Journal of Power Sources 606 (2024) 234547

Contents lists available at ScienceDirect

Journal of Power Sources

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

Multi-interest adaptive unscented Kalman filter based on improved matrix decomposition methods for lithium-ion battery state of charge estimation

Sijing Wang [a,1], Pan Huang [a,1], Cheng Lian [a,b,*], Honglai Liu [a,b]

[a] State Key Laboratory of Chemical Engineering, Shanghai Engineering Research Center of Hierarchical Nanomaterials, School of Chemical Engineering, East China University of Science and Technology, Shanghai, 200237, China
[b] School of Chemistry and Molecular Engineering, East China University of Science and Technology, Shanghai, 200237, China

HIGHLIGHTS

- For any battery-powered device, correct SOC estimation is critical.
- A battery state-space model was developed for state of charge (SOC) estimation.
- Model parameter identification using bias-compensated FFRLS approach.
- Optimization of traditional UKF to improve the stability of the algorithm.
- The algorithm was validated using battery data from various discharge conditions.

ARTICLE INFO

Keywords:
State of charge
Bias-compensated FFRLS
Optimization MIAUKF
Stability
Robustness

ABSTRACT

Accurately predicting the state of charge (SOC) is crucial to improving Li-ion battery performance. However, available model-based estimation approaches still face challenges in handling model uncertainty and measurement noise effects on parameter identification. Besides, the widely used unscented Kalman filter (UKF) algorithm has limitations in electric vehicles as it requires the error covariance matrix to maintain positive definiteness, limiting its applicability under certain conditions. This study introduces the bias-compensated forgetting factor recursive least squares (BCFFRLS) method for parameter estimation within the second-order *RC* equivalent circuit model specific to the INR18650-20R battery. Furthermore, we propose a novel algorithm named the optimization multi-interest adaptive unscented Kalman filter (O-MIAUKF). This algorithm is designed to address stability and robustness issues with traditional UKF encounters in dynamic environments. Experimental validation demonstrates that the O-MIAUKF algorithm excels in maintaining strong stability and robustness in various working conditions, accurately estimating SOC even with a non-positive covariance matrix. The SOC estimation error remains stable at 0.8 %, which is lower than that of the current Extended Kalman Filter (EKF), UKF, and Dual Extended Kalman Filter (DEKF).

1. Introduction

Li-ion batteries serve as the main power storage units in electric vehicles (EVs) as a result of their substantial energy density and extended service life [1]. For the safe and valid functioning of Li-ion batteries in EVs, it is imperative to have a robust and reliable Battery Management System (BMS) [2,3]. A fundamental function of the BMS is the accurate estimation of the state of charge (SOC) [4]. During EV driving, inaccurate SOC estimates can lead to the driver being unable to predict range. Additionally, this can result in both overcharging and over-discharging of the battery, thereby elevating the risk of permanent damage to its internal composition [5,6].

At present, three principal approaches have been suggested for SOC estimation: the experimental method [7–9], the data-driven technique [10–12], and the model-based approach [13,14]. Among these, the model-based approaches strike a more effective balance with regard to

* Corresponding author. State Key Laboratory of Chemical Engineering, Shanghai Engineering Research Center of Hierarchical Nanomaterials, School of Chemical Engineering, East China University of Science and Technology, Shanghai, 200237, China.
E-mail address: liancheng@ecust.edu.cn (C. Lian).
[1] Equally contributing authors.

https://doi.org/10.1016/j.jpowsour.2024.234547
Received 3 January 2024; Received in revised form 28 March 2024; Accepted 12 April 2024
Available online 25 April 2024
0378-7753/

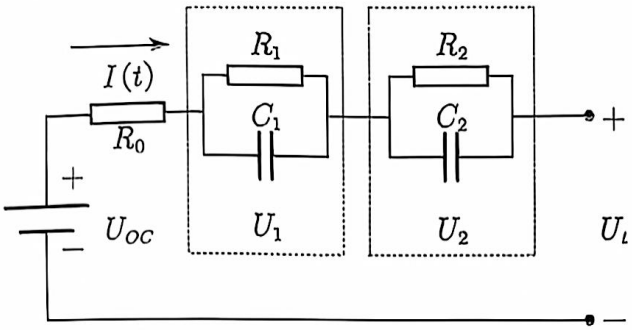

Fig. 1. The second-order *RC* model.

accuracy, complexity, real-time performance, and computational expenses, making it a promising option for EV applications [15]. Model-based approaches usually consist of two stages: modeling of the battery and implementation of algorithms. In the stage of battery modeling [16], developing a mathematical model is crucial to accurately represent the battery's behavior. Typically, this model is divided into four main types: the empirical model [17], the electrochemical model [18], the equivalent circuit model (ECM) [19], and the electrochemical impedance model [20]. In contrast, ECM stands out as the most apt for SOC estimation, owing to its wide-ranging applicability and relatively low computational requirements [21]. Once the model is established, identifying the model parameters becomes crucial to determine SOC values [22]. An offline parameter identification method is used in conjunction with Hybrid Pulse Power Characterization (HPPC) experiments to determine these parameters. Nevertheless, this approach fails to precisely represent the battery's dynamic response in the presence of intricate operating conditions [23]. This issue can be resolved through the application of online parameter identification techniques. The recursive least squares (RLS) method [24] is a common choice, but it can suffer from data saturation issues that lead to inaccuracies in parameter identification. To solve these problems, Zhang et al. [25] have introduced the RLS method incorporating a forgetting factor. Additionally, Chen et al. [26] have proposed the bias-compensated recursive least squares method to enhance parameter identification efficacy by addressing the impact of model noise on estimation outcomes. In the phase of algorithm implementation, algorithms from the Kalman Filter (KF) family [27] are extensively used for SOC estimation, owing to their capability to effectively combine model predictions and actual measurements to deal with noise and uncertainty. However, given the highly nonlinear nature of battery models [28], the current SOC estimation mainly employs the following four algorithms: Extended Kalman Filter (EKF) [29], Unscented Kalman Filter (UKF) [30], Cubature Kalman Filter (CKF) [31], and Particle Filter (PF) [32]. Each of these algorithms can deal with nonlinear mathematical models. Moreover, the constant covariance of measurement and process noise within the system can diminish the estimation's overall performance [33]. An adaptive link [34,35] has been introduced to the KF family algorithm to elevate the precision of SOC estimation, which enables the online correction of measurement and process noise covariance.

Many researchers have found that there are still many limitations in the online methods for estimating SOC. For instance, the model may not fully capture the battery behavior in parameter identification, and the processing capabilities may be insufficient for data under extreme conditions [36,37]. In addition, to ensure that the UKF works appropriately, the error covariance matrix remains positively definite [38]. Nevertheless, as the algorithm progresses, in successive updates, the error covariance matrix risks evolving into a matrix that is not positive definite, which can lead to an abnormal stopping of the algorithm. Currently, Wang et al. [39–41] use the UKF algorithm, this approach not only initializes the error covariance matrix as a positively definite diagonal matrix but also does not compute the sigma points for the priori probability distribution of the state variable at the kth moment. This avoids the process of updating the system's error covariance matrix, enhancing algorithm stability, but its accuracy is also substantially reduced. Consequently, achieving stable and accurate SOC estimation remains challenging and highly focused in the field [42]. Therefore, it is necessary to find ways to overcome the above limitations, thereby enhancing the algorithm's accuracy and stability.

In our research, a combined approach of online parameter identification and state estimation is introduced as a solution to these two issues. We use the bias-compensated forgetting factor recursive least squares (BCFFRLS) approach for real-time identification of model parameters to mitigate the impact of data saturation and uncertainty noise. To enhance the stability of the traditional UKF, three distinct matrix decomposition methods, Singular Value Decomposition, Eigenvalue Decomposition, and QR Decomposition, replace the Cholesky Decomposition in the original algorithm. SOC estimation is additionally performed under HPPC conditions, where we compare the stability and operational speed of the optimized UKF algorithms using these three methods. Then, we selected the more rapid Singular Value Decomposition method, building upon this, the multi-interest theory and the adaptive link were introduced to refine SOC estimation accuracy. The efficacy of the SVD-MIAUKF algorithm was then validated under dynamic testing conditions, with its estimation outcomes being analyzed and contrasted against those from various algorithms like EKF, UKF, and DEKF. Furthermore, the robustness of the presented joint estimation approach was evaluated with the use of inaccurate initial values.

2. Battery model and parameters identification

2.1. Second-order RC model

The common ECMs are the Rint model [43], the PNGV model [44], the n-order *RC* model [45], and GNL model [46]. This study employs the second-order *RC* model to simulate the dynamically varying battery properties, as illustrated in Fig. 1.

This model depicts the battery's internal resistance characteristics using ohmic internal resistance, and the two *RC* loops are utilized to represent the concentration differential polarization and the electrochemical polarization within the battery, each loop corresponding to one of these aspects respectively. Compared to other ECMs, the second-order *RC* model boasts superior simulation precision and simpler parameter identification. The functional equations of the circuit can be obtained by applying the law of Kirchhoff:

$$\begin{cases} U_t = U_{OC} - I(t)R_0 - U_1 - U_2 \\ \dfrac{dU_1}{dt} = \dfrac{I(t)}{C_1} - \dfrac{U_1}{R_1C_1} \\ \dfrac{dU_2}{dt} = \dfrac{I(t)}{C_2} - \dfrac{U_2}{R_2C_2} \end{cases} \tag{1}$$

Where C_1 and C_2 are the polarization capacitance, R_1 and R_2 are used to denote the battery's polarization resistance, U_{OC} represents the open-circuit voltage, U_1 and U_2 are the voltages across two respective *RC* loops, U_t signifies the terminal voltage, $I(t)$ denotes the battery's charging or discharging current during its operation, and R_0 is the inherent ohmic resistance. U_{OC} can be determined by the *SOC* [47,48], U_t and $I(t)$ can be measured by a sensor.

