An improved adaptive weights correction-particle swarm optimization-unscented particle filter method for high-precision online state of charge estimation of lithium-ion batteries. output.

LI, Z., WANG, S., YU, C., QI, C., SHEN, X. and FERNANDEZ, C.

2024

An improved adaptive weights correction-particle swarm optimization-unscented particle filter method for high-precision online state of charge estimation of lithium-ion batteries

Zehao Li[1] · Shunli Wang[1,2] · Chunmei Yu[1] · Chuangshi Qi[1] · Xianfeng Shen[1] · Carlos Fernandez[3]

[1] School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China

[2] School of Electrical Engineering, Sichuan University, Chengdu 610065, China

[3] School of Pharmacy and Life Sciences, Robert Gordon University, AB10-7GJ, Aberdeen, UK

Abstract

In the battery management system (BMS), the state of charge (SOC) of lithium-ion batteries is an indispensable part, and the accuracy of SOC estimation has attracted wide attention. Accurate SOC estimation can improve the efficiency of battery use while ensuring battery safety and improving battery life. Taking ternary lithium battery as the research object, this paper proposes a parameter identification method using adaptive forgetting factor recursive least squares and an improved joint unscented particle filter algorithm to estimate SOC. Firstly, an adaptive method is used to select the appropriate forgetting factor value to improve the accuracy of the forgetting factor recursive least squares (FFRLS) method. Meanwhile, the improved particle swarm (IPSO) optimization algorithm that incorporates variable weights and shrinkage factors is utilized to make the best choice of the noise for the unscented Kalman filter (UKF) algorithm in order to improve the estimation accuracy of the UKF algorithm. At the same time, the UKF algorithm is used as the suggestion density function of the particle filter (PF) algorithm to form the unscented particle filter (UPF) algorithm. In this paper, the AFFRLS algorithm and IPSO-SDUPF algorithm are combined to estimate the SOC of Li-ion batteries in real time. Experimental results under different working conditions show that the proposed algorithm has good convergence and high stability for SOC estimation of lithium-ion batteries. The maximum estimation errors of this algorithm are 1.137% and 0.797% for BBDST and DST conditions at 25 °C, and 1.015% and 1.029% for BBDST and DST conditions at 35 °C, which are lower than those of the commonly used algorithms of EKF, SDUKF, IPSO-SDUKF, and SDUPF, and provide a reference for future. The maximum estimation errors are lower than those of the commonly used EKF, SDUKF, IPSO-SDUKF, and SDUPF algorithms, which provide a reference for the future high-precision SOC estimation of Li-ion batteries.

Keywords Lithium-ion battery · Charge state · Adaptive unscented Kalman filter algorithm · Adaptive forgetting factor recursive least squares method · Particle filter algorithm

Introduction

Today, because lithium-ion batteries have a high energy density, high electromotive force, no memory effect, large output power, long life, and other characteristics, they are widely used in electric vehicles, laptops, aerospace electronics, and other consumer electronic products [1, 2]. such as cell phones. With the wide application of lithium-ion batteries, the accuracy of the state of charge (SOC) of lithium-ion batteries [3] has attracted greater attention.

In the battery management system (BMS), one of the core parameters is the battery state of charge [4]. Yu et al., in their article, showed that safety standards are presented in terms of battery materials, battery cells, battery modules, battery systems, battery management systems, and vehicles [5]. The accurate prediction of the battery's SOC not only means the safe use of the battery, avoiding overcharging and over-discharge but also effectively protects the battery and extends the battery life [6, 7]. At the same time, the accuracy of SOC estimation will affect the operation effect of BMS.

Under certain discharge conditions, the remaining capacity of the rated capacity is defined as the SOC [8] value of the battery, which is a relative value, expressed as a percentage, and its value ranges from 0 to 100%.

