of computational speed and memory requirements of the GJ method on full matrix and solution of linear equation system (LES) obtained by LU decomposition on banded matrix.

	Speed / ms	Memory[a] / kB	FLOP[b] / –
Solution via GJ method for $\mathbf{J_g}$			
Matrix inversion $\mathbf{J_g^{-1}}$	16.67	64.80	289,034
Matrix–vector mult. $\mathbf{\Delta x} = \mathbf{J_g^{-1}g}$	0.76	33.12	16,200
Total	17.43	65.52	305,234
Solution via LU decomposition for $\mathbf{J_B}$			
LU decomposition $\mathbf{L_B U_B}$	1.69	8.28	20,902
Forward substitution $\mathbf{L_B y} = \mathbf{g}$	0.08	8.64	1,848
Backward substitution $\mathbf{U_B \Delta x} = \mathbf{y}$	0.07	8.64	1,938
Total	1.84	8.64	24,688

[a] Memory required to store matrices and vectors, which is reused after one-time initialization.
[b] Total number of floating-point operations (FLOP) comprising `add.f`, `sub.f`, `mul.f`, `div.f`, `madd.f` and `msub.f` [67], the last two being compound operations and therefore counted twice.

instabilities occur where Φ_s takes very small absolute values $|\Phi_s| \lessapprox 1\times10^{-3}$ in the anode domain. The solid potential during charging is below the anode OCP due to the voltage drop caused by the electric current i_s within the active material. This is why zero crossings of the value of this state variable occur. An adjustment of the absolute convergence criterion for Φ_s alone was investigated to remedy this instability. In this way, the termination of the main iteration loop could be locally decoupled from the convergence of the specific variable.

The sensitivity analysis in Table 3 reveals a reduction of the maximum number of iterations $n_{\mathrm{iter,max}}$ from 15 to 4 for the CC phase and from 13 to 3 for the CV phase with the setting $\epsilon_{\mathrm{abs}}(\Phi_s) = 1\times10^{-4}$. These maximum numbers are very close to the mean values of both phases and thus indicate that the goal of stability optimization is achieved. Again, the influence of this adjustment on the relative errors of the local state variables according to Table 2 is investigated. This is below 0.1% for all state variables and therefore not examined any further.

4.4. Further improvements in real-time capability

First, the computation time is analyzed when using the method described in Section 3.4 to solve Eq. (6). The code following an explicit GJ approach, as described in Section 4.1, is used as a reference. A computation time of 17.43 ms is achieved for the 90 × 90 matrix resulting from the discretization described in Section 2.2, using binary32 precision according to Table 4. By comparison, using the algorithm for $\mathbf{J_B}$ (see Section 3.4), a total time of 1.84 ms is obtained, which corresponds to a reduction of 89.4%. Analyzing the number of floating-point operations (FLOP) confirms that this reduction in computation time mainly results from the reduction of the number of FLOP as this decreases by 91.9% and is thus a similar value. The memory footprint is also significantly lower. A total of 65.52 kB of memory is required for $\mathbf{J_g}$, $\mathbf{J_g}^{-1}$, $\mathbf{g}$, and $\mathbf{\Delta x}$ for the GJ method. By comparison, only 13.2% of this amount of memory is used for $\mathbf{J_B}$ and $\mathbf{g}$ when applying the optimized in-place approach.

The performance of the whole PP-p2D model is now evaluated using this method to solve the LES as part of the Newton–Raphson method, and the real-time capability is measured taking into account the above reported improvements. To analyze the influence of the compiler settings explained in Section 3.3, these are varied and the effect on computation time as well as memory requirements is measured, again to allow comparisons for 1C CC/CV charging.

Without code optimization by the compiler (-O0), the mean execution time is 32.4 ms according to Table 5. This is equal to the mean number of iterations $\bar{n}_{\mathrm{iter}} = 3.0609$ for $\epsilon_{\mathrm{abs}}(\Phi_s) = 1\times10^{-4}$ derived from Table 3 multiplied by the execution time of `JacobianCalc` and `SolveBandLES`. The difference from the reported time for `RunModel` is due to updates of solid concentration profiles and the averaged temperature calculation, performed once per time step. The required flash memory space is subdivided into that required for program code, consisting of executable program instructions, and that for data. Data comprises global as well as all static variables (e.g., mathematical constants and model parameters). These are mainly the lookup tables containing the data for the anode and cathode OCP separately for lithiation and delithiation (2 ⋅ 15 kB + 2 ⋅ 1.5 kB), their derivatives (2 ⋅ 12 kB + 2 ⋅ 1.1 kB), and the entropy coefficients (2 ⋅ 0.8 kB). In addition, the data segment contains the initial values for arrays with runtime initialization.

The code optimization by the compiler affects speed and code size, but not the memory requirement for data. The first level of optimization (-O1) yields a reduction in the convergence duration of 12.7% and a reduction of 25.1% in the code size. For level -O2 an additional reduction in the execution time of 9.9% is achieved. In addition, varying the trade-off between speed and code size only results in small changes to the execution time of `RunModel`. It can be concluded that compiler optimization allows an improvement in the execution time of about 22%. This is particularly remarkable because it does not influence model calculation accuracies.

4.5. Closed-loop formulation based on EKF

After optimizing the implementation of the PP-p2D model, it can now be used as part of the observer framework for state estimation as described in Section 3.2. First, different CC/CV charge profiles are examined because these are commonly used in the field. Fig. 5 presents the results obtained for current rates of 0.5C, 1C, and 2C during the CC phase. At a time $t_0 = 200$ s during charging, the state estimator (in green / light blue) is activated. Then, using the voltage values, it adjusts the state of the embedded PP-p2D so that a convergence to the plotted reference (dark blue, dashed) occurs. Previously calculated simulations of the charging sequences performed with the PP-p2D model with identical parameterization are used as reference in Fig. 5a, c, and e. The voltage *measurement* values, z_k, are emulated from the simulation and corrupted with an additive white Gaussian noise with a standard deviation $\sigma = 1$ mV. This is a rather conservative assumption compared to the attainable relative accuracy of cell voltage measurements (CVMs) of modern battery monitoring integrated circuits (e.g., Ref. [68]).

The absolute estimation error (AEE) of the cell voltage falls below 5 mV after periods of 226 s, 193 s, and 349 s following activation of the state observer for 0.5C, 1C, and 2C, respectively. Thereafter, the AEE decreases further and remains in the range of the standard deviation of the introduced noise for the profiles analyzed. In the CV phase, Fig. 5b reveals a slight increase of the maximum AEE for $I < 0.1$C, which can be attributed to the range switching introduced in Eq. (5). However,

Table 5
Performance of the optimized 6-2-6 PP-p2D model implementation in C-code on AURIX microcontroller.

	Compiler settings			
	Code optimization `--optimize`	Trade-off `--tradeoff`	Speed[b]/ms	Code size/kB
`JacobianCalc`	-O0	4	8.35	18.8
`SolveBandLES`			1.84	0.97
`RunModel`[a]			32.4	31.1
`RunModel`[a]	-O1	4	28.3	23.3
`RunModel`[a]	-O2	4	25.5	20.1
		2	25.3	21.5
		0	25.2	22.7

[a] Complete PP-p2D model including initialization, update of concentrations in the active materials as well as calculation of current averaged cell temperature.

[b] Mean number of required clock cycles divided by maximum frequency of MCU $f_{\text{clk}} = 300$ MHz for 1C CC/CV charging, $I_{\text{CV}} \leq 0.01$C.

this still positively affects the stability of the state estimate and allows its use even below the currents presented here at less than C/500.