Following discretization within the time domain, the state space formula at the discrete time step k is given by:

$$\begin{cases} U_{t,k} = U_{OC}(SOC_k) - I_{t,k-1}R_0 - U_{1,k} - U_{2,k} \\ U_{1,k} = U_{1,k-1}\exp\left(-\dfrac{T_s}{\tau_1}\right) + I_{t,k-1}R_1\left(1 - \exp\left(-\dfrac{T_s}{\tau_1}\right)\right) \\ U_{2,k} = U_{2,k-1}\exp\left(-\dfrac{T_s}{\tau_2}\right) + I_{t,k-1}R_2\left(1 - \exp\left(-\dfrac{T_s}{\tau_2}\right)\right) \end{cases} \tag{2}$$

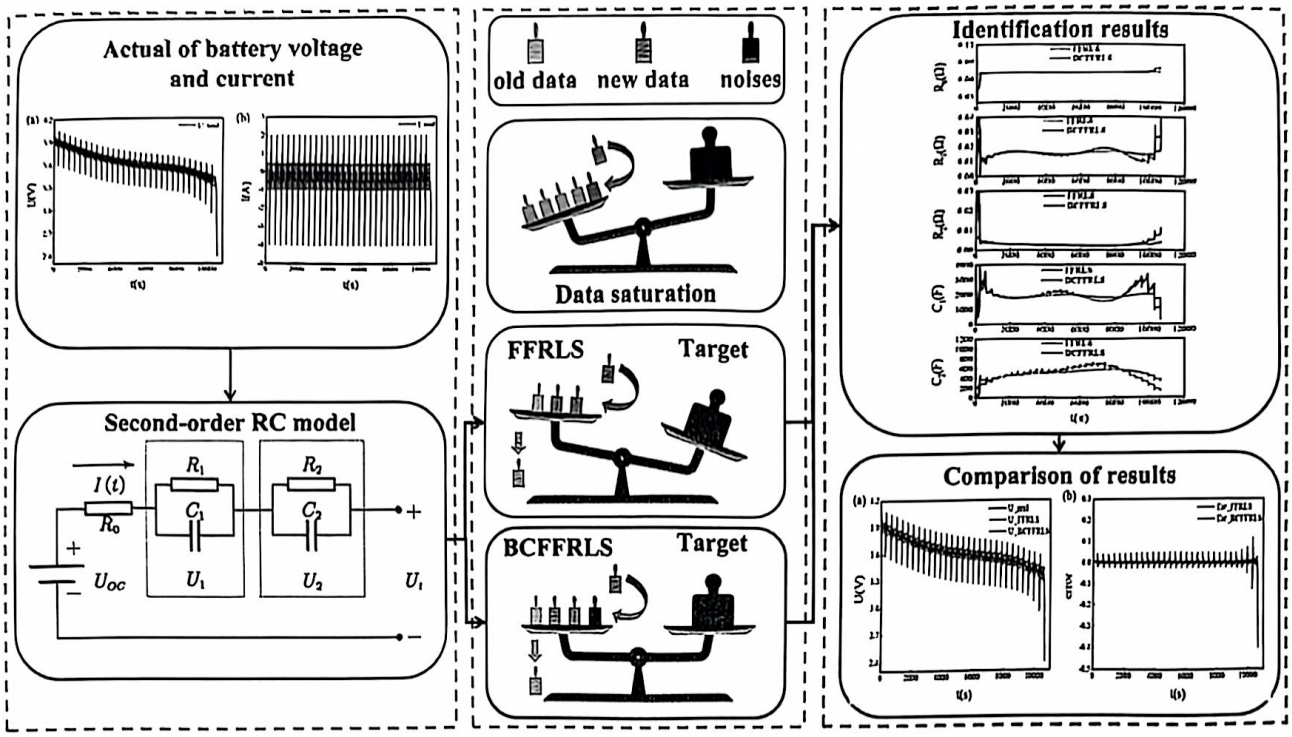

Fig. 2. Diagram of the BCFFRLS method.

Where T_s represents the interval of sampling time, τ_1 and τ_2 are the respective time constants, defined as $\tau_1 = R_1C_1, \tau_2 = R_2C_2$.

2.2. Improved method for online parameter identification

Accurate and real-time identification and adjustment of model parameters are vital for improving SOC estimation accuracy. Our research focuses on the widely adopted second-order RC equivalent model, utilizing the BCFFRLS approach for parameter identification. The underlying concept is illustrated in Fig. 2.

Fig. 2 provides a comprehensive view of the BCFFRLS method, outlining its overall procedure. Moving forward, we delve into the detailed workings of the BCFFRLS method. Starting from Eq. (1), the state equation is formulated in the frequency domain by utilizing the Laplace transform, as shown in Eq. (3):

$$U_{OC}(s) - U_t(s) = I(s)\left(R_0 + \frac{R_1}{R_1C_1s+1} + \frac{R_2}{R_2C_2s+1}\right) \tag{3}$$

Where is known that $\tau_1 = R_1C_1$ and $\tau_2 = R_2C_2$. By setting $U(s) = U_{OC}(s) - U_t(s)$, we can deduce the battery's transfer function and present it in Eq. (4):

$$G(s) = \frac{U(s)}{I(s)} = \frac{R_0 + R_1 + R_2 + (R_0\tau_1 + R_0\tau_2 + R_2\tau_1 + R_1\tau_2)s + R_0\tau_1\tau_2s^2}{1 + (\tau_1 + \tau_2)s + \tau_1\tau_2s^2} \tag{4}$$

Let:

$$\begin{cases} a = R_0 \\ b = \tau_1\tau_2 \\ c = \tau_1 + \tau_2 \\ d = R_0 + R_1 + R_2 \\ e = R_0\tau_1 + R_0\tau_2 + R_2\tau_1 + R_1\tau_2 \end{cases} \tag{5}$$

Can be obtained:

$$G(s) = \frac{U(s)}{I(s)} = \frac{d + es + abs^2}{1 + cs + bs^2} \tag{6}$$

To maintain consistency in the system before and after discretization, it is necessary to employ a bilinear transformation during the discretization process, let $s = \frac{2(1-Z^{-1})}{T(1+Z^{-1})}$. The discretized transfer function can be derived:

$$G(Z^{-1}) = \frac{(dT^2 - 2Te + 4ab)Z^{-2} + (2dT^2 - 8ab)Z^{-1} + dT^2 + 2Te + 4ab}{(T^2 - 2Tc + 4b)Z^{-2} + (2T^2 - 8b)Z^{-1} + (T^2 + 2Tc + 4b)} \tag{7}$$

To simplify the calculation, let:

$$\begin{cases} k_1 = -\dfrac{2T^2 - 8b}{T^2 + 2Tc + 4b} \\ k_2 = -\dfrac{T^2 - 2Tc + 4b}{T^2 + 2Tc + 4b} \\ k_3 = \dfrac{dT^2 + 2Te + 4ab}{T^2 + 2Tc + 4b} \\ k_4 = \dfrac{2dT^2 - 8ab}{T^2 + 2Tc + 4b} \\ k_5 = \dfrac{dT^2 - 2Te + 4ab}{T^2 + 2Tc + 4b} \end{cases} \tag{8}$$

Is derived as:

$$G(Z^{-1}) = \frac{U(Z^{-1})}{I(Z^{-1})} = \frac{k_3Z^{-2} + k_4Z^{-1} + k_3}{1 - k_2Z^{-2} - k_1Z^{-1}} \tag{9}$$

Where the constants from k_1 to k_5 serve as the corresponding coefficient values, transforming Eq. (9) into a differential equation for the system:

$$U(k) = k_1U(k-1) + k_2U(k-2) + k_3I(k) + k_4I(k-1) + k_5I(k-2) \tag{10}$$

Where $I(k)$ represents the input of the system, and $U(k)$ denotes the

output of the system, let:

$$\begin{cases} \varphi(k) = [U(k-1), U(k-2), I(k), I(k-1), I(k-2)]^T \\ \theta = [k_1, k_2, k_3, k_4, k_5]^T \end{cases} \tag{11}$$

Define the sensor sampling error at the k instant as $e(k)$, this occasion:

$$U(k) = \varphi^T(k)\theta + e(k) \tag{12}$$

Expanding $\varphi(k)$ to N dimensions, where k takes values of 3, 4, ... up to $N+2$, the following equation can be obtained:

$$\begin{cases} U = [U(k)\ U(k-1)\ U(k-2) \cdots U(k-N)]^T \\ e = [e(k)\ e(k-1)\ e(k-2) \cdots e(k-N)]^T \\ \varphi(k) = \begin{bmatrix} U(k-1)\ U(k-2)\ I(k)\ I(k-1)\ I(k-2) \\ U(k-2)\ U(k-3)\ I(k-1)\ I(k-2)\ I(k-3) \\ \vdots\ \vdots\ \vdots\ \vdots\ \vdots \\ U(k-N-1)\ U(k-N-2)\ I(k-N)\ I(k-N-1)\ I(k-N-2) \end{bmatrix}^T \end{cases} \tag{13}$$

Taking the generalized function $J(\theta)$:

$$J(\theta) = \sum_{i=1}^{N} (U - \varphi\theta)^2 = \sum_{i=1}^{N} (e(i+2))^2 \tag{14}$$

Given that the objective of the least squares method is to minimize $J(\theta)$, to locate the extremum of $J(\theta)$, we set:

$$\frac{\partial J(\theta)}{\partial(\theta)} = \frac{\partial}{\partial(\theta)} [(U - \varphi\theta)^T (U - \varphi\theta)] = 0 \tag{15}$$

Can be obtained:

$$\widehat{\theta} = [\varphi^T \varphi]^{-1} \varphi^T Y \tag{16}$$

A recursive operation on the process above is the RLS approach, as shown in Eq. (17):

$$\begin{cases} K(k+1) = P(k)\varphi(k+1) \times (1 + \varphi^T(k+1)P(k)\varphi(k+1))^{-1} \\ e(k+1) = U(k+1) - \varphi^T(k+1)\widehat{\theta}(k) \\ \widehat{\theta}(k+1) = \widehat{\theta}(k) + K(k+1)e(k+1) \\ P(k+1) = P(k) - K(k+1)\varphi^T(k+1)P(k) \end{cases} \tag{17}$$

where $\widehat{\theta}(k)$ is the previously estimated reference value of the system, $\varphi^T(k+1)\widehat{\theta}(k)$ represents the observation present magnitude, $U(k+1)$ signifies the system's actual observed value, and $e(k+1)$ denotes the prediction error. Multiplying the prediction error by the $K(k+1)$ gain term serves as a correction to the current prediction, resulting in the optimal estimate $\widehat{\theta}(k+1)$ at this point in time. The eligible $\widehat{\theta}(0)$ and $P(0)$ must be provided to obtain the gain term $K(k+1)$, which in turn initiates the least squares method, typically, $\widehat{\theta}(0)$ can assume any value, $P_0 = \alpha E$, with α being as large as feasible, and E functioning as a unit array.

The RLS algorithm features an infinite memory length, for the battery system, the accumulation of older data during recursive computations can hinder the ability of the algorithm to represent the characteristics of new data. To prevent the aforementioned situation, the introduction of the forgetting factor λ, thus:

$$\frac{1}{P(k+1)} = \frac{\lambda}{P(k)} + \varphi(k+1)\varphi^T(k+1) \tag{18}$$

Hence, even when $(k+1)$ is substantially large, $P(k+1)$ does not approach zero, effectively addressing the issue of data saturation. When $\lambda = 1$, the method is a standard least squares approach [49].