At present, there are many SOC estimation methods for lithium-ion batteries, such as the Kalman filter algorithm [9], discharge experiment method, particle filter algorithm, ampere-hour integration method, and neural network method. The most commonly used is the ampere-hour integral method, but the ampere-hour integral method will gradually accumulate errors. The discharge experiment method is the simplest and most accurate method in traditional SOC prediction [10–12], but its efficiency is not high. However, the neural network method makes it difficult to establish a relatively accurate mathematical model for the whole process. Although the neural network method does not need an accurate mathematical model and can learn the internal law of a nonlinear system by learning sample data, it is easy to forget the old sample in the training process and fall into the local minimum when applying the standard BP neural network algorithm [13]. The convergence speed of the network is slow, and the determination of the number of nodes in the hidden layer is mostly based on the empirical formula, which lacks sufficient guidance of professional theory. This method cannot meet real-time requirements, so it is usually not used alone. The Kalman filter method, which cannot be ignored, provides the optimal solution for linear problems [14–17]. So far, the Kalman filter is still widely used. However, the Kalman filter is unable to solve the nonlinear problem of the charge and discharge state of lithium-ion batteries. As time goes by, the extended Kalman filter becomes a powerful tool to solve nonlinear filtering, but the EKF algorithm linearizes the battery model through Taylor series expansion, ignores the higher order term, and only uses the first-order partial derivative term of the non-linear function Taylor expansion, which will lead to large errors in state estimation. At this time, the UKF algorithm is proposed to directly approximate the random distribution by using the weighted summation of samples [18–21], avoiding the analytical derivation of nonlinear functions. For any nonlinear system, the unscented Kalman filter can obtain the posterior mean and covariance estimates accurate to the third order [22, 23]. However, the UKF algorithm assumes that the statistical characteristics of the system noise obey the Gaussian distribution when estimating the battery SOC, which leads to the reduction and loss of accuracy [24–26]. Subsequently, the particle filter was proposed, and the particle filter algorithm was not limited by the noise distribution. Particle filters can handle any nonlinear model and any noise distribution [27, 28]. However, the particle filter algorithm itself also has many problems, for example, the particle filter algorithm has a particle degradation phenomenon, and although resampling can reduce this phenomenon to a certain extent, but greatly increases the calculation amount. On this basis, this paper combines an unscented Kalman filter and particle filter to obtain an unscented particle filter algorithm [29]. PSO algorithm is widely used as one of the optimization algorithms, and in this paper [30], an improved quantum particle swarm optimization (IQPSO) algorithm is proposed to optimize the parameters. At the same time, the improved PSO algorithm is used to adjust the noise in the unscented Kalman filter algorithm, which can effectively ensure that the number of particles will not decrease significantly and improve the accuracy of SOC estimation results. However, the Cholesky decomposition in the UKF algorithm will not work in the face of singular matrices. For this problem, square-root unscented Kalman filter algorithm is used in this article [31] instead of UKF algorithm based on Cholesky decomposition. We use singular decomposition instead of Cholesky decomposition in this article to solve the singular matrix problem. Hao et al. used the UPF algorithm to estimate SOC within the article but did not improve the UPF algorithm within the article [32]. Hong et al., within the article, improved the PF algorithm to predict the lifetime of lithium-ion batteries, verifying the wide range of uses of the PF algorithm [33]. Zhang et al., in their article, proposed the use of the AWCPF algorithm for SOC estimation of lithium-ion batteries, but only a first-order RC equivalent circuit model was used within the article, and the model accuracy was not high enough [34].

At present, the Thevenin model is often used to estimate SOC for ternary lithium-ion batteries, and the second-order resistance-capacitance (RC) model is an improvement of the Thevenin model [35–37]. In contrast, the second-order RC model can more accurately describe the operating characteristics of the battery. At the same time, compared with the offline parameter identification method, the online parameter identification method can better simulate the capacitance and internal resistance changes of the battery in the real working state [38–40]. Therefore, in this paper, an adaptive FFRLS method is used to identify on-line parameters of the improved Thevenin model, to improve the estimation accuracy of the improved unscented particle filter algorithm [41, 42].