In addition to the AEE of the cell voltage, the ability of the state observer to correctly recover the model internal states is of particular interest. In Fig. 5c, the estimated surface concentration c_{ss} is shown for the anode and cathode domains. The nodes closest to the current collectors were evaluated and plotted normalized by the corresponding maximum concentrations. The utilized SoC range of the individual electrodes is less than 100% [27] due to the specific balancing of the cell in use. For the cathode, the AEE of the state observer is less than 0.5% at 342 s, 431 s, and 808 s for 0.5C, 1C, and 2C, respectively. The AEE for the anode also falls below this threshold after 139 s, 154 s, and 329 s, respectively. Even for 2C, the mean absolute error for the SoC estimation in the anode is only 0.07% starting from the AEE of less than 0.5%. The full curves are shown in Fig. 5d.

During fast charging of lithium-ion cells, the anode potential difference at the anode–separator interface between solid and electrolyte potential $\Phi_{\text{s}}(x_{\text{a,sep}}) - \Phi_{\text{l}}(x_{\text{a,sep}})$, plotted in Fig. 5e, is an important parameter. If it drops below 0 V vs. Li/Li$^+$, lithium ions may be reduced to lithium metal at the surface of the anode instead of intercalating into the anode active material particles [69,70]. This effect is called lithium plating and may not be fully reversible, resulting in loss of lithium inventory, loss of active material due to partial shielding of the anode, and other degradation effects negatively affecting the battery lifetime [71]. Precise knowledge of the temperature, SoC, and current-dependent anode potential is therefore essential for an optimal fast charging protocol for the cells. The PP-p2D model embedded in the state observer now enables real-time estimation of this potential and shows good agreement with the reference simulation. As soon as 350–400 s after activating the observer, the AEE for all charge profiles is below 2 mV in magnitude for a step size of 1 s almost irrespective of the current rate, as can be seen in Fig. 5f.

These results demonstrate that the closed-loop formulation is able to quickly recover the cell voltage as well as important model internal states under unknown SoC and during different CC/CV charging profiles. However, it remains unclear whether this can be sustained for larger initial SoC errors than those resulting from activating the estimation at $t_0 = 200$ ms. Additionally, the stability when applied to dynamic charge profiles must be evaluated. Both aspects will be treated in Section 4.6. Prior to this, the performance in terms of computation time and memory requirements of the complete state observer is studied.

The convergence duration and the number of iterations per step are given in Fig. 6. For a CC/CV charge profile at 2C, the convergence duration for the PP-p2D model prediction (downward-pointing triangle, black) has a mean value of 31.4 ms. It is thus marginally lower than the convergence duration shown in Table 5 for 1C of 32.4 ms, which is due to minor differences in the number of iterations. The complete state observer takes 62.3 ms for a time step of 1 s and is thus about 16 times faster than real-time. Considering the number of iterations needed to meet the convergence criteria of the Newton–Raphson method (downward-pointing triangle, blue), the number of iterations

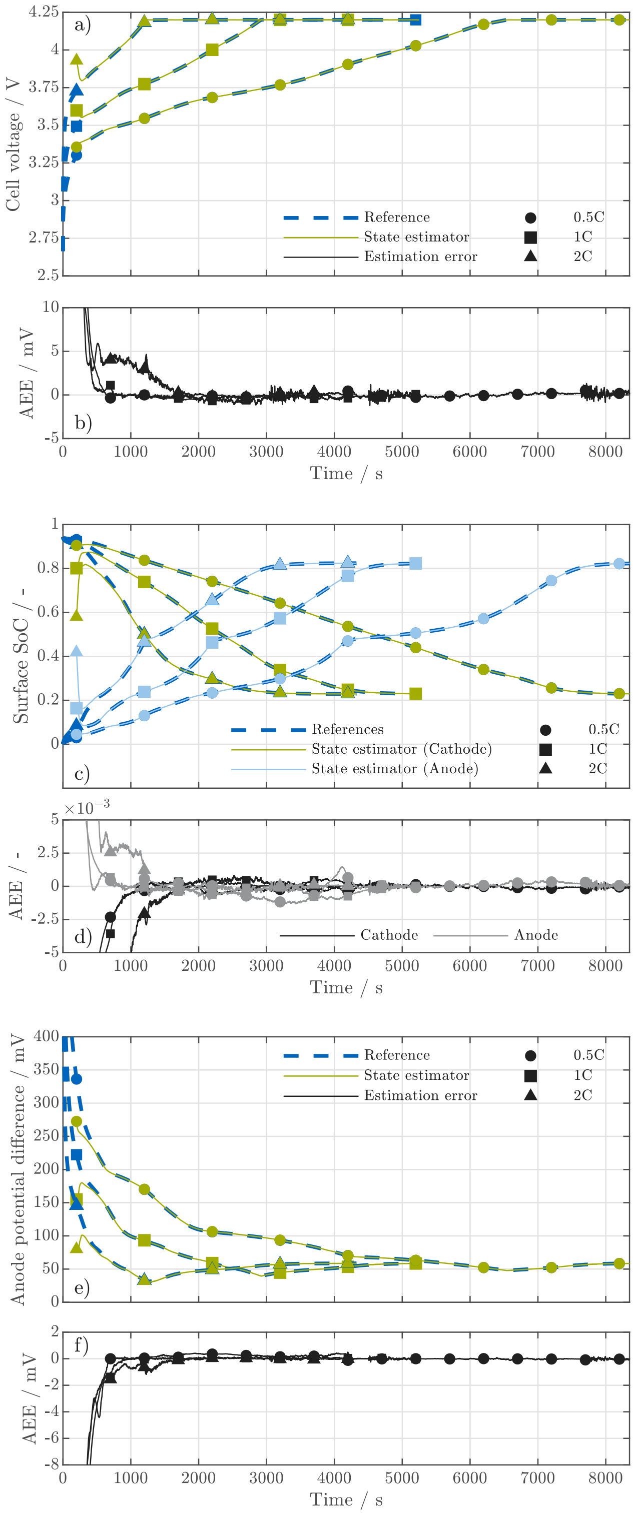


Fig. 5. Results of the state observer and study of the absolute estimation error (AEE) for the cell voltage (a–b), surface SoC (c–d), and potential difference at the anode–separator interface (e–f) on AURIX microcontroller. The analyzes were performed for CC/CV charging at 0.5C, 1C, and 2C and termination of CV phase at C/100 (C/500 for 2C scenario).

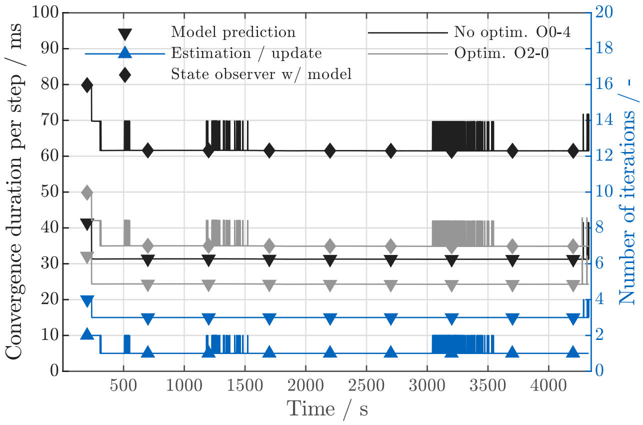


Fig. 6. Performance of the state observer including PP-p2D model on AURIX microcontroller. For the CC/CV charging at 2C already used in Fig. 5, the convergence duration and the number of iterations per time step are plotted. The *a priori* result of the PP-p2D model is referred to as the model prediction, the adjustment of the state variables by the EKF is called estimation/update step. In addition, the results are given for two compiler settings O0-4 and O2-0.