Yet, in the practical operation of EVs, acquiring current and voltage data from Li-ion batteries frequently encounters unpredictable noise, which can result in inaccuracies in the identification outcomes. Therefore, we use the bias compensation approach to better fit the actual data, particularly in the case of model uncertainty or measurement error. The sequential steps of the finalized algorithm are outlined as follows:

$$\begin{cases} K(k+1) = P(k)\varphi(k+1) \times (\lambda + \varphi^T(k+1)P(k)\varphi(k+1))^{-1} \\ e(k+1) = U(k+1) - \varphi^T(k+1)\widehat{\theta}(k) \\ \widehat{\theta}(k+1) = \widehat{\theta}(k) + K(k+1)e(k+1) \\ P(k+1) = \lambda^{-1}[P(k) - P(k)K(k+1)\varphi^T(k+1)] \\ J(k+1) = J(k) + [e(k+1)]^2 \times (\lambda + \varphi^T(k+1)P(k)\varphi(k+1))^{-1} \\ \widehat{\sigma}^2(k+1) = J(k+1) \times (k[1 + \widehat{\theta}_{BC}^T(k)D\widehat{\theta}(k)])^{-1} \\ D = \begin{bmatrix} I_{n_a} & 0 \\ 0 & 0_{n_b} \end{bmatrix} \\ \widehat{\theta}_{BC}(k+1) = \widehat{\theta}(k+1) + k\widehat{\sigma}^2(k+1)P(k+1)D\widehat{\theta}_{BC}(k) \end{cases} \tag{19}$$

where $J(k+1)$ is the standard value for the system's error, $\widehat{\sigma}^2(k+1)$ is the current noise variance of the system, D is the matrix of correlation coefficients, I is the second-order identity matrix, $\widehat{\theta}_{BC}(k+1)$ is the current estimate of the parameter's value after the compensation for deviation, and in general, $\widehat{\theta}_{BC}(0)$ and $\widehat{\theta}(0)$ can be an arbitrary value.

Following the determination of $\widehat{\theta}_{BC}$ value using the algorithm above, the values from a to e can be deduced in accordance with Eq. (8):

$$\begin{cases} a = \dfrac{k_3 + k_5 - k_4}{1 + k_1 - k_2} \\ b = \dfrac{T^2(1 + k_1 - k_2)}{4(1 - k_1 - k_2)} \\ c = \dfrac{T(1 + k_2)}{1 - k_1 - k_2} \\ d = \dfrac{k_3 + k_4 + k_5}{1 - k_1 - k_2} \\ e = \dfrac{T(k_3 - k_5)}{1 - k_1 - k_2} \end{cases} \tag{20}$$

The recursive algorithm can derive the coefficients on the right-hand side of this equation, while the left side representing the model's unknown parameter, thus completing the parameter identification derivation process. According to Eq. (5) and Eq. (20), the values of the five parameters can be ascertained.

$$\begin{cases} R_0 = a \\ R_1 = \dfrac{\tau_1(d-a) + ac - e}{\tau_1 - \tau_2} \\ R_2 = d - a - R_1 \\ C_1 = \dfrac{\tau_1}{R_1} \\ C_2 = \dfrac{\tau_2}{R_2} \\ \tau_1, \tau_2 = \dfrac{c \pm \sqrt{c^2 - 4b}}{2} \end{cases} \tag{21}$$

3. SOC estimation algorithm

3.1. SOC definition

In the case of Li-ion batteries, the SOC is defined as the proportion of the residual capacity relative to the battery's total actual capacity, and the SOC at any given time t is obtained from the coulomb counting approach:

$$SOC(t) = SOC(t_0) - \frac{\int_{t_0}^{t} \eta I(t)dt}{C_N} \tag{22}$$

Where η represents the Coulombic efficiency, typically indicating the proportion of electrons that are available versus those consumed during the charge and discharge processes, and is assumed to be 0.9 during charging and 1 during discharging [50], and in this study, we assume η to be 1, C_N denotes the actual useable capacity of the battery. Discretization of Eq. (22) yields a discrete formulation of the *SOC* function:

$$SOC_t = SOC_{t-1} - \frac{T_s I_{k-1}}{C_N} \tag{23}$$

3.2. UKF algorithm

Typically, the nonlinear discrete state space equations for a Li-ion battery circuit model are as follows:

$$\begin{cases} x_t = Ax_{t-1} + Bu_{t-1} + w_{t-1} \\ y_t = U_{OC}(SOC_t) + Cx_t + Du_t + v_t \end{cases} \tag{24}$$

Where x and u are the state variables and inputs of the system, respectively. A and B represent the dynamic properties of the state equations, y is system output, C and D primarily depict the dynamic properties of the observation equations, w and v represent the system's process noise and measurement noise, respectively. In practical applications, these represent independent Gaussian white noises, each having a mean of zero, and k is the moment of iteration.

The procedural steps for implementing the UKF are as outlined:

1) Ascertain the initial value:

$$\begin{cases} \widehat{x_0^+} = E(x_0) \\ P_0^+ = E\left[(x_0 - \widehat{x_0^+})(x_0 - \widehat{x_0^+})^T\right] \end{cases} \tag{25}$$

2) Compute the sigma points for the state variable x_{k-1} at time k-1:

$$\begin{cases} x_{k-1}^0 = \widehat{x_{k-1}^+} \\ x_{k-1}^i = \widehat{x_{k-1}^+} + \sqrt{(L+\lambda)P_{k-1}^+}, i = 1,2...L \\ x_{k-1}^i = \widehat{x_{k-1}^+} - \sqrt{(L+\lambda)P_{k-1}^+}, i = L+1, L+2, ...2L \end{cases} \tag{26}$$

Where L is typically set to 3, representing the state vector's dimension, the weights are computed as follows:

$$\begin{cases} \lambda = \alpha^2(L+k_l) - L \\ W_m^0 = \frac{\lambda}{L+\lambda}, W_m^i = \frac{1}{2(L+\lambda)}, i = 1,2....2L \\ W_c^0 = \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta, W_c^i = \frac{1}{2(L+\lambda)}, i = 1,2....2L \end{cases} \tag{27}$$

Where k_l is the secondary scaling factor, commonly set to 0 in the unscented transform, α is the scaling factor that denotes the proximity of the sampled point to the mean, typically chosen as 0.01, β is generally selected as 2 under Gaussian distribution conditions [51].

3) Update $\widehat{x_k^-}$ and P_k^-:

$$\begin{cases} x_k^- = \sum_{i=0}^{2L} W_m^i x_k^i \\ P_k^- = \sum_{i=0}^{2L} \left(W_c^i (x_k^i - x_k^-)(x_k^i - x_k^-)^T\right) + Q_k \end{cases} \tag{28}$$

4) Compute the sigma points for the state variable x_k at time k:

$$\begin{cases} x_k^0 = \widehat{x_k^-} \\ x_k^i = \widehat{x_k^-} + \sqrt{(L+\lambda)P_k^-}, i = 1,2...L \\ x_k^i = \widehat{x_k^-} - \sqrt{(L+\lambda)P_k^-}, i = L+1, L+2, ...2L \end{cases} \tag{29}$$

5) Update the observed value $\widehat{y_k^-}$ and the observed variance predictions P_{yy} and P_{xy}:

$$\begin{cases} y_k^- = \sum_{i=0}^{2L} W_m^i y_k^i \\ P_{yy} = \sum_{i=0}^{2L} \left(W_c^i (y_k^i - y_k^-)(y_k^i - y_k^-)^T\right) + R_k \\ P_{xy} = \sum_{i=0}^{2L} W_c^i (x_k^i - x_k^-)(y_k^i - y_k^-)^T \end{cases} \tag{30}$$

6) Calculate K_k:

$$K_k = \frac{P_{xy}}{P_{yy}} \tag{31}$$

7) Update $\widehat{x_k^+}$ and P_k^+:

$$\begin{cases} \widehat{x_k^+} = \widehat{x_k^-} + K_k(y_k - \widehat{y_k^-}) \\ P_k^+ = P_k^- - K_k P_{yy} K_k^T \end{cases} \tag{32}$$

3.3. Optimization of UKF utilizing various matrix decomposition methods

In the traditional UKF, Cholesky Decomposition is used to find the matrix square root for acquiring the sigma points essential to the algorithm. However, Cholesky Decomposition necessitates the matrix being a positively definite symmetric matrix, which affects the stability of UKF in practical applications. In view of this, we decompose the error covariance matrix using three methods: QR Decomposition, Eigenvalue Decomposition and Singular Value Decomposition, since these three types of decomposition can be used to both find the square root of the matrix and to break down the non-positive definite matrix. Subsequently, the matrix square root, positive definite matrix, QR Decomposition, Eigenvalue Decomposition and Singular Value Decomposition are described.

Matrix square root: During the computation of sigma points, directly squaring the matrix is not feasible because a matrix might have multiple square roots and only square matrices possess square roots. In contrast, to compute the diagonal matrix square root, it is only necessary to replace each element on the diagonal with its square root. Thus, three matrix decomposition methods can be used to calculate the matrix square root.

Positive definite matrix: In general, the matrix P is considered to be a positive definite matrix if it is an n-dimensional matrix and satisfies the condition that $xPx^T > 0$ for all nonzero vector x. Furthermore, if all the eigenvalues of an nth order matrix are greater than 0, then the matrix also fulfills the condition of being positive definite matrix. Therefore, based on the comparison of the eigenvalues with 0, covariance matrices can be categorized as positive definite, negative definite, semi-positive definite, indefinite, and semi-negative definite.