In this paper, to accurately describe the operating state of ternary lithium-ion batteries under complex operating conditions, the estimation of the charge state of lithium-ion batteries is investigated, taking into account the dynamic accuracy of the characterization and the precise estimation of the battery state. Online parameter identification of the equivalent circuit model is performed by the AFFRLS algorithm, while SOC estimation is performed using the improved SDUPF algorithm. The improved SDUPF algorithm is optimized by using an adaptive weighted PSO algorithm. The PSO algorithm with adaptive weights can adjust the noise well, which is conducive to improving the accuracy of the algorithm, while the addition of the singular decomposition method can make the system more stable. Meanwhile, under multiple temperatures, the algorithm has good performance even under different operating conditions as confirmed in this paper.

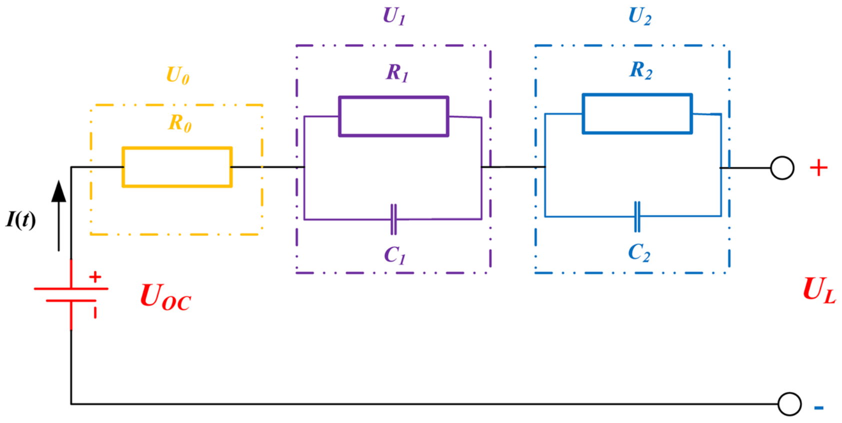
Equivalent circuit model and parameter identification method

Second-order RC equivalent circuit model

At present, the commonly used equivalent circuit models are the Thevenin model, second-order RC equivalent model, Rint model, PNGV model, and so on. Among them, the Thevenin equivalent circuit model has a simple structure, only needs a small number of parameters and a low computational cost to complete the modeling, and is widely used. In this paper, the second-order RC equivalent circuit model is used to add a set of RC circuits compared with the Thevenin equivalent circuit model [43, 44].

Although a set of RC circuits is added, the computational complexity is not significantly increased, and the working state of the battery can be better simulated in actual work and better estimation accuracy can be achieved. Among them, the second-order RC equivalent circuit model structure is shown in Fig. 1.

Fig. 1 Second-order RC equivalent circuit model



In Fig.1, U_{OC} represents the open circuit voltage, U_L represents the terminal voltage, and R_0 represents the ohmic internal resistance. The parallel links of R_1 and C_1 represent the electrochemical polarization process of the battery, while the parallel links of R_2 and C_2 represent the concentration difference polarization process of the battery. According to Kirchowski's law, the voltage and current expressions of the equivalent circuit can be obtained by analyzing the constructed second-order RC equivalent circuit model as shown in eq. (1).

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

Among them, the open circuit voltage can be characterized by the state variable SOC and the state space variable $x_k = [SOC_k, U_1, U_2]^T$ is selected, the equivalent circuit is discretized to obtain a discrete state space expression. As shown in eq. (2),

$$\begin{cases} x_{k+1} = A_k x_k + B_k I_k + W_k \\ U_{L,k+1} = \left[\frac{\partial U_{OC}}{\partial SOC} - 1 - 1\right] x_k - R_0 I_k + V_k \end{cases} \tag{2}$$

$$A_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{\frac{\Delta t}{\tau_1}} & 0 \\ 0 & 0 & e^{\frac{\Delta t}{\tau_2}} \end{bmatrix} \tag{3}$$

$$B_k = \begin{bmatrix} -\frac{\eta \Delta t}{Q_N} \\ R_1\left(1 - e^{\frac{T}{\tau_1}}\right) \\ R_2\left(1 - e^{\frac{T}{\tau_2}}\right) \end{bmatrix} \tag{4}$$

Here, Δt is the sampling interval, $\tau_1 = R_1C_1$, $\tau_2 = R_2C_2$, Q_N is the rated capacity of the cell, W_k is the state error, V_k is the measurement error, and is the zero-mean white noise of the covariance matrices Q and R, respectively, and η is the Coulomb efficiency.