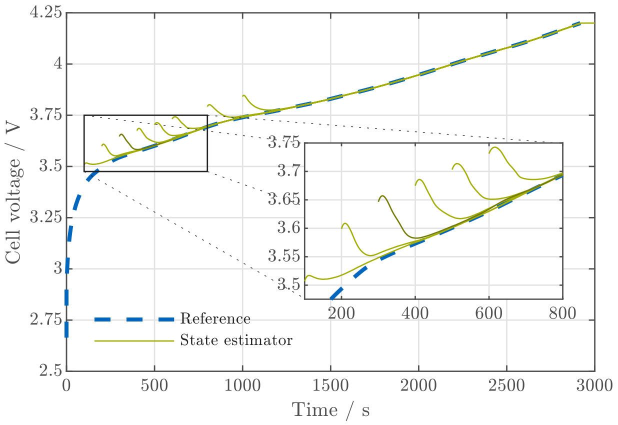


Fig. 7. Influence of the initial SoC error on the convergence of the state observer for a CC charge profile at 1C.

for the model prediction is almost constant at 3. Only immediately after the activation of the observer algorithm (for 36 s) and at the end of the CV phase from $I < \mathrm{C}/450$, the number of iterations partially increase to 4 per step. One to two additional iterations are executed (upwards-pointing triangle, blue) for the final calculation of a consistent solution after the estimation of the dynamic state variables $\hat{\mathbf{x}}_{j,k}$.

The influences of different compiler settings, introduced in Section 3.3 for the PP-p2D model, have already been examined in Table 5. In it, a decrease of the convergence duration of about 22% was found if -O2 rather than -O0 was chosen for the code optimization. The same time savings can also be observed for the profile shown in Fig. 6. However, the convergence duration for the full state observer is only 35.6 ms on average with code optimization enabled (gray) and is therefore cut by more than 42%. The reason for this significant improvement is that the EKF algorithm requires the computation of several matrix–matrix and matrix–vector products, which can be optimized using various mechanisms by the compiler. As an example, the *a priori* calculation of the state prediction error covariance matrix $\mathbf{P}_{k|k-1}$ is analyzed:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathsf{T}} + \mathbf{Q}_k \,. \tag{7}$$

Using the *a posteriori* result $\mathbf{P}_{k-1|k-1}$ of the previous time step, the covariance matrix of the state estimation error is updated via the state transition matrix $\mathbf{F}_k$ and by adding process noise $\mathbf{Q}_k$. For the two matrix multiplications and the addition about 12.8 ms are needed with the settings -O0-4, whereas the execution only takes 1.5 ms when optimization -O2-0 is enabled. The main reason for this speedup is that the array elements are only loaded into registers once and are then used for multiple calculations. Compared to the non-optimized implementation, many load operations and the address calculations of the array elements are omitted, the latter being further simplified by *constant propagation* and *constant folding*. These results emphasize the importance of the actual implementation and optimization on an application-oriented MCU for assessing the achievable real-time capability of the EKF as a state observer incorporating the presented PP-p2D model. With a total of 35.6 ms for a step size of 1 s, it runs 28 times faster than real-time and does not involve any further model simplifications.

Next, the memory requirements of the full state observer are briefly analyzed and compared to the values for the PP-p2D model given in Table 5. The top-level function `RunStateObserver` requires 82.7 kB of flash memory for data and thus 19.9 kB more than the model itself. A major part of this is used for the matrices of the EKF, which are generated as local arrays with runtime initialization. Without code optimization (-O0) and a trade-off with a focus on size (4), the code of the full estimation algorithm requires 45.6 kB flash memory, which is 13.2 kB more or about 41% when compared to the model without EKF. With code optimization -O2, the memory footprint decreases to 30.4 kB, thus only 19% more than the bare PP-p2D model. If the trade-off is additionally shifted towards speed as implemented in the configuration shown in Fig. 6, the resulting code size is 34.5 kB. Therefore, the overall flash memory requirement for the latter configuration is about 117.2 kB, which is less than 1.5% of the available memory of the microcontroller being used. This is why optimization for speed rather than code size seems to be a very reasonable approach. An analysis of the required SRAM, which is independent of the compiler settings, indicates use of 59.3 kB corresponding to only 19.5% of the total SRAM available per core. This also includes the memory for the banded Jacobian $\mathbf{J}_\mathrm{B}$ (8.28 kB) stored as a global variable.

To summarize, the study of computation time and memory requirements of the state observer implementation with integrated optimized 6-2-6 PP-p2D model reported in this paper has revealed that a mean convergence duration of 35.6 ms (50.1 ms maximum) with a 117.2 kB flash memory and 59.3 kB SRAM footprint is achievable for a 2C CC/CV charge profile. The PP-p2D model accounts for 25.2 ms and uses 93.9 kB of flash memory. Given a step size of 1 s, this leads to a utilization level of the MCU of only 3.6%, with a flash memory utilization of less than 1.5%. Compared to the reference implementation presented in Section 4.1, this translates into a saving of 98.1% for the PP-p2D model.

4.6. Stability analysis for dynamic charge profile

First, the influence of the initial SoC error on the convergence of the EKF is studied. This can be controlled by the choice of the start time t_0. In Fig. 7, multiple start-up phases of the state estimator are plotted for different t_0. Except for $t_0 = 300$ s, the time until the voltage residual y_k falls below 5 mV is found to be less than 250 s. For $t_0 = 300$ s, the residual also initially falls below 5 mV, but then increases again to 6.5 mV before falling below 1 mV as highlighted in dark green in Fig. 7. This might be explained by the shape of the anode OCP, but it is of subordinate importance and therefore will not be further discussed. As a result, it is important to note that when using the initialization method previously described [53], the choice of the start time does not determine the performance of the state observer and robust convergence can be ensured in all cases.