QR Decomposition: The QR Decomposition is the decomposition of a matrix into the product of two matrices, which can be applied to general square matrices without requiring the matrix to be positive definite. The decomposition principle can be expressed by Eq. (33).

$$P = QR \tag{33}$$

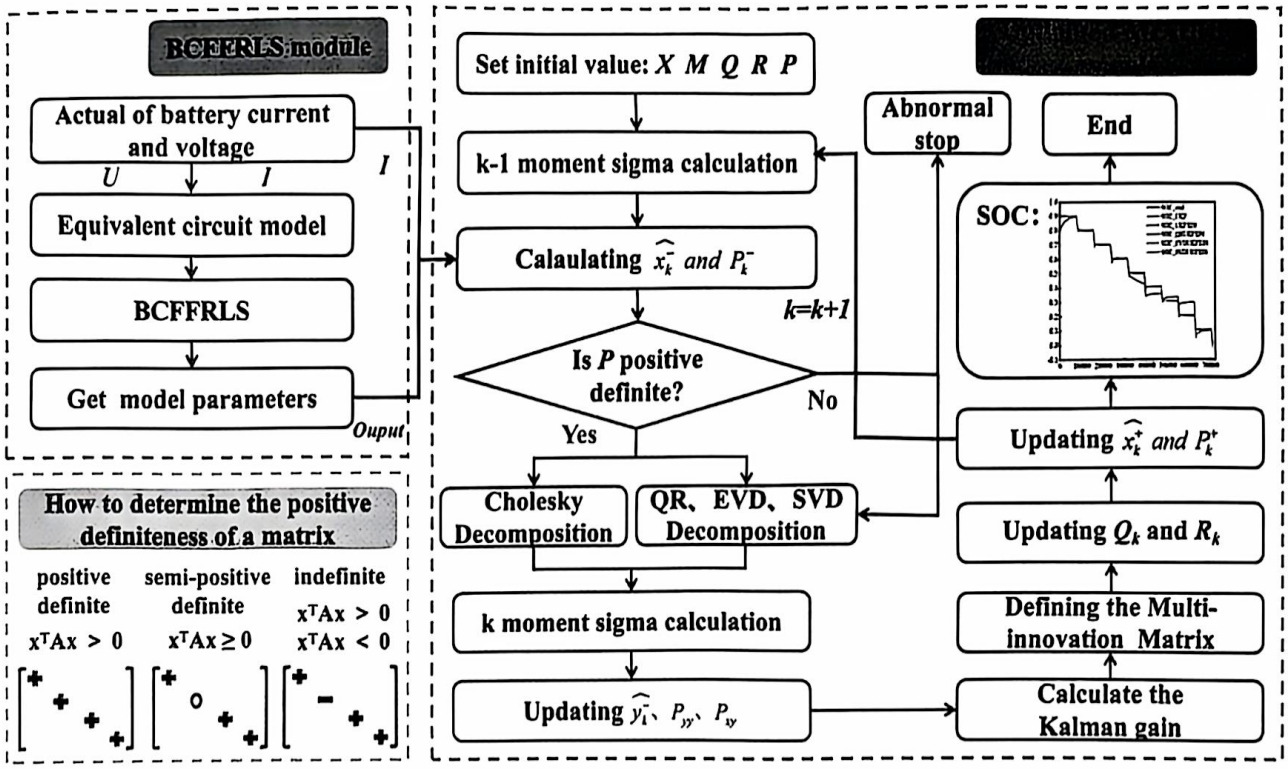

Fig. 3. Flowchart of the BCFFRLS-O-MIAUKF method.

Table 1
The OCV-SOC correspondence relationship.

OCV(V)	4.166	4.05	3.94	3.839	3.753	3.664	3.625	3.599	3.555	3.467	3.385
SOC	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0

Where P is the matrix to be broken down, Q is an orthogonal matrix, R is an upper triangular matrix, and RQ is a diagonal matrix by transformation. At this point, $\sqrt{P}$ can be expressed using the equation below:

$$\sqrt{P}=\sqrt{RQ} \tag{34}$$

Eigenvalue Decomposition: The process of Eigenvalue Decomposition entails breaking down a matrix into a product of its eigenvector and eigenvalue matrices. Note that only diagonalizable matrices can be subjected to eigenvalue decomposition. The decomposition principle can be expressed by Eq. (35).

$$P=Q\Lambda Q^T \tag{35}$$

Where Q is a standard orthogonal matrix, there is $QQ^T=1$, Λ is the diagonal matrix. At this point, $\sqrt{P}$ can be expressed using the equation below:

$$\sqrt{P}=Q\sqrt{\Lambda}Q^T \tag{36}$$

Singular Value Decomposition: The Singular Value Decomposition, on the other hand, is a generalization of the Eigenvalue Decomposition to arbitrary matrices, and is an important matrix decomposition method in linear algebra, its decomposition principle can be expressed by Eq. (37).

$$P=U\Sigma V^T \tag{37}$$

Where U is the left singular matrix and V is the right singular matrix, Σ is the diagonal matrix. At this point, $\sqrt{P}$ can be expressed using the equation below:

$$\sqrt{P}=U\sqrt{\Sigma}V^T \tag{38}$$

In practical implementations of the UKF, the error covariance matrix is typically set as a symmetric matrix that is positive definite. Therefore, all three of the aforementioned decomposition methods can be used to replace Cholesky Decomposition, the following is an example of the result of Singular Value Decomposition, bringing Eq. (38) into Eq. (26), obtaining a new formula for calculating the sigma sampling point, here's an example of calculating the sigma sampling points at time k-1:

$$\begin{cases} x_{k-1}^0=\widehat{x_{k-1}^+} \\ x_{k-1}^i=\widehat{x_{k-1}^+}+\sqrt{(L+\lambda)}U\sqrt{\Sigma}V^T, i=1,2...L \\ x_{k-1}^i=\widehat{x_{k-1}^+}-\sqrt{(L+\lambda)}U\sqrt{\Sigma}V^T, i=L+1,L+2,...2L \end{cases} \tag{39}$$

3.4. MIAUKF algorithm

Section 3.2 describes the implementation process of the UKF. Based on this, the MIAUKF includes two additional steps: defining the multi-interest matrix, adjusting the covariance matrix for process noise (Q_k) and the matrix for measurement noise (R_k):

1) Defining the multi-interest matrix $E_{p,k}$ and $K_{p,k}$:

$$\begin{cases} E_{p,k}=\left[e_k,e_{k-1},\cdots,e_{k-p+1}\right] \\ K_{p,k}=\left[K_k,K_{k-1},\cdots,K_{k-p+1}\right] \\ e_k=y_k-\bar{y_k} \end{cases} \tag{40}$$

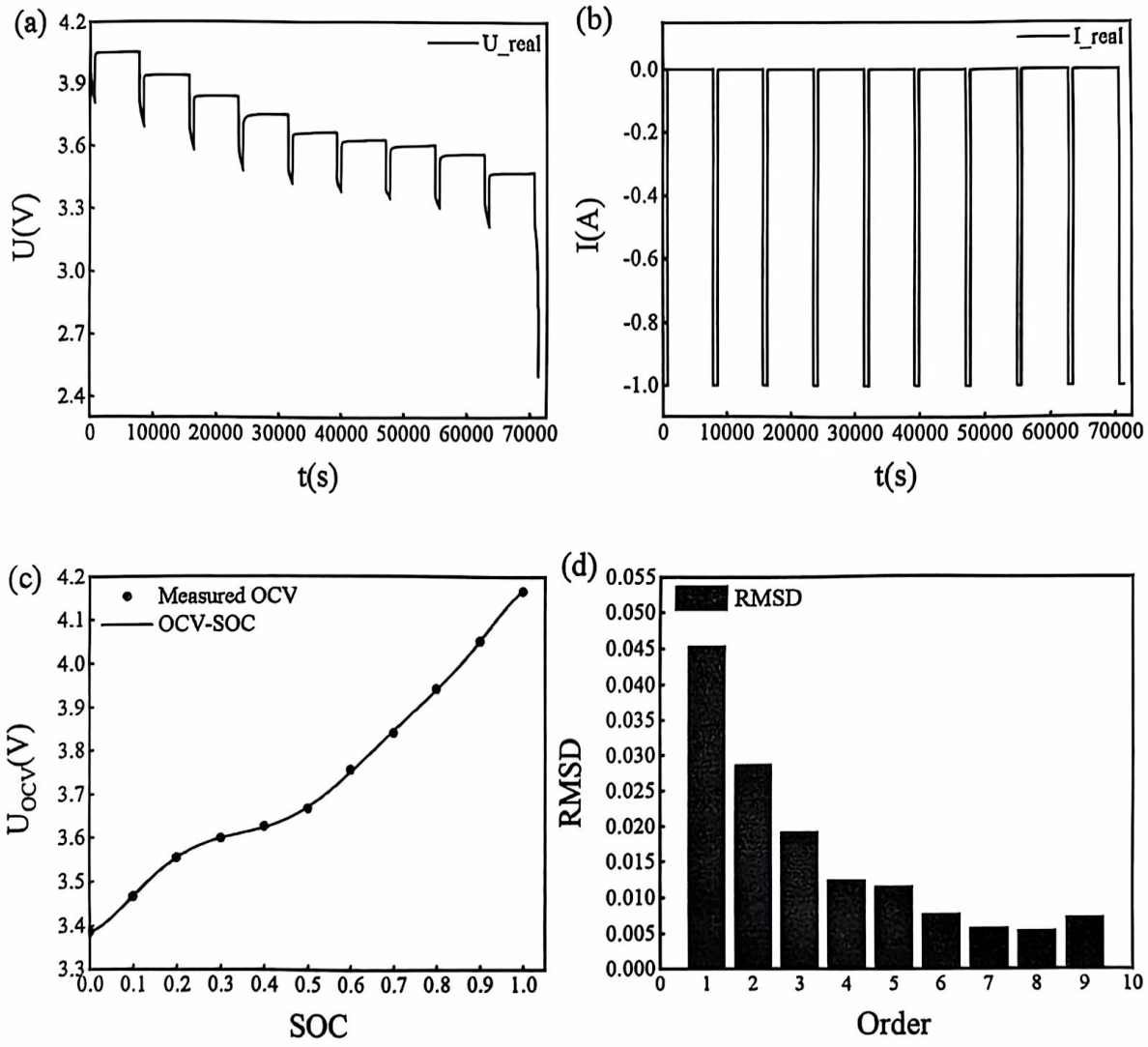

Fig. 4. (a) Voltage Profile from the HPPC trial. (b) Current Profile from the HPPC trial. (c) Fitted OCV-SOC curve. (d) RMSD of curves fitted at varying orders.

where p is the new interest length.

2) Adaptive adjustment of Q_k and R_k:

$$\begin{cases} Q_k = K_k F_k K_k^T \\ F_k = \dfrac{1}{M} \sum_{j=k-M+1}^{k} \varepsilon_k \varepsilon_k^T \\ R_k = F_k + C_k P_{k-1} C_k^T \end{cases} \tag{41}$$

where M is the size of the opening window.

3) Update $\widehat{x_k^+}$ and P_k^-:

$$\begin{cases} \widehat{x_k^+} = \widehat{x_k^-} + K_{pk} E_{pk} \\ P_k^+ = P_k^- - K_k P_{yy} K_k^T \end{cases} \tag{42}$$

The aforementioned formula illustrates that the algorithm adjusts the system noise covariance using the novel interest sequence from both measured and estimated voltages. In updating the state variable estimates, it incorporates past novel interest vectors to construct multi-novel interest matrix, enhancing the precision of the system state variable estimates. However, this continual adjustment of the system noise covariance matrix also elevates the likelihood of the covariance matrix turning into a non-positive definite. Consequently, introducing a new matrix decomposition method is essential to preserve the algorithm's stability.