Online parameter identification based on the AFFRLS algorithm

In this paper, ternary lithium batteries are selected for testing, the nominal capacity of the battery is 70 Ah, and the actual capacity is 68 Ah. Since the parameters in the model are greatly affected by the temperature, the experiments are considered to be conducted at two temperatures, 25 °C and 35 °C, and the experiments are carried out in a constant temperature box. Because the charge and discharge of lithium-ion battery is a complex dynamic process [45], and in the process of charge and discharge, the equivalent circuit model parameters of lithium-ion battery will change with the change of the depth of charge and discharge, and the experimental environment is easy to affect the reaction process in the battery. In the SOC estimation of the battery, the offline experimental data will be utilized to determine the values of each parameter er in the battery equivalent circuit model by function fitting.

However, when this method is used, absolute errors occur in the estimated results. Therefore, it is an effective method to improve SOC estimation accuracy to determine model parameters on-line and modify parameter values in real-time. Based on the second-order RC equivalent circuit model, the modified forgetting factor recursive least square method is used to determine the model parameters.

Here, the AFFRLS algorithm will be used to determine the online parameters of the equivalent circuit model. Compared with the RLS algorithm, the addition of the forgetting factor makes the algorithm avoid being influenced too much by the old data and avoids data saturation problems [46, 47]. Compared with the FFRLS algorithm, the adaptive forgetting factor is added here. Higher parameter estimation accuracy can be obtained by automatically adjusting the forgetting factor [48]. The Laplace equation for the Li-ion battery model is shown in eqs. (5) and (6).

$$E(s) - U(s) = I(S)\left(R + \frac{R_S}{1 + R_S C_S s} + \frac{R_P}{1 + R_P C_P s}\right) \tag{5}$$

$$G(s) = \left(R + \frac{R_S}{1 + R_S C_S s} + \frac{R_P}{1 + R_P C_P s}\right) = \frac{R\tau_s\tau_p s^2 + \left(R\tau_s + R\tau_p + R_p\tau_s + R_s\tau_p\right)s + R + R_p + R_s}{\tau_s\tau_p s^2 + \left(\tau_s + \tau_p\right)s + 1} = \frac{Rs^2 + \frac{1}{\tau_s\tau_p}\left(R\tau_s + R\tau_p + R_p\tau_s + R_s\tau_p\right)s + \frac{R+R_P+R_s}{\tau_s\tau_p}}{s^2 + \frac{\left(\tau_s+\tau_p\right)}{\tau_s\tau_p}s + \frac{1}{\tau_s\tau_p}} \tag{6}$$

Using a bilinear transformation for discretization, letting $s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$ yields the discretized transfer function as shown in eq. (7).

$$G\left(z^{-1}\right) = \frac{k_3 + k_4 z^{-1} + k_5 z^{-2}}{1 - k_1 z^{-1} - k_2 z^{-2}} \tag{7}$$

where the parameters included are those to be estimated, converting the equation into a difference equation yields eqs. (8), (9), and (10).

$$y_k = E_k - U_k = k_1 y_{k-1} + k_2 y_{k-2} + k_3 I_k + k_4 I_{k-1} + k_5 I_{k-2} \tag{8}$$

$$h_k = \left[y_{k-1}, y_{k-2}, I_k, I_{k-1}, I_{k-2}\right]^T \tag{9}$$

$$\theta = \left(k_1, k_2, k_3, k_4, k_5\right)^T \tag{10}$$

where I_k is the system input, y_k is the system output, and let the sensor sampling error at moment k be v_k, eq. (11) can be obtained.

$$y_k = h_k^T\theta + v_k \tag{11}$$

Here, extending h_k to N dimensions, giving eq. (12), while taking the generic function $J(\theta)$ gives eq. (13).