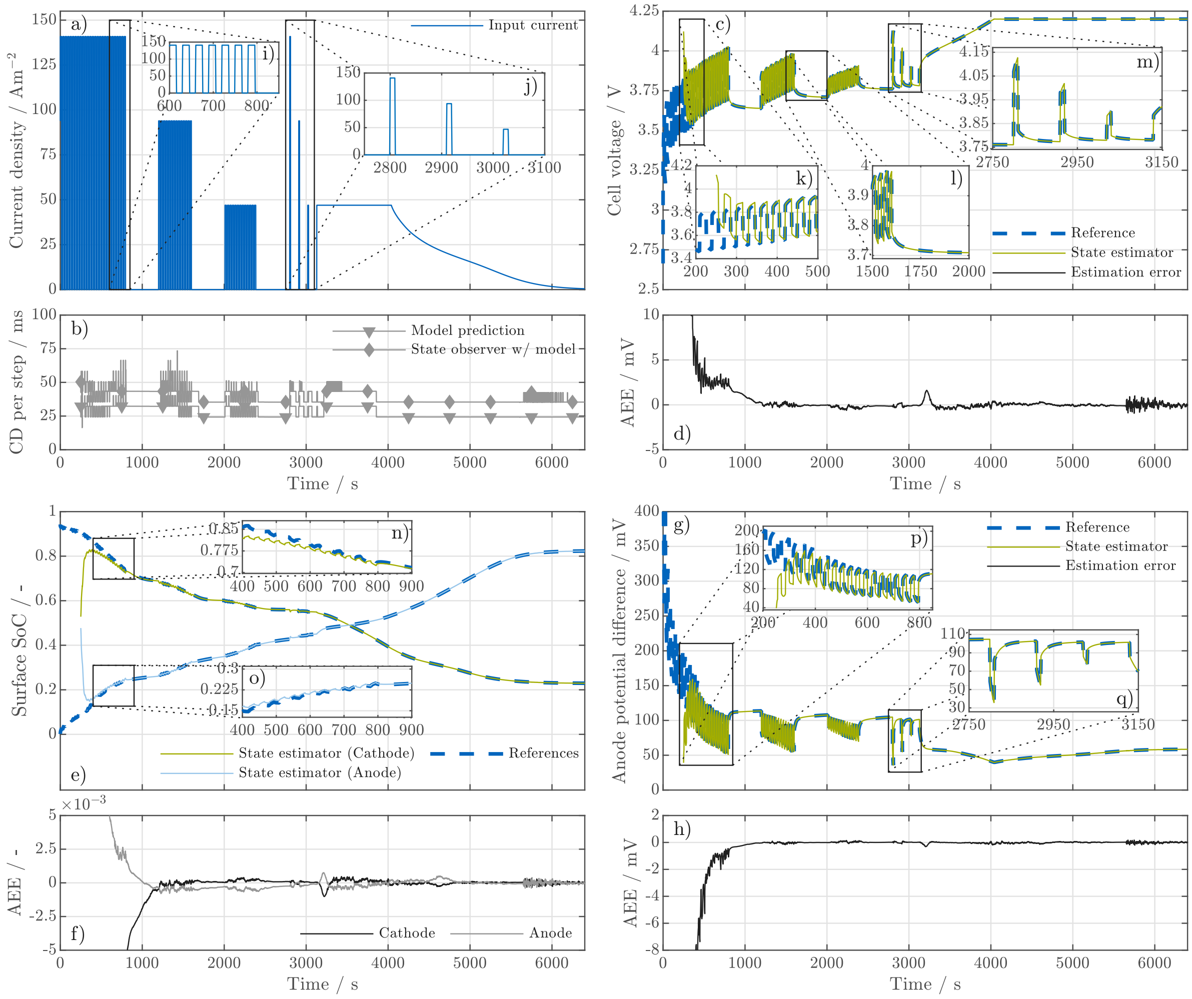


Fig. 8. Performance and robustness study of the state observer for an artificial charge profile at up to 3C with different phases: pulse charging with 15 s constant input current at 3C, 2C, 1C followed by an idle period of 15 s (a, i) and single pulses with a duration of 10 s followed by a relaxation of 100 s (a, j). Prolonged rest phases lasting 400 s after pulse charge sequences and 1C CC/CV charging with termination of CV phase at C/100. Analysis of convergence duration (CD) per time step (b), cell voltage (c, k–m), surface SoC (e, n, o), and potential difference at the anode–separator interface (g, p, o). Supplementary evaluation of the corresponding AEEs (d, f, h).

Next, the performance of the state estimator for the given input current profile depicted in Fig. 8a is examined to check the suitability for future dynamic fast charging strategies. As in Section 4.5, previously calculated noise-corrupted simulations with identical parameterization are used as a reference. After the start of the EKF at $t_0 = 250$ s, Fig. 8k within Fig. 8c depicts the estimated cell voltage compared to the reference. Despite the dynamic 3C pulses, the AEE of the cell voltage shown in Fig. 8d drops below 5 mV from 259 s after activation. During the subsequent 400 s relaxation phase, the AEE for the cell voltage declines, and throughout the following pulse charge sections at 2C and 1C, it does not exceed 0.6 mV. With a standard deviation in this range of around 0.2 mV, the AEE is well below the noise level $\sigma = 1$ mV of the CVM. The surface SoCs in Fig. 8e and the potential difference at the anode–separator interface in Fig. 8g reveal a similar result compared to the CC/CV charge profiles in Fig. 5. The magnified plots in Fig. 8n and o show the solid concentration estimates for cathode and anode. At the end of the 3C pulse charge phase, the AEE of the anode potential according to Fig. 8h is about 1 mV.

After pulse current charging at 3C, 2C and 1C, the synthetic input current profile contains three 10 s duration single current pulses with the same absolute values as shown in Fig. 8j. The cell voltage estimate for this interval, shown in Fig. 8m, is found to be in very good agreement with the reference. The maximum AEE of 0.35 mV is almost negligible. Similarly, the estimates of the model internal states do not have any relevant deviations during the single pulses, as observed in Fig. 8f, h and q.

In Fig. 8b the convergence duration per step is given analogously to Fig. 6 for the compiler setting O2-0. During the single pulses, an increase from about 35 ms up to 51.4 ms can be observed. This is explained by an increasing number of iterations for the Newton–Raphson approach until the convergence criteria are met. During the pulse charge phases, the increase in the convergence duration is even more pronounced. A maximum of 66.5 ms, or even 73.5 ms for a single time step, is required for a step size of 1 s. Still, the mean convergence duration of 44 ms is only about 9 ms above the value for CC/CV charging at 2C studied in Fig. 6. This clearly indicates that the applied optimization of the convergence stability described in Section 4.3 and tested under CC/CV conditions also yields good results for highly dynamic charge profiles.

5. Conclusion

Aiming for accurate state monitoring and operation management, white-box electrochemical models provide in-depth insights into the local internal states of lithium-ion batteries. In this work, the implementation of an EKF-based state observer using the full-order p2D model with PP approximation for solid-state diffusion on an application-oriented state-of-the-art automotive microcontroller is proposed.