3.5. Battery model equation

For *SOC* estimation of batteries with KF series of algorithms, it's crucial to determine the observation equations and state equations of the battery system, aligning with the mathematical relationships of the second-order *RC* model.

By regarding the state variables of the system as the *SOC*, U_1 and U_2, and choosing the U_t as the observation variable, the state and observation equations of the system can be formulated by integrating Eq. (2) and Eq. (23).

$$\begin{cases} \begin{bmatrix} SOC_k \\ U_{1,k} \\ U_{2,k} \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \exp\left(-\frac{T_s}{\tau_1}\right) & 0 \\ 0 & 0 & \exp\left(-\frac{T_s}{\tau_2}\right) \end{pmatrix} \begin{pmatrix} SOC_{k-1} \\ U_{1,k-1} \\ U_{2,k-1} \end{pmatrix} + \begin{pmatrix} -\frac{T_s}{C_n} \\ R_1\left(1-\exp\left(-\frac{T_s}{\tau_1}\right)\right) \\ R_2\left(1-\exp\left(-\frac{T_s}{\tau_2}\right)\right) \end{pmatrix} u_{k-1} \\ U_{t,k} = U_{OC}(SOC_k) + (0 \quad -1 \quad -1) \begin{pmatrix} SOC_k \\ U_{1,k} \\ U_{2,k} \end{pmatrix} - u_{k-1}R_0 \end{cases} \tag{43}$$

From Eq. (24):

$$\begin{cases} x_k = [SOC_k \ U_{1,k} \ U_{2,k}]^T \\ u_k = I_k \\ A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \exp\left(-\frac{T_s}{\tau_1}\right) & 0 \\ 0 & 0 & \exp\left(-\frac{T_s}{\tau_2}\right) \end{pmatrix} \\ B = \begin{pmatrix} -\frac{T_s}{C_n} \\ R_1\left(1-\exp\left(-\frac{T_s}{\tau_1}\right)\right) \\ R_2\left(1-\exp\left(-\frac{T_s}{\tau_2}\right)\right) \end{pmatrix} \\ C = (0 \quad -1 \quad -1) \\ D = [-R_0] \end{cases} \tag{44}$$

After obtaining the parameters and state space equations, the SOC estimation for the battery can be achieved using the above algorithmic steps. Fig. 3 illustrates the general framework of the BCFFRLS-O-MIAUKF algorithm. This study utilizes a second-order *RC* model to examine the discharge behavior of Li-ion batteries, and BCFFRLS is fused with the optimized UKF to accomplish real-time identification of model parameters and precise online SOC estimation. The joint algorithm addresses the stability issue that plagues traditional UKF without significantly increasing algorithmic complexity, thus fully embodying the advantages of both algorithms.

4. Experimental analyses

4.1. Experimental test platform

In this study, the experimental setup comprises a battery test system responsible for controlling and documenting the battery's charge and discharge cycles, this system plays a key role in recording data like voltage, current, and time. Additionally, a thermostat regulates the ambient temperature around the battery. To maintain consistency and Open Circuit Voltage (*OCV*) stability before and after the experiment, all experiments were carried out at a steady temperature of 25 °C, without taking into account the potential impact of battery aging on the obtained result. Utilizing this equipment framework, all lithium battery testing experiments can be completed.

4.2. HPPC experiment

The *OCV* of Li-ion batteries reflects its terminal voltage at no load when the internal electrochemical reaction has reached equilibrium. Typically, after an adequate period of rest, the *OCV* is expected to be numerically similar to the battery's terminal voltage. Given that direct measurement of the terminal voltage is impractical, the *OCV* is used as a proxy to determine the terminal voltage at a given *SOC*.

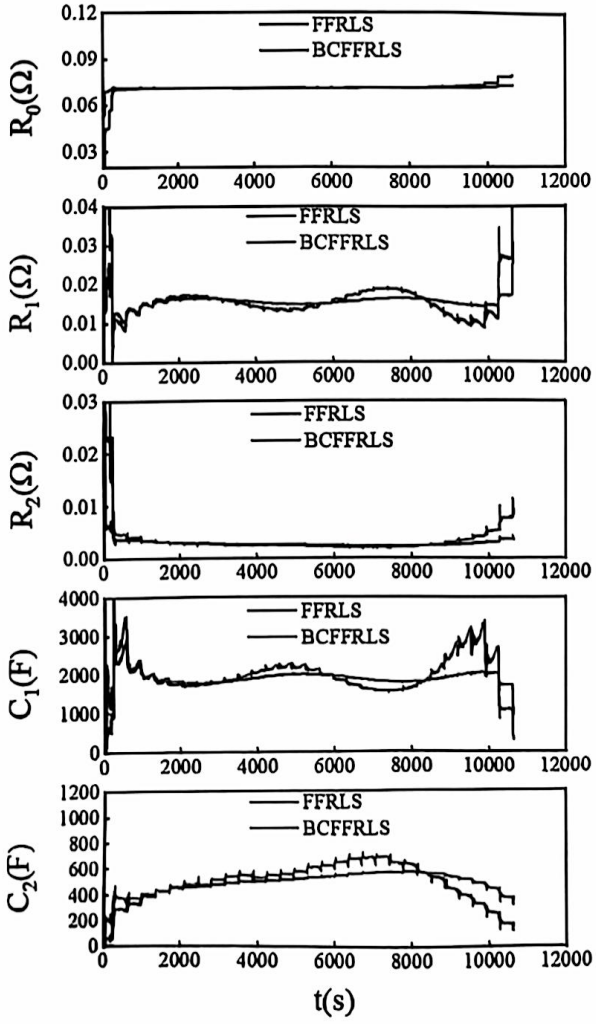

Fig. 5. Parameter identification outcomes by FFRLS and BCFFRLS in the DST trial.

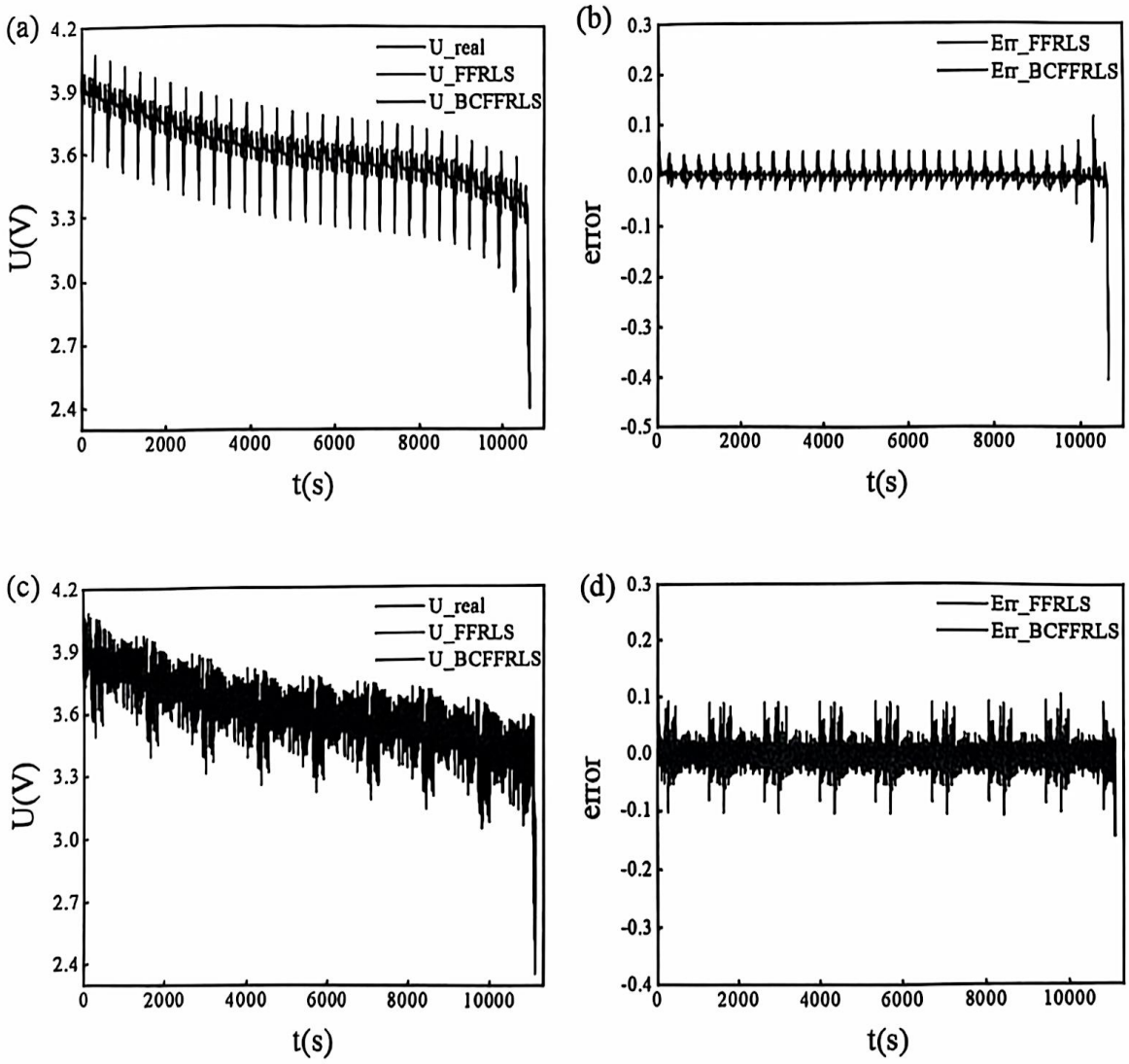

Fig. 6. (a) Comparison of the simulated and actual voltages in the DST trial. (b) Voltage error in the DST trial. (c) Comparison of the simulated and actual voltages in the FUDS trial. (d) Voltage error in the FUDS trial.

Table 2
Performance indicators of FFRLS algorithm and BCFFRLS algorithm under DST.

Algorithm	Max	Mean	RMSD
FFRLS	0.4040	0.0106	0.0207
BCFFRLS	0.3724	0.0052	0.0170

Table 3
Performance indicators of FFRLS algorithm and BCFFRLS algorithm under FUDS.

Algorithm	Max	Mean	RMSD
FFRLS	0.1068	0.0145	0.0191
BCFFRLS	0.1423	0.0076	0.0128

Whereas the functional link between *OCV* and *SOC* is nonlinear, in an effort to obtain data to determine the association between them, constant current HPPC experiments must be performed on the battery in the following steps: (1) After fully charging the battery, leave it to stand for 2 h. Mark the *SOC* at this moment as 1, and the initial *OCV* is recorded. (2) The battery underwent a discharge process at a current of 1A until the *SOC* of the battery decreased to 90 %. (3) Following the discharge, the battery was allowed to rest for 2 h, after which the *OCV* is documented. (4) Repeat the aforementioned steps (2) and (3) until the *SOC* decreases to 0 %, or the experiment concludes when the discharge cut-off voltage is reached. Table 1 below illustrates the measured OCV-SOC relationships, whilst Fig. 4(a) and (b) depicts the corresponding changes in voltage and current.