$$\mathbf{J}_\mathrm{B} = \left[\begin{array}{cccccccc|c|cccccccc|c}
0 & & & & & & & l & \frac{b-1}{2} & u & & & & & & & b-1 & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \\
0 & 0 & 0 & 0 & 0 & \frac{\partial g_{1,j}}{\partial c_{\mathrm{l},j-1}} & \cdots & \frac{\partial g_{1,j}}{\partial j_{\mathrm{n},j-1}} & \frac{\partial g_{1,j}}{\partial c_{\mathrm{l},j}} & \frac{\partial g_{1,j}}{\partial \Phi_{\mathrm{l},j}} & \frac{\partial g_{1,j}}{\partial c_{\mathrm{ss},j}} & \frac{\partial g_{1,j}}{\partial \Phi_{\mathrm{s},j}} & \frac{\partial g_{1,j}}{\partial i_{\mathrm{l},j}} & \frac{\partial g_{1,j}}{\partial j_{\mathrm{n},j}} & \frac{\partial g_{1,j}}{\partial c_{\mathrm{l},j+1}} & \cdots & \frac{\partial g_{1,j}}{\partial j_{\mathrm{n},j+1}} & n_x \cdot j \\
0 & 0 & 0 & 0 & \frac{\partial g_{2,j}}{\partial c_{\mathrm{l},j-1}} & \cdots & \frac{\partial g_{2,j}}{\partial j_{\mathrm{n},j-1}} & \frac{\partial g_{2,j}}{\partial c_{\mathrm{l},j}} & \frac{\partial g_{2,j}}{\partial \Phi_{\mathrm{l},j}} & \frac{\partial g_{2,j}}{\partial c_{\mathrm{ss},j}} & \frac{\partial g_{2,j}}{\partial \Phi_{\mathrm{s},j}} & \frac{\partial g_{2,j}}{\partial i_{\mathrm{l},j}} & \frac{\partial g_{2,j}}{\partial j_{\mathrm{n},j}} & \frac{\partial g_{2,j}}{\partial c_{\mathrm{l},j+1}} & \cdots & \frac{\partial g_{2,j}}{\partial j_{\mathrm{n},j+1}} & 0 & \\
0 & 0 & 0 & \frac{\partial g_{3,j}}{\partial c_{\mathrm{l},j-1}} & \cdots & \frac{\partial g_{3,j}}{\partial j_{\mathrm{n},j-1}} & \frac{\partial g_{3,j}}{\partial c_{\mathrm{l},j}} & \frac{\partial g_{3,j}}{\partial \Phi_{\mathrm{l},j}} & \frac{\partial g_{3,j}}{\partial c_{\mathrm{ss},j}} & \frac{\partial g_{3,j}}{\partial \Phi_{\mathrm{s},j}} & \frac{\partial g_{3,j}}{\partial i_{\mathrm{l},j}} & \frac{\partial g_{3,j}}{\partial j_{\mathrm{n},j}} & \frac{\partial g_{3,j}}{\partial c_{\mathrm{l},j+1}} & \cdots & \frac{\partial g_{3,j}}{\partial j_{\mathrm{n},j+1}} & 0 & 0 & \\
0 & 0 & \frac{\partial g_{4,j}}{\partial c_{\mathrm{l},j-1}} & \cdots & \frac{\partial g_{4,j}}{\partial j_{\mathrm{n},j-1}} & \frac{\partial g_{4,j}}{\partial c_{\mathrm{l},j}} & \frac{\partial g_{4,j}}{\partial \Phi_{\mathrm{l},j}} & \frac{\partial g_{4,j}}{\partial c_{\mathrm{ss},j}} & \frac{\partial g_{4,j}}{\partial \Phi_{\mathrm{s},j}} & \frac{\partial g_{4,j}}{\partial i_{\mathrm{l},j}} & \frac{\partial g_{4,j}}{\partial j_{\mathrm{n},j}} & \frac{\partial g_{4,j}}{\partial c_{\mathrm{l},j+1}} & \cdots & \frac{\partial g_{4,j}}{\partial j_{\mathrm{n},j+1}} & 0 & 0 & 0 & \\
0 & \frac{\partial g_{5,j}}{\partial c_{\mathrm{l},j-1}} & \cdots & \frac{\partial g_{5,j}}{\partial j_{\mathrm{n},j-1}} & \frac{\partial g_{5,j}}{\partial c_{\mathrm{l},j}} & \frac{\partial g_{5,j}}{\partial \Phi_{\mathrm{l},j}} & \frac{\partial g_{5,j}}{\partial c_{\mathrm{ss},j}} & \frac{\partial g_{5,j}}{\partial \Phi_{\mathrm{s},j}} & \frac{\partial g_{5,j}}{\partial i_{\mathrm{l},j}} & \frac{\partial g_{5,j}}{\partial j_{\mathrm{n},j}} & \frac{\partial g_{5,j}}{\partial c_{\mathrm{l},j+1}} & \cdots & \frac{\partial g_{5,j}}{\partial j_{\mathrm{n},j+1}} & 0 & 0 & 0 & 0 & \\
\frac{\partial g_{6,j}}{\partial c_{\mathrm{l},j-1}} & \cdots & \frac{\partial g_{6,j}}{\partial j_{\mathrm{n},j-1}} & \frac{\partial g_{6,j}}{\partial c_{\mathrm{l},j}} & \frac{\partial g_{6,j}}{\partial \Phi_{\mathrm{l},j}} & \frac{\partial g_{6,j}}{\partial c_{\mathrm{ss},j}} & \frac{\partial g_{6,j}}{\partial \Phi_{\mathrm{s},j}} & \frac{\partial g_{6,j}}{\partial i_{\mathrm{l},j}} & \frac{\partial g_{6,j}}{\partial j_{\mathrm{n},j}} & \frac{\partial g_{6,j}}{\partial c_{\mathrm{l},j+1}} & \cdots & \frac{\partial g_{6,j}}{\partial j_{\mathrm{n},j+1}} & 0 & 0 & 0 & 0 & 0 & n_x \cdot j+5 \\
\vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\multicolumn{8}{c}{\underbrace{\hphantom{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}}_{\text{Memory for } \mathbf{L}_\mathrm{B}}} & & \multicolumn{8}{c}{\underbrace{\hphantom{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}}_{\text{Memory for } \mathbf{U}_\mathrm{B}}} &
\end{array}\right] \tag{A.1}$$

Box I.

Through various optimizations, especially in respect of numerical precision, EKF tuning, and solver criteria, the convergence duration for the PP-p2D model could be reduced by 98.1% compared to a straightforward reference implementation. For CC/CV charge profiles up to 2C, the convergence duration of the full state observer is found to be 28 times faster than real-time, and even for highly dynamic charge profiles up to 3C, it is still at least 13 times faster than real-time. The lean memory footprint of the full state observer consumes less than 1.5% of the available microcontroller flash memory and only 19.5% of the SRAM available per core. These results suggest that the frequently reported increased complexity of electrochemistry-based models in comparison to ECMs is no longer a problem for their utilization in state estimators for battery management systems in electric vehicles. Going forward, increased clock frequencies and optimized hardware architectures are expected to support this trend.

Based on these achievements, a broad field for further research in two main directions emerges: firstly, current and future work of the authors will consider the advanced experimental evaluation of the presented state observer on a real battery cell during operation, focusing on the anode potential as a key limiting factor for preventing lithium plating during fast charging. Based on this, novel adaptive fast charging strategies will be designed. The performance and convergence stability of the state observer for different cathode active materials, especially those with flat regions of the OCV–SoC curve, e.g., lithium iron phosphate (LFP), should also be investigated to prove broad applicability for different cell chemistries. Secondly, future work will investigate the incorporation of aging mechanisms such as solid electrolyte interphase growth together with online parameter identification and adaptation of model parameters during operation. This will enable more accurate modeling of battery function throughout the battery's life as well as more accurate state-of-power prediction and SoH estimation.

CRediT authorship contribution statement

F.F. Oehler: Conceptualization, Methodology, Investigation, Software, Writing – original draft, Visualization, Writing – review & editing. **K. Nürnberger:** Software, Writing – review & editing. **J. Sturm:** Writing – review & editing. **A. Jossen:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Federal Ministry for Economic Affairs and Energy, Germany within the project LImeSI [03ETE019A].

Appendix A. Memory allocation for banded Jacobian

See Eq. (A.1) given in Box I.

Appendix B. Parameterization

See Table B.6.

Table B.6
PP-p2D model parameterization for NMC-811/SiG cell.

	Silicon–graphite (SiG)	Separator	Nickel-rich (NMC-811)
Geometry			
Thickness L	86.67 μm[a]	12.00 μm[a]	66.17 μm[a]
Particle radius R_p	6.08 μm[a]		3.78 μm[a]
Porosity ϵ_l	21.61%[a]	45.00%[a]	17.09%[a]
Active material fraction ϵ_s	69.39%		74.51%[a]
Inactive fraction $\epsilon_\mathrm{s,na}$	9.0%[a]		8.4%[a]
Bruggeman coefficient liquid β	1.5[a]		1.85[a]

(continued on next page)

Table B.6 (*continued*).