The data in Table 1 enables fitting of the OCV-SOC relationship. The OCV-SOC fitted curves for various orders were derived using MATLAB's Curve Fitting Tool. The accuracy of fitting was assessed using the Root Mean Square Deviation (RMSD), with a lower RMSD value indicating a superior model fit and greater precision in data prediction. Typically, the RMSD diminishes as the order increases, but there is a potential for over-fitting at higher orders. For an optimal balance of accuracy and complexity, we chose an eighth-order fitted curve with its RMSD of 0.0054 to depict the OCV-SOC relationship. The OCV-SOC fitting curves and RMSD curves are depicted in Fig. 4(c) and (d), with the corresponding mathematical relationships are detailed in Eq. (45).

$$U_{OCV}(SOC) = -95.6819SOC^8 + 332.3658SOC^7 - 427.7337SOC^6 + 228.6584SOC^5 - 19.3592SOC^4 - 24.6255SOC^3 + 6.7687SOC^2 + 0.3883SOC + 3.385 \quad (45)$$

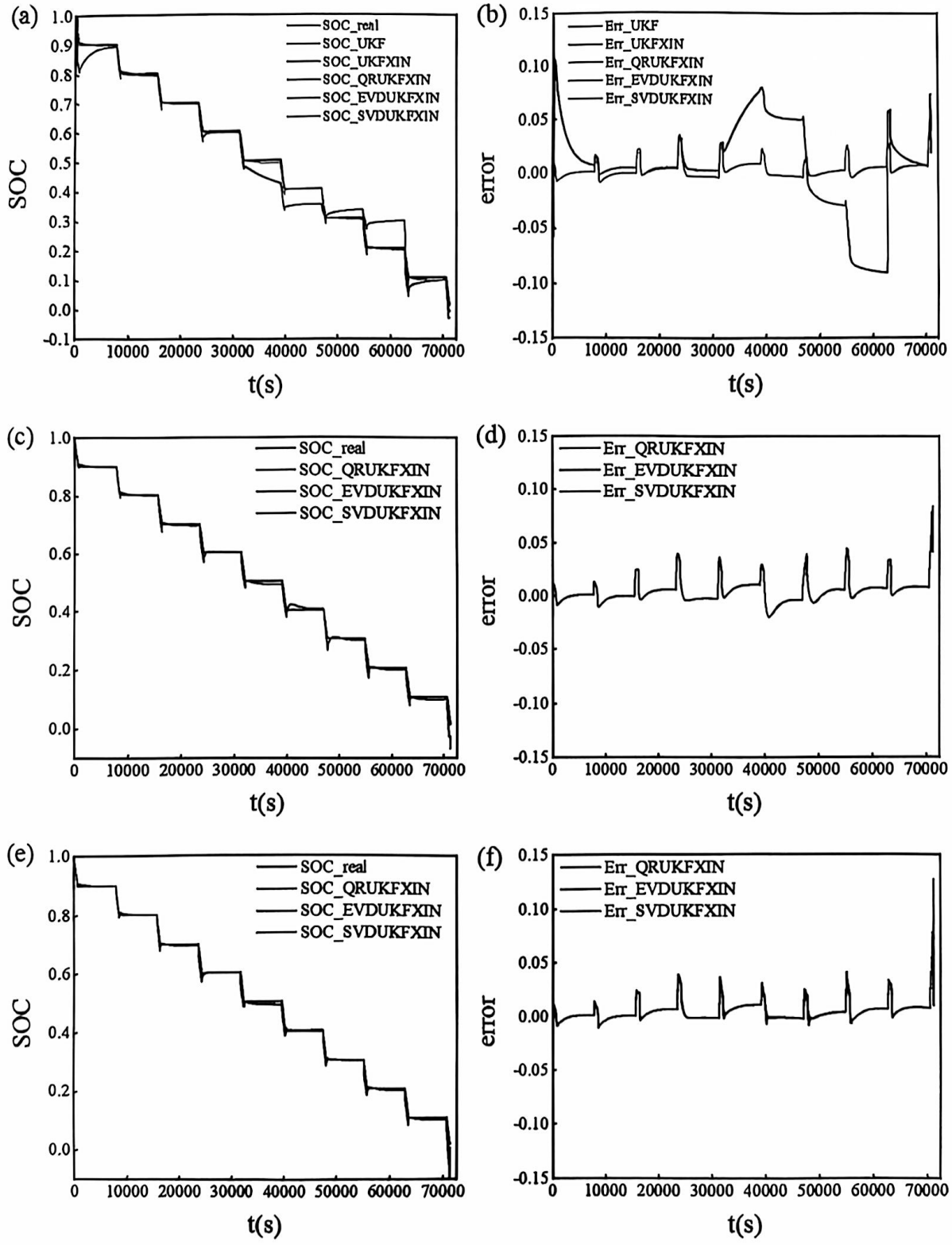

Fig. 7. (a) SOC estimation outcomes when the P is positive definite. (b) SOC estimation errors when the P is positive definite. (c) SOC estimation outcomes when the P is semi-positive definite. (d) SOC estimation errors when the P is semi-positive definite. (e) SOC estimation outcomes when the P is indefinite. (f) SOC estimation errors when the P is indefinite.

Table 4
The performance indicators of the UKF algorithm when the P is positive definite.

Algorithm	Max	Mean	RMSD	Time(s)
UKF	0.1060	0.0311	0.0421	2.2247
UKFXIN	0.0747	0.0057	0.0095	2.4035
QRUKFXIN	0.0747	0.0057	0.0095	5.6483
EVDUKFXIN	0.0747	0.0057	0.0095	2.9043
SVDUKFXIN	0.0747	0.0057	0.0095	2.8986

Table 5
The performance indicators of the UKF algorithm when the P is semi-positive definite.

Algorithm	Max	Mean	RMSD	Time(s)
QRUKFXIN	0.2084	0.0074	0.0110	5.3171
EVDUKFXIN	0.0787	0.0074	0.0116	2.7556
SVDUKFXIN	0.0787	0.0074	0.0116	2.8381

Table 6
The performance indicators of the UKF algorithm when the P is indefinite.

Algorithm	Max	Mean	RMSD	Time(s)
QRUKFXIN	0.1271	0.0063	0.0101	6.3091
EVDUKFXIN	0.1271	0.0059	0.0101	6.2799
SVDUKFXIN	0.0793	0.0061	0.0102	2.8911

5. Results and discussion

5.1. Parameter identification results

To verify the feasibility of the constructed model and the effectiveness of the proposed parameter identification method, various algorithms were employed for parameter identification. The parameter identification results are presented in Fig. 5. The outcomes of the identified parameters and the actual operating current data are fed into a second-order RC model. Subsequently, different algorithms were employed to derive simulated and actual voltage curves under two operational conditions, to validate the accuracy of the model and parameters. Results from the simulations under Dynamic Stress Test (DST) and Federal Urban Driving Scheme (FUDS) conditions, relevant for urban driving scenarios, are shown in Fig. 6.

Fig. 6(b) and (d) illustrate that the voltages simulated by the BCFFRLS algorithm are more aligned with the actual experimental voltages when compared to those produced by the FFRLS algorithm. This phenomenon can be attributed to the BCFFRLS algorithm reducing the impact of measurement errors on parameter estimation by minimizing the square of the sum of the model bias and measurement residuals. Consequently, the BCFFRLS algorithm achieves more precise simulation results.

Tables 2 and 3 list three performance metrics for both algorithms: Maximum Error (Max), RMSD, and Mean Absolute Error (MAE). Lower values in these metrics indicate the greater simulation accuracy and superior performance of the chosen algorithm. Therefore, when comparing these metrics of the BCFFRLS and FFRLS algorithms during model parameter identification, it is apparent that the BCFFRLS algorithm significantly outperforms the FFRLS algorithm.

5.2. Stability and velocity analysis of SOC estimation

In real-world implementations of the UKF, considering the update of the error covariance matrix, it may become non-positive definite during the iteration process, which poses certain challenges to the algorithm's stability. Therefore, in this section, we take the HPPC case as an example to estimate the *SOC*, and firstly, we verify whether the proposed three matrix decomposition methods can replace the Cholesky Decomposition in the original UKF by configuring the positive definite covariance matrix (P). The initial configurations for the P, Q_k and R_k are provided in Eq. (46). Fig. 6 illustrates the outcomes of SOC estimation using different UKF algorithms in HPPC operational conditions.

$$\begin{cases} P = \begin{pmatrix} 1\times10^{-3} & 0 & 0 \\ 0 & 1\times10^{-3} & 0 \\ 0 & 0 & 1\times10^{-2} \end{pmatrix} \\ Q = \begin{pmatrix} 1\times10^{-5} & 0 & 0 \\ 0 & 1\times10^{-5} & 0 \\ 0 & 0 & 1\times10^{-5} \end{pmatrix} \\ R = 0.01 \end{cases} \tag{46}$$

Overall, the *SOC* estimation outcomes from various UKF algorithms correspond with the actual *SOC*. As indicated in Fig. 7(b), the *SOC* estimations from UKF exhibit greater discrepancies in the later stages of the estimation process, which is because UKF does not compute the sigma points for the priori probability distribution of the state variable at time k and does not update the P, resulting in a significant decrease in *SOC* estimation accuracy at the later stage. For a more nuanced comparison of the accuracy among five UKF algorithms, three performance indicators for *SOC* estimation were calculated: MAX, RMSD and MAE. Table 4 shows that the *SOC* estimation results for UKFXIN, QR-UKFXIN, EVD-UKFXIN and SVD-UKFXIN are remarkably similar, with identical values for these three metrics. As explained in Section 3.3, when the other initial parameters are consistent and the initial P is symmetric positive definite, the QR Decomposition, Eigenvalue Decomposition and Singular Value Decomposition can replace the Cholesky decomposition.

Next, we compare the stability of five different UKF algorithms. In practical applications of the UKF, instances of semi-negative and negative definite error covariance matrices are infrequent, leading us to primarily concentrate on cases with semi-positive and indefinite error covariance matrices. It is essential to note that our focus is solely on alterations to the error covariance matrix, while keeping the Q_k and R_k parameters constant. The specific details are as follows:

1) The initial P is a symmetric semi-positive definite matrix, defined as

$$P = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1\times10^{-3} & 0 \\ 0 & 0 & 1\times10^{-2} \end{pmatrix}$$

2) The initial P is a symmetric indefinite matrix, defined as

$$P = \begin{pmatrix} 1\times10^{-3} & 0 & 0 \\ 0 & -1\times10^{-3} & 0 \\ 0 & 0 & 1\times10^{-2} \end{pmatrix}$$

When the P is non-positive definite, the traditional UKF algorithm and the UKFXIN algorithm that considers covariance updating are ineffective in estimating SOC. Conversely, this study proposes innovative algorithms, namely QR-UKFXIN, EVD-UKFXIN, and SVD-UKFXIN, which have demonstrated successful SOC estimation capabilities. The estimation outcomes are illustrated in Fig. 7. Because of the difficulty in finding distinctions from the charts, we have also calculated the three-performance metrics for *SOC* estimation, as detailed in Tables 5 and 6. It's noteworthy that the absolute values of the diagonal elements of the initial P are identical in both cases, resulting in the same three performance metrics for their *SOC* estimations.

The matrix decomposition results of the three algorithms might vary slightly due to factors like computer character length limitations and differences in decomposition methods, further causing discrepancies in the estimation results for *SOC*. Furthermore, by analyzing the runtime of these algorithms, it's observed that all three proposed algorithms exhibit

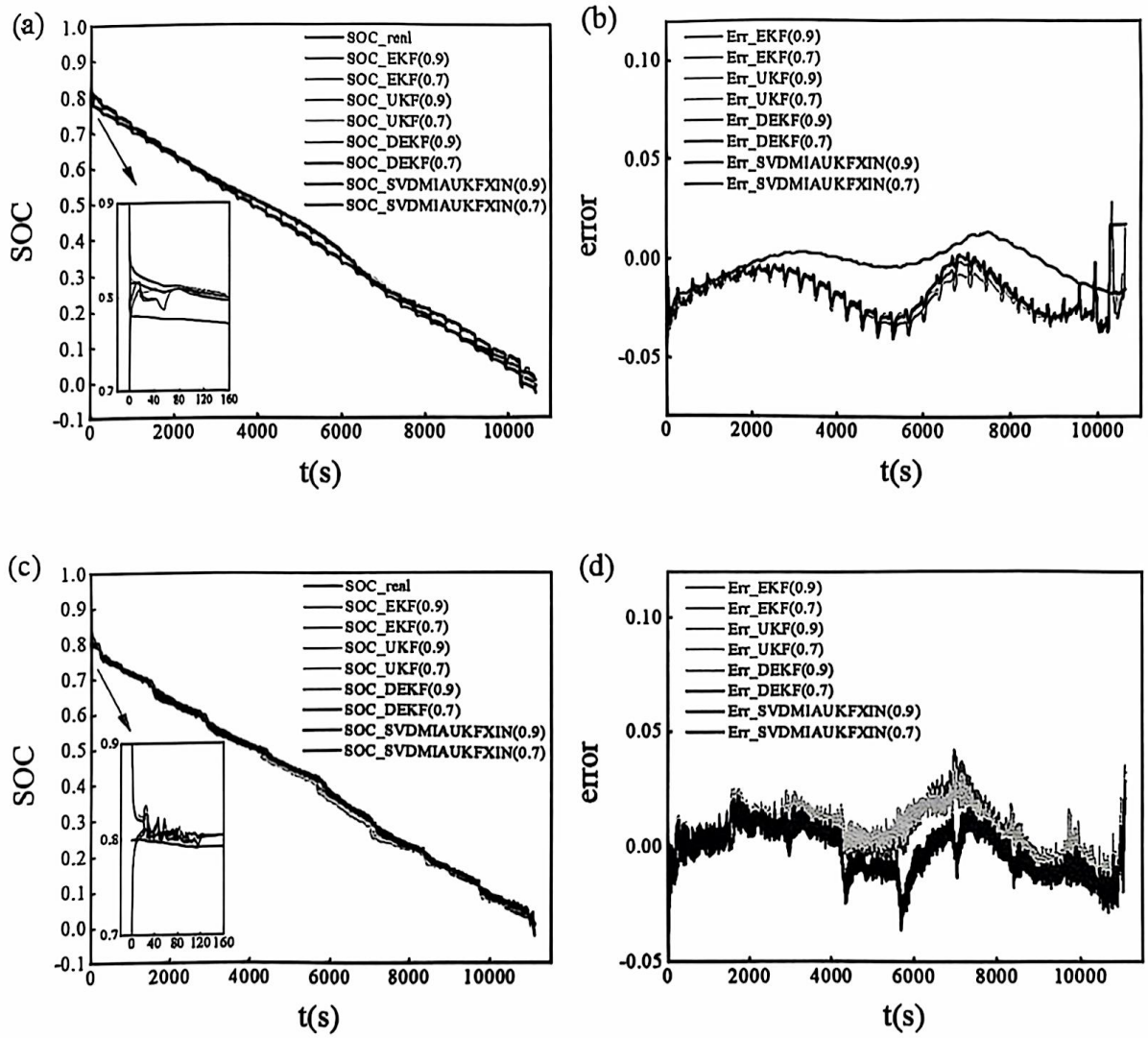

Fig. 8. (a) SOC estimation outcomes in the DST trial. (b) SOC estimation errors in the DST trial. (c) SOC estimation outcomes in the FUDS trial. (d) SOC estimation errors in the FUDS trial.

Table 7
Performance indicators of different KF algorithm in the DST trial.

Algorithm (SOC)	Max	Mean	RMSD
EKF (0.9)	0.1	0.0195	0.0219
EKF (0.7)	0.1	0.0193	0.0218
UKF (0.9)	0.1	0.0193	0.0212
UKF (0.7)	0.1	0.0192	0.0210
DEKF (0.9)	0.1	0.0171	0.0197
DEKF (0.7)	0.1	0.0171	0.0197
SVDMIAUKF (0.9)	0.1	0.0071	0.0092
SVDMIAUKF (0.7)	0.1	0.0070	0.0092

Table 8
Performance indicators of different KF algorithm in the FUDS trial.

Algorithm (SOC)	Max	Mean	RMSD
EKF (0.9)	0.1	0.0105	0.0130
EKF (0.7)	0.1	0.0104	0.0130
UKF (0.9)	0.1	0.0104	0.0123
UKF (0.7)	0.1	0.0103	0.0122
DEKF (0.9)	0.1	0.0090	0.0104
DEKF (0.7)	0.1	0.0089	0.0103
SVDMIAUKF (0.9)	0.1	0.0076	0.0091
SVDMIAUKF (0.7)	0.1	0.0076	0.0091

longer run-times compared to the traditional UKF. Notably, the runtime of the SVD-UKFXIN algorithm is similar to that of the traditional UKF, making it a viable option for practical applications.

5.3. Robustness analysis of SOC estimation

It is widely recognized that the precision of KF series of algorithms is significantly influenced by the accuracy of the initial parameters. However, in practical applications, the initial *SOC* values may be incorrect, presenting some challenges to the robustness of the algorithm. Therefore, in this section, we take the DST and FUDS working conditions as an example to estimate the *SOC*, where the initial *SOC* is set to 0.9 and 0.7 to verify the robustness of the algorithms, and take the SVD-UKFXIN algorithm as an example, building upon this, the multi-interest theory and the adaptive link were introduced to refine *SOC* estimation accuracy. The estimation outcomes are also evaluated and compared with the results from several prevalent algorithms such as EKF, UKF, and DEKF. When a substantial discrepancy exists between the initial *SOC* estimation and the actual *SOC*, choosing a smaller window *M* could result in an enlarged covariance matrix for process and measurement

noise. subsequently impacting the accuracy of *SOC* estimation outcomes. Therefore, in such circumstances, we have set M at 100 while keeping other parameters constant.

The robustness analysis outcomes are depicted in Fig. 8(a) and (c). Through local zoom-in observation, it is shown that the proposed methods rapidly rectify the initial SOC discrepancies and accurately follow the real *SOC*. Moreover, the *SOC* estimation error curves in Fig. 8 (b) and (d) illustrate that once the initial error is adjusted, the *SOC* estimation error remains consistently in a range of 0 %–5 % for various initial *SOC* situations. Performance metrics for the four algorithms are calculated in Tables 7 and 8. The results demonstrate that the Max, Mean, and RMSD values for the SVD-MIAUKFXIN algorithm are lower than those recorded for the EKF, UKF and DEKF algorithms. Consequently, this indicates that the SVD-MIAUKFXIN algorithm is satisfactorily robust to inaccurate initial *SOC* values and surpasses the other algorithms in performance.

6. Conclusions

In this paper, we introduce an innovative approach to enhance the accuracy and stability of SOC estimation in battery management systems. Our innovation involves the adoption of three new matrix decomposition methods to replace the traditional Cholesky Decomposition utilized in the original UKF. Additionally, we optimize the FFRLS algorithm using bias compensation, introducing the BCFFRLS algorithm for improved parameter estimation accuracy. The feasibility of the constructed model and the proposed BCFFRLS algorithm is verified under DST and FUDS trials, and the accuracy reaches 0.76 %, compared to the FFRLS algorithm. Then the HPPC trial verifies that the QR-UKFXIN, EVD-UKFXIN and SVD-UKFXIN can estimate the *SOC* even when the error covariance matrix is non-positive, which enhances the stability of the UKF series of algorithms in practical applications. Finally, the accuracy of the proposed algorithm was verified in DST and FUDS experiments, and the robustness of the algorithm was also tested using two different initial SOC values. The outcomes indicate that, compared to the EKF, UKF, and DEKF, the accuracy of the proposed algorithm reaches 0.7 %. These outcomes confirms that the BCFFRLS-SVDMIAUKF proposed here can effectively estimate the *SOC* and demonstrates high robustness and convergence.

Indeed, the algorithm presented in this study has its imperfections. To optimize the FFRLS algorithm using the bias compensation idea, the initial value of the parameters must be set twice, this results in a large error at the start of the estimation. Future work will focus on optimizing the initial parameter values and the forgetting factor to minimize this error. Furthermore, due to the introduction of multi-interest theory and the adaptive link, coupled with the application of Singular Value Decomposition to improve the algorithm's stability, unavoidably leads to increased computational complexity and longer execution times. To address these challenges, subsequent research will explore methods to optimize computational efficiency without compromising the algorithm's robustness. Moreover, exploring the application of novel matrix decomposition techniques within Cubature Kalman Filtering could offer valuable insights, potentially broadening the scope of our findings. This direction invites further investigation into refining filter algorithms for improved performance and reliability in nuanced applications.