	Silicon–graphite (SiG)	Separator	Nickel-rich (NMC-811)
Transport			
Solid diffusivity D_s	5×10^{-14} m^2 s^{-1}[a]		5×10^{-13} m^2 s^{-1}[a]
Solid conductivity σ_s[a]	100 S m^{-1}		0.17 S m^{-1}[b]
Film resistance R_f	3.5×10^{-3} Ω m^2[c]		0 Ω m^2[e]
Thermodynamics			
Equilibrium potential E_{eq}	Ref. [27]		Ref. [27]
Entropic coefficient $\frac{\partial E_{eq}}{\partial T}$	Ref. [27]		Ref. [27]
Max. solid concentration $c_{s,max}$	35 613 mol m^{-3}[e]		50 060 mol m^{-3}[a]
Stoichiometry (0–100% SoC)	0.215–85.24%[a]		94.23–22.24%[a]
Kinetics			
Reaction rate constant k	8×10^{-9} m s^{-1}[e]		8×10^{-9} m s^{-1}[e]
Transfer coefficient $\alpha_{a/c}$	0.5[e]		0.5[e]
Electrolyte	1 M LiPF$_6$ in PC/EC/DMC		
Salt diffusivity D_l / cm^2 s^{-1}	$10^{-4.43-\frac{54}{T-229-5c_l}-0.22c_l}$[d]		
Ionic conductivity κ_l / S m^{-1}	$0.1c_l(-10.5 + 0.0740T - 6.96 \times 10^{-5}T^2 + 0.6680c_l - 0.0178c_lT + 2.80 \times 10^{-5}c_lT^2 + 0.4940c_l^2 - 8.86 \times 10^{-4}c_l^2T)^2$[d]		
Activity $\frac{d \ln f_\pm}{d \ln c_l}$ / –	$(0.601 - 0.240c_l^{0.5} + 0.982(1 - 0.0052(T - 294))c_l^{1.5})(1 - t_+^0)^{-1} - 1$[d]		
Li$^+$ transference number t_+^0 / –	0.38[d]		
Ref. concentration $c_{l,ref}$ / mol L^{-1}	1.0[d]		
Thermal			
Active area A_{act} / m^2	0.07134[f]		
Specific heat capacity c_p / J kg^{-1} K^{-1}	791.86[f]		
Transfer coefficient h_∞ / W m^{-2} K^{-1}	0.58[f]		
Ambient temperature T_{amb} / K	299.15		
Global			
External resistances R_{ext}	13×10^{-4} Ω m^2[e]		

[a] Ref. [27].

[b] Ref. [72].

[c] Ref. [57].

[d] Ref. [73].

[e] Estimated.

[f] Ref. [44].

References

[1] Y. Wang, J. Tian, Z. Sun, L. Wang, R. Xu, M. Li, Z. Chen, A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems, Renew. Sustain. Energy Rev. 131 (2020) 110015, http://dx.doi.org/10.1016/j.rser.2020.110015.

[2] H. Dai, B. Jiang, X. Hu, X. Lin, X. Wei, M. Pecht, Advanced battery management strategies for a sustainable energy future: Multilayer design concepts and research trends, Renew. Sustain. Energy Rev. 138 (2021) 110480, http://dx.doi.org/10.1016/j.rser.2020.110480.

[3] S. Steinhorst, M. Lukasiewycz, S. Narayanaswamy, M. Kauer, S. Chakraborty, Smart cells for embedded battery management, in: 2014 IEEE International Conference on Cyber-Physical Systems, Networks, and Applications, IEEE, 2014, pp. 59–64, http://dx.doi.org/10.1109/CPSNA.2014.22.

[4] R. Zhang, B. Xia, B. Li, L. Cao, Y. Lai, W. Zheng, H. Wang, W. Wang, State of the art of lithium-ion battery SOC estimation for electrical vehicles, Energies 11 (7) (2018) 1820, http://dx.doi.org/10.3390/en11071820.

[5] N. Noura, L. Boulon, S. Jemeï, A review of battery state of health estimation methods: Hybrid electric vehicle challenges, World Electr. Veh. J. 11 (4) (2020) 66, http://dx.doi.org/10.3390/wevj11040066.

[6] J. Rivera-Barrera, N. Muñoz-Galeano, H. Sarmiento-Maldonado, SoC Estimation for lithium-ion batteries: Review and future challenges, Electronics 6 (4) (2017) 102, http://dx.doi.org/10.3390/electronics6040102.

[7] X. Lin, Y. Kim, S. Mohan, J.B. Siegel, A.G. Stefanopoulou, Modeling and estimation for advanced battery management, Annu. Rev. Control Robot. Auton. Syst. 2 (1) (2019) 393–426, http://dx.doi.org/10.1146/annurev-control-053018-023643.

[8] K.C. Ndeche, S.O. Ezeonu, Implementation of coulomb counting method for estimating the state of charge of lithium-ion battery, Phys. Sci. Int. J. (2021) 1–8, http://dx.doi.org/10.9734/psij/2021/v25i330244.

[9] I.B. Espedal, A. Jinasena, O.S. Burheim, J.J. Lamb, Current trends for state-of-charge (soc) estimation in lithium-ion battery electric vehicles, Energies 14 (11) (2021) 3284, http://dx.doi.org/10.3390/en14113284.

[10] W. Li, J. Zhang, F. Ringbeck, D. Jöst, L. Zhang, Z. Wei, D.U. Sauer, Physics-informed neural networks for electrode-level state estimation in lithium-ion batteries, J. Power Sources 506 (2021) 230034, http://dx.doi.org/10.1016/j.jpowsour.2021.230034.

[11] W. Li, N. Sengupta, P. Dechent, D. Howey, A. Annaswamy, D.U. Sauer, Online capacity estimation of lithium-ion batteries with deep long short-term memory networks, J. Power Sources 482 (2021) 228863, http://dx.doi.org/10.1016/j.jpowsour.2020.228863.

[12] Y. Li, K. Liu, A.M. Foley, A. Zülke, M. Berecibar, E. Nanini-Maury, J. van Mierlo, H.E. Hoster, Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review, Renew. Sustain. Energy Rev. 113 (2019) 109254, http://dx.doi.org/10.1016/j.rser.2019.109254.

[13] Y. Li, C. Zou, M. Berecibar, E. Nanini-Maury, J.C.-W. Chan, P. van den Bossche, J. van Mierlo, N. Omar, Random forest regression for online capacity estimation of lithium-ion batteries, Appl. Energy 232 (2018) 197–210, http://dx.doi.org/10.1016/j.apenergy.2018.09.182.

[14] W. Waag, C. Fleischer, D.U. Sauer, Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles, J. Power Sources 258 (2014) 321–339, http://dx.doi.org/10.1016/j.jpowsour.2014.02.064.

[15] A.M. Bizeray, J.-H. Kim, S.R. Duncan, D.A. Howey, Identifiability and parameter estimation of the single particle lithium-ion battery model, IEEE Trans. Control Syst. Technol. 27 (5) (2019) 1862–1877, http://dx.doi.org/10.1109/TCST.2018.2838097.

[16] U. Krewer, F. Röder, E. Harinath, R.D. Braatz, B. Bedürftig, R. Findeisen, Review—dynamic models of li-ion batteries for diagnosis and operation: A review and perspective, J. Electrochem. Soc. 165 (16) (2018) A3656–A3673, http://dx.doi.org/10.1149/2.1061814jes.

[17] B. Wu, W.D. Widanage, S. Yang, X. Liu, Battery digital twins: Perspectives on the fusion of models, data and artificial intelligence for smart battery management systems, Energy AI 1 (2020) 100016, http://dx.doi.org/10.1016/j.egyai.2020.100016.

[18] W. Lai, F. Ciucci, Mathematical modeling of porous battery electrodes—Revisit of newman's model, Electrochim. Acta 56 (11) (2011) 4369–4377, http://dx.doi.org/10.1016/j.electacta.2011.01.012.

[19] J. Moškon, J. Žuntar, S.D. Talian, R. Dominko, M. Gaberšček, A powerful transmission line model for analysis of impedance of insertion battery cells: A case study on the nmc-li system, J. Electrochem. Soc. 167 (14) (2020) 140539, http://dx.doi.org/10.1149/1945-7111/abc769.

[20] Z. Geng, S. Wang, M.J. Lacey, D. Brandell, T. Thiringer, Bridging physics-based and equivalent circuit models for lithium-ion batteries, Electrochim. Acta 372 (2021) 137829, http://dx.doi.org/10.1016/j.electacta.2021.137829.