CRediT authorship contribution statement

Sijing Wang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Pan Huang:** Writing – review & editing, Visualization, Validation, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Cheng Lian:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization. **Honglai Liu:** Resources, Project administration, Funding acquisition.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 22278127), the Fundamental Research Funds for the Central Universities (No. 2022ZFJH004), and Shanghai Pilot Program for Basic Research (22T01400100-18).

References

[1] R. Xiong, J. Cao, Q. Yu, et al., Critical review on the battery state of charge estimation methods for electric vehicles, IEEE Access 6 (2017) 1832–1843.
[2] M. Waseem, M. Ahmad, A. Parveen, et al., Battery technologies and functionality of battery management system for EVs: current status, key challenges, and future prospectives, J. Power Sources 580 (2023) 233349.
[3] M. Shen, Q. Gao, A review on battery management system from the modeling efforts to its multiapplication and integration, Int. J. Energy Res. 43 (10) (2019) 5042–5075.
[4] S. Piller, M. Perrin, A. Jossen, Methods for state-of-charge determination and their applications, J. Power Sources 96 (1) (2001) 113–120.
[5] R.J. Ball, R. Kurian, R. Evans, et al., Failure mechanisms in valve regulated lead/acid batteries for cyclic applications, J. Power Sources 109 (1) (2002) 189–202.
[6] J. Garche, A. Jossen, H. Doring, The influence of different operating conditions, especially over-discharge, on the lifetime and performance of lead/acid batteries for photovoltaic systems, J. Power Sources 67 (1–2) (1997) 201–212.
[7] I. Snihir, W. Rey, E. Verbitskiy, et al., Battery open-circuit voltage estimation by a method of statistical analysis, J. Power Sources 159 (2) (2006) 1484–1487.
[8] A.B. Djordjevic, D.M. Karanovic, Battery testing by calculated discharge-curve method—constant resistive load algorithm, J. Power Sources 162 (2) (2006) 920–926.
[9] X. Li, T. Long, J. Tian, et al., Multi-state joint estimation for a lithium-ion hybrid capacitor over a wide temperature range, J. Power Sources 479 (2020) 228677.
[10] D.N.T. How, M.A. Hannan, M.S.H. Lipu, et al., State of charge estimation for lithium-ion batteries using model-based and data-driven methods: a review, IEEE Access 7 (2019) 136116–136136.
[11] S. Wang, Y. Fan, et al., Improved anti-noise adaptive long short-term memory neural network modeling for the robust remaining useful life prediction of lithium-ion batteries, Reliab. Eng. Syst. Saf. 230 (2023) 108920.
[12] S. Wang, F. Wu, P. Takyi-Aninakwa, et al., Improved singular filtering-Gaussian process regression-long short-term memory model for whole-life-cycle remaining capacity estimation of lithium-ion batteries adaptive to fast aging and multi-current variations, Energy 284 (2023) 128677.
[13] D. Huang, Z. Chen, C. Zheng, et al., A model-based state-of-charge estimation method for series-connected lithium-ion battery pack considering fast-varying cell temperature, Energy 185 (2019) 847–861.
[14] Z. Xi, M. Dahmardeh, B. Xia, et al., Learning of battery model bias for effective state of charge estimation of lithium-ion batteries, IEEE Trans. Veh. Technol. 68 (9) (2019) 8613–8628.
[15] M.K. Tran, M. Mathew, S. Janhunen, et al., A comprehensive equivalent circuit model for lithium-ion batteries, incorporating the effects of state of health, state of charge, and temperature on model parameters, J. Energy Storage 43 (2021) 103252.
[16] J. Shin, W. Kim, K. Yoo, et al., Vehicular level battery modeling and its application to battery electric vehicle simulation, J. Power Sources 556 (2023) 232531.
[17] Q. Zhang, Q. Guo, R.E. White, Semi-empirical modeling of charge and discharge profiles for a LiCoO2 electrode, J. Power Sources 165 (1) (2007) 427–435.
[18] F. Ringbeck, M. Garbade, D U. Sauer, Uncertainty-aware state estimation for electrochemical model-based fast charging control of lithium-ion batteries, J. Power Sources 470 (2020) 228221.
[19] J.R. Miller, S. Butler, Storage system design based on equivalent-circuit-model simulations: comparison of eight different electrochemical capacitor storage systems, J. Power Sources 491 (2021) 229441.
[20] Q.K. Wang, Y.J. He, J.N. Shen, et al., State of charge-dependent polynomial equivalent circuit modeling for electrochemical impedance spectroscopy of lithium-ion batteries, IEEE Trans. Power Electron. 33 (10) (2017) 8449–8460.
[21] C. Li, N. Cui, Z. Cui, et al., Novel equivalent circuit model for high-energy lithium-ion batteries considering the effect of nonlinear solid-phase diffusion, J. Power Sources 523 (2022) 230993.
[22] K. Yang, Y. Tang, Z. Zhang, Parameter identification and state-of-charge estimation for lithium-ion batteries using separated time scales and extended kalman filter, Energies 14 (4) (2021) 1054.

[23] Z. Yu, L. Xiao, H. Li, et al., Model parameter identification for lithium batteries using the coevolutionary particle swarm optimization method, IEEE Trans. Ind. Electron. 64 (7) (2017) 5690–5700.
[24] J. Tian, Y. Wang, D. Yang, et al., A real-time insulation detection method for battery packs used in electric vehicles, J. Power Sources 385 (2018) 1–9.
[25] V.H. Duong, H.A. Bastawrous, Pang Zhang, et al., Online state of charge and model parameters estimation of the LiFePO4 battery in electric vehicles using multiple adaptive forgetting factors recursive least-squares, J. Power Sources 296 (2015) 215–224.
[26] Y. Li, J. Chen, F. Lan, Enhanced online model identification and state of charge estimation for lithium-ion battery under noise corrupted measurements by bias compensation recursive least squares, J. Power Sources 456 (2020) 227984.
[27] M. Khodarahmi, V. Maihami, A review on Kalman filter models, Arch. Comput. Methods Eng. 30 (1) (2023) 727–747.
[28] G. Galuppini, M.D. Berliner, D.A. Cogswell, et al., Nonlinear identifiability analysis of multiphase porous electrode theory-based battery models: a lithium iron phosphate case study, J. Power Sources 573 (2023) 233009.
[29] B. Xiong, J. Zhao, Z. Wei, et al., Extended Kalman filter method for state of charge estimation of vanadium redox flow battery using thermal-dependent electrical model, J. Power Sources 262 (2014) 50–61.
[30] T. Wang, S. Chen, H. Ren, et al., Model-based unscented Kalman filter observer design for lithium-ion battery state of charge estimation, Int. J. Energy Res. 42 (4) (2018) 1603–1614.
[31] S. Zhuang, Y. Gao, A. Chen, et al., Research on estimation of state of charge of li-ion battery based on cubature kalman filter, J. Electrochem. Soc. 169 (10) (2022) 100521.
[32] A. Tulsyan, Y. Tsai, R.B. Gopaluni, et al., State-of-charge estimation in lithium-ion batteries: a particle filter approach, J. Power Sources 331 (2016) 208–223.
[33] J. Han, D. Kim, M. Sunwoo, State-of-charge estimation of lead-acid batteries using an adaptive extended Kalman filter, J. Power Sources 188 (2) (2009) 606–612.
[34] F. Sun, X. Hu, Y. Zou, et al., Adaptive unscented Kalman filtering for state of charge estimation of a lithium-ion battery for electric vehicles, Energy 36 (5) (2011) 3531–3540.
[35] Z. Liu, Z. Zhao, Y. Qiu, et al., State of charge estimation for Li-ion batteries based on iterative Kalman filter with adaptive maximum correntropy criterion, J. Power Sources 580 (2023) 233282.
[36] Z. Wei, C. Zou, F. Leng, et al., Online model identification and state-of-charge estimate for lithium-ion battery with a recursive total least squares-based observer, IEEE Trans. Ind. Electron. 65 (2) (2017) 1336–1346.
[37] B. Xia, X. Zhao, R. De Callafon, et al., Accurate Lithium-ion battery parameter estimation with continuous-time system identification methods, Appl. Energy 179 (2016) 426–436.
[38] S. Zhang, X. Guo, X. Zhang, An improved adaptive unscented kalman filtering for state of charge online estimation of lithium-ion battery, J. Energy Storage 32 (2020) 101980.
[39] G. Dong, J. Wei, Z. Chen, et al., Remaining dischargeable time prediction for lithium-ion batteries using unscented Kalman filter, J. Power Sources 364 (2017) 316–327.
[40] W. Zhang, W. Shi, Z. Ma, Adaptive unscented Kalman filter based state of energy and power capability estimation approach for lithium-ion battery, J. Power Sources 289 (2015) 50–62.
[41] C. Xu, T. Cleary, D. Wang, et al., Online state estimation for a physics-based Lithium-Sulfur battery model, J. Power Sources 489 (2021) 229495.
[42] S. Lee, J. Kim, J. Lee, et al., State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge, J. Power Sources 185 (2) (2008) 1367–1373.
[43] Y. Zheng, W. Gao, M. Ouyang, et al., State-of-charge inconsistency estimation of lithium-ion battery pack using mean-difference model and extended Kalman filter, J. Power Sources 383 (2018) 50–58.
[44] X. Liu, W. Li, A. Zhou, PNGV equivalent circuit model and SOC estimation algorithm for lithium battery pack adopted in AGV vehicle, IEEE Access 6 (2018) 23639–23647.
[45] L. Zhang, H. Peng, Z. Ning, et al., Comparative research on RC equivalent circuit models for lithium-ion batteries of electric vehicles, Appl. Sci. 7 (10) (2017) 1002.
[46] X. Ding, D. Zhang, J. Cheng, et al., An improved Thevenin model of lithium-ion battery with high accuracy for electric vehicles, Appl. Energy 254 (2019) 113615.
[47] C. Zhang, J. Jiang, L. Zhang, et al., A generalized SOC-OCV model for lithium-ion batteries and the SOC estimation for LNMCO battery, Energies 9 (11) (2016) 900.
[48] F. Elmahdi, L. Ismail, M. Noureddine, Fitting the OCV-SOC relationship of a battery lithium-ion using genetic algorithm method[C]//E3S Web of Conferences, EDP Sci. 234 (2021) 00097.
[49] T. Ouyang, P. Xu, J. Chen, et al., Improved parameters identification and state of charge estimation for lithium-ion battery with real-time optimal forgetting factor, Electrochim. Acta 353 (2020) 136576.
[50] G.L. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation, J. Power Sources 134 (2) (2004) 277–292.
[51] F. Yang, S. Zhang, W. Li, et al., State-of-charge estimation of lithium-ion batteries using LSTM and UKF, Energy 201 (2020) 117664.