[21] M. Doyle, T.F. Fuller, J. Newman, Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell, J. Electrochem. Soc. 140 (6) (1993) 1526–1533, http://dx.doi.org/10.1149/1.2221597.

[22] T.F. Fuller, M. Doyle, J. Newman, Simulation and optimization of the dual lithium ion insertion cell, J. Electrochem. Soc. 141 (1) (1994) 1–10, http://dx.doi.org/10.1149/1.2054684.

[23] M. Ecker, T.K.D. Tran, P. Dechent, S. Käbitz, A. Warnecke, D.U. Sauer, Parameterization of a physico-chemical model of a lithium-ion battery, J. Electrochem. Soc. 162 (9) (2015) A1836–A1848, http://dx.doi.org/10.1149/2.0551509jes.

[24] M. Ecker, S. Käbitz, I. Laresgoiti, D.U. Sauer, Parameterization of a physico-chemical model of a lithium-ion battery, J. Electrochem. Soc. 162 (9) (2015) A1849–A1857, http://dx.doi.org/10.1149/2.0541509jes.

[25] J. Schmalstieg, C. Rahe, M. Ecker, D.U. Sauer, Full cell parameterization of a high-power lithium-ion battery for a physico-chemical model: Part i. physical and electrochemical parameters, J. Electrochem. Soc. 165 (16) (2018) A3799–A3810, http://dx.doi.org/10.1149/2.0321816jes.

[26] J. Schmalstieg, D.U. Sauer, Full cell parameterization of a high-power lithium-ion battery for a physico-chemical model: Part ii. thermal parameters and validation, J. Electrochem. Soc. 165 (16) (2018) A3811–A3819, http://dx.doi.org/10.1149/2.0331816jes.

[27] J. Sturm, A. Rheinfeld, I. Zilberman, F.B. Spingler, S. Kosch, F. Frie, A. Jossen, Modeling and simulation of inhomogeneities in a 18650 nickel-rich, silicon-graphite lithium-ion cell during fast charging, J. Power Sources 412 (2019) 204–223, http://dx.doi.org/10.1016/j.jpowsour.2018.11.043.

[28] W. Li, D. Cao, D. Jöst, F. Ringbeck, M. Kuipers, F. Frie, D.U. Sauer, Parameter sensitivity analysis of electrochemical model-based battery management systems for lithium-ion batteries, Appl. Energy 269 (2020) 115104, http://dx.doi.org/10.1016/j.apenergy.2020.115104.

[29] A. Pozzi, G. Ciaramella, S. Volkwein, D.M. Raimondo, Optimal design of experiments for a lithium-ion cell: Parameters identification of an isothermal single particle model with electrolyte dynamics, Ind. Eng. Chem. Res. 58 (3) (2019) 1286–1299, http://dx.doi.org/10.1021/acs.iecr.8b04580.

[30] K. Gopalakrishnan, G.J. Offer, A composite single particle lithium-ion battery model through system identification, IEEE Trans. Control Syst. Technol. (2021) 1–13, http://dx.doi.org/10.1109/TCST.2020.3047776.

[31] S. Atlung, K. West, T. Jacobsen, Dynamic aspects of solid solution cathodes for electrochemical power sources, J. Electrochem. Soc. 126 (8) (1979) 1311–1321, http://dx.doi.org/10.1149/1.2129269.

[32] G. Ning, B.N. Popov, Cycle life modeling of lithium-ion batteries, J. Electrochem. Soc. 151 (10) (2004) A1584, http://dx.doi.org/10.1149/1.1787631.

[33] A. Jokar, B. Rajabloo, M. Désilets, M. Lacroix, Review of simplified pseudo-two-dimensional models of lithium-ion batteries, J. Power Sources 327 (2016) 44–55, http://dx.doi.org/10.1016/j.jpowsour.2016.07.036.

[34] V.R. Subramanian, V.D. Diwakar, D. Tapriyal, Efficient macro-micro scale coupled modeling of batteries, J. Electrochem. Soc. 152 (10) (2005) A2002–A2008, http://dx.doi.org/10.1149/1.2032427.

[35] V.R. Subramanian, V. Boovaragavan, V.D. Diwakar, Toward real-time simulation of physics based lithium-ion battery models, Electrochem. Solid-State Lett. 10 (11) (2007) A255, http://dx.doi.org/10.1149/1.2776128.

[36] Z. Khalik, M. Donkers, H.J. Bergveld, Model simplifications and their impact on computational complexity for an electrochemistry-based battery modeling toolbox, J. Power Sources 488 (2021) 229427, http://dx.doi.org/10.1016/j.jpowsour.2020.229427.

[37] J.C. Forman, S. Bashash, J.L. Stein, H.K. Fathy, Reduction of an electrochemistry-based li-ion battery model via quasi-linearization and padé approximation, J. Electrochem. Soc. 158 (2) (2011) A93, http://dx.doi.org/10.1149/1.3519059.

[38] S. Marelli, M. Corno, Model-based estimation of lithium concentrations and temperature in batteries using soft-constrained dual unscented kalman filtering, IEEE Trans. Control Syst. Technol. 29 (2) (2021) 926–933, http://dx.doi.org/10.1109/TCST.2020.2974176.

[39] T.-S. Dao, C.P. Vyasarayani, J. McPhee, Simplification and order reduction of lithium-ion battery model based on porous-electrode theory, J. Power Sources 198 (2012) 329–337, http://dx.doi.org/10.1016/j.jpowsour.2011.09.034.

[40] G. Fan, X. Li, M. Canova, A reduced-order electrochemical model of li-ion batteries for control and estimation applications, IEEE Trans. Veh. Technol. 67 (1) (2018) 76–91, http://dx.doi.org/10.1109/TVT.2017.2738780.

[41] S. Mazumder, J. Lu, Faster-than-real-time simulation of lithium ion batteries with full spatial and temporal resolution, Int. J. Electrochem. 2013 (2013) http://dx.doi.org/10.1155/2013/268747.

[42] L. Xia, E. Najafi, Z. Li, H.J. Bergveld, M. Donkers, A computationally efficient implementation of a full and reduced-order electrochemistry-based model for li-ion batteries, Appl. Energy 208 (2017) 1285–1296, http://dx.doi.org/10.1016/j.apenergy.2017.09.025.

[43] S. Han, Y. Tang, S.K. Rahimian, A numerically efficient method of solving the full-order pseudo-2-dimensional (P2D) li-ion cell model, J. Power Sources 490 (2021) 229571, http://dx.doi.org/10.1016/j.jpowsour.2021.229571.

[44] J. Sturm, S. Ludwig, J. Zwirner, C. Ramirez-Garcia, B. Heinrich, M.F. Horsche, A. Jossen, Suitability of physicochemical models for embedded systems regarding a nickel-rich, silicon-graphite lithium-ion battery, J. Power Sources 436 (2019) 226834, http://dx.doi.org/10.1016/j.jpowsour.2019.226834.

[45] V. Senthil Kumar, Reduced order model for a lithium ion cell with uniform reaction rate approximation, J. Power Sources 222 (2013) 426–441, http://dx.doi.org/10.1016/j.jpowsour.2012.09.013.

[46] M.K.S. Verma, S. Basu, R.S. Patil, K.S. Hariharan, S.P. Adiga, S.M. Kolake, D. Oh, T. Song, Y. Sung, On-board state estimation in electrical vehicles: Achieving accuracy and computational efficiency through an electrochemical model, IEEE Trans. Veh. Technol. 69 (3) (2020) 2563–2575, http://dx.doi.org/10.1109/TVT.2020.2966266.

[47] C. Campestrini, T. Heil, S. Kosch, A. Jossen, A comparative study and review of different kalman filters by applying an enhanced validation method, J. Energy Storage 8 (2016) 142–159, http://dx.doi.org/10.1016/j.est.2016.10.004.

[48] X. Lai, C. Qin, W. Gao, Y. Zheng, W. Yi, A state of charge estimator based extended Kalman filter using an electrochemistry-based equivalent circuit model for lithium-ion batteries, Appl. Sci. 8 (9) (2018) 1592, http://dx.doi.org/10.3390/app8091592.

[49] P. Shrivastava, T.K. Soon, M.Y.I.B. Idris, S. Mekhilef, Overview of model-based online state-of-charge estimation using Kalman filter family for lithium-ion batteries, Renew. Sustain. Energy Rev. 113 (2019) 109233, http://dx.doi.org/10.1016/j.rser.2019.06.040.

[50] S. Yang, S. Zhou, Y. Hua, X. Zhou, X. Liu, Y. Pan, H. Ling, B. Wu, A parameter adaptive method for state of charge estimation of lithium-ion batteries with an improved extended Kalman filter, Sci. Rep. 11 (1) (2021) 5805, http://dx.doi.org/10.1038/s41598-021-84729-1.

[51] A.M. Bizeray, S. Zhao, S.R. Duncan, D.A. Howey, Lithium-ion battery thermal-electrochemical model-based state estimation using orthogonal collocation and a modified extended kalman filter, J. Power Sources 296 (2015) 400–412, http://dx.doi.org/10.1016/j.jpowsour.2015.07.019.

[52] K.D. Stetzel, L.L. Aldrich, M.S. Trimboli, G.L. Plett, Electrochemical state and internal variables estimation using a reduced-order physics-based model of a lithium-ion cell and an extended kalman filter, J. Power Sources 278 (2015) 490–505, http://dx.doi.org/10.1016/j.jpowsour.2014.11.135.

[53] J. Sturm, H. Ennifar, S.V. Erhard, A. Rheinfeld, S. Kosch, A. Jossen, State estimation of lithium-ion cells using a physicochemical model based extended kalman filter, Appl. Energy 223 (2018) 103–123, http://dx.doi.org/10.1016/j.apenergy.2018.04.011.

[54] W. Li, Y. Fan, F. Ringbeck, D. Jöst, X. Han, M. Ouyang, D.U. Sauer, Electrochemical model-based state estimation for lithium-ion batteries with adaptive unscented kalman filter, J. Power Sources 476 (2020) 228534, http://dx.doi.org/10.1016/j.jpowsour.2020.228534.

[55] S.S. Haykin, Kalman Filtering and Neural Networks, in: Adaptive and Learning Systems for Signal Processing, Communications, and Control, John Wiley & Sons, Inc., New York, 2001.

[56] Y. Lin, L. Bao, X. Jia, Convergence analysis of a variant of the newton method for solving nonlinear equations, Comput. Math. Appl. 59 (6) (2010) 2121–2127, http://dx.doi.org/10.1016/j.camwa.2009.12.017.

[57] J. Mao, W. Tiedemann, J. Newman, Simulation of temperature rise in li-ion cells at very high currents, J. Power Sources 271 (2014) 444–454, http://dx.doi.org/10.1016/j.jpowsour.2014.08.033.

[58] K. Kumaresan, G. Sikha, R.E. White, Thermal model for a li-ion cell, J. Electrochem. Soc. 155 (2) (2008) A164, http://dx.doi.org/10.1149/1.2817888.

[59] T.G. Tranter, R. Timms, P.R. Shearing, D.J.L. Brett, Communication—Prediction of thermal issues for larger format 4680 cylindrical cells and their mitigation with enhanced current collection, J. Electrochem. Soc. 167 (16) (2020) 160544, http://dx.doi.org/10.1149/1945-7111/abd44f.

[60] V.A. Bavdekar, A.P. Deshpande, S.C. Patwardhan, Identification of process and measurement noise covariance for state and parameter estimation using extended kalman filter, J. Process Control 21 (4) (2011) 585–601, http://dx.doi.org/10.1016/j.jprocont.2011.01.001.

[61] Infineon Technologies AG, 32-Bit TC38x Single-chip microcontroller: data sheet, 2021, URL https://www.infineon.com/dgdl/Infineon-TC38x-DataSheet-v01_10-EN.pdf?fileId=5546d4626f229553016fb316e6cf748b.

[62] IEEE standard for floating-point arithmetic. http://dx.doi.org/10.1109/IEEESTD.2008.4610935.

[63] S. Attaway, MATLAB: A Practical Introduction to Programming and Problem Solving, Butterworth-Heinemann, Oxford, 2009.

[64] TASKING BV, Tasking vx-toolset for tricore user guide, 2019, URL https://www.tasking.com/support/tricore/ctc_user_guide_v6.3r1.pdf.

[65] J. Thorson, Gaussian elimination on a banded matrix, 1979, URL http://sep.stanford.edu/data/media/public/oldreports/sep20/20_11.pdf.

[66] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, second ed., in: Texts in Applied Mathematics, vol. 37, Springer Science+Business Media S.A., New York, 2007, http://dx.doi.org/10.1007/b98885.

[67] Infineon Technologies AG, TriCore 32-bit unified processor core v1.6 instruction set: User manual, 2012, URL https://www.infineon.com/dgdl/tc1_6_architecture_vol2.pdf?fileId=db3a3043372d5cc801374ad9c0653ad9.

[68] Infineon Technologies AG, TLE9012AQU - li-ion battery monitoring and balancing IC: Data sheet, 2020, URL https://www.infineon.com/dgdl/Infineon-TLE9012AQU-DataSheet-v01_10-EN.pdf?fileId=5546d46272e49d2a017348151bd40cd5.

[69] C. von Lüders, J. Keil, M. Webersberger, A. Jossen, Modeling of lithium plating and lithium stripping in lithium-ion batteries, J. Power Sources 414 (2019) 41–47, http://dx.doi.org/10.1016/j.jpowsour.2018.12.084.

[70] A. Tomaszewska, Z. Chu, X. Feng, S. O'Kane, X. Liu, J. Chen, C. Ji, E. Endler, R. Li, L. Liu, Y. Li, S. Zheng, S. Vetterlein, M. Gao, J. Du, M. Parkes, M. Ouyang, M. Marinescu, G. Offer, B. Wu, Lithium-ion battery fast charging: A review, ETransportation 1 (2019) 100011, http://dx.doi.org/10.1016/j.etran.2019.100011.

[71] X.-L. Gao, X.-H. Liu, W.-L. Xie, L.-S. Zhang, S.-C. Yang, Multiscale observation of li plating for lithium-ion batteries, Rare Metals 40 (11) (2021) 3038–3048, http://dx.doi.org/10.1007/s12598-021-01730-3.

[72] H.-J. Noh, S. Youn, C.S. Yoon, Y.-K. Sun, Comparison of the structural and electrochemical properties of layered li[$\mathrm{ni}_x\mathrm{co}_y\mathrm{mn}_z$]$\mathrm{o}_2$ (x=1/3, 0.5, 0.6, 0.7, 0.8 and 0.85) cathode material for lithium-ion batteries, J. Power Sources 233 (2013) 121–130, http://dx.doi.org/10.1016/j.jpowsour.2013.01.063.

[73] L.O. Valøen, J.N. Reimers, Transport properties of lipf_6-based li-ion battery electrolytes, J. Electrochem. Soc. 152 (5) (2005) A882, http://dx.doi.org/10.1149/1.1872737.