$$\begin{cases} h_k = \begin{bmatrix} y_2 & y_1 & I_3 & I_2 & I_1 \\ y_3 & y_2 & I_4 & I_3 & I_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{N+1} & y_N & I_{N+2} & I_{N+1} & I_N \end{bmatrix}^T \\ Y = \left[y_3 \; y_4 \; y_5 \; \cdots \; y_{N+2}\right]^T \\ v = \left[v_3 \; v_4 \; v_5 \; \cdots \; v_{N+2}\right]^T \end{cases} \tag{12}$$

$$J(\theta) = \sum_{i=1}^{N} (Y - h\theta)^2 = \sum_{i=1}^{N} \left(v_{i+2}\right)^2 \tag{13}$$

Since the principle of least squares is to minimize $J(\theta)$, find the extreme value of $J(\theta)$ and make eqs. (14) and (15).

$$\frac{\partial J(\theta)}{\partial(\theta)} = \frac{\partial}{\partial(\theta)}\left[(Y - h\theta)^T(Y - h\theta)\right] = 0 \tag{14}$$

$$\hat{\theta} = \left[h^T h\right]^{-1} h^T Y \tag{15}$$

Due to the "filter saturation" phenomenon of the least square method, with the increase of the number of algorithm iterations, the value of the gain K and P becomes smaller and smaller, which makes the algorithm's ability to correct the data weaker and weaker, and the degree of data saturation becomes larger and larger, and eventually, the error of parameter identification becomes larger and larger. The final analysis results show that the algorithm's ability to correct the data is getting weaker and weaker, and the saturation of the data is getting bigger and bigger. Therefore, in order to improve the accuracy of the parameter identification results, the least squares method with the forgetting factor is chosen to identify the parameters in this paper. In the process of recognition, the role of the forgetting factor is to make the long-running data less weighted and the latest observation data more weighted. In this paper, the forgetting factor $\lambda\,(0 < \lambda < 1)$ is introduced. The influence of old data will be diminished, while the feedback effect of new data will be enhanced. At the same time, the variability of λ will be realized on the basis of FFRLS. The forgetting factor is adjusted according to the square of the time-averaged estimate of the autocorrelation of the a priori and a posteriori errors. The improved recursive equation is shown in eq. (16).

$$\begin{cases} \hat{\theta}_{k+1} = \hat{\theta}_k + K_{k+1}\left[y_{k+1} - h_k^T\hat{\theta}_k\right] \\ K_{k+1} = P_k h_k\left[\lambda_k + h_k^T P_k h_k\right]^{-1} \\ P_{k+1} = \frac{1}{\lambda_k}\left[I - K_{k+1}h_k^T\right]P_k \\ \lambda_{k+1} = 1 - \frac{y_{k+1} - h_k^T\hat{\theta}_k}{1 + h_k^T P_k h_k \eta} \end{cases} \tag{16}$$

where $\hat{\theta}_k$ is the parameter estimate at moment k; y_{k+1} is the observed value at moment $k + 1$; the set of system samples vector is $h_k = [y_{k-1}, y_{k-2}, -u_k, -u_{k-1}, -u_{k-2}]^T$ and the $P_0 = aI$ where a tries to go to a larger value, the I is the unit matrix. The least squares mathematical form of the second-order RC cell model is shown in eq. (17). Let the time constant be as in eq. (18):

$$U_{OC} = U_L + \left(\frac{R_1}{R_1 C_1 S + 1} + \frac{R_2}{R_2 C_2 S + 1} + R_0 \right) I \tag{17}$$

$$\begin{cases} \tau_1 = R_1 C_1 \\ \tau_2 = R_2 C_2 \\ a = \tau_1 \tau_2 \\ b = \tau_1 + \tau_2 \\ c = R_0 + R_1 + R_2 \\ d = R_1 \tau_2 + R_2 \tau_1 + R_0 \left(\tau_1 + \tau_2 \right) \end{cases} \tag{18}$$

After discretizing eq. (17), k_1 to k_5 are introduced instead of the actual parameters for abstract expression, as shown in eq. (19). In eq. (19), $\theta = (k , k , k , k , k)^T$. After obtaining the parameter vector, the actual parameters can be obtained by eq. (18), as in eq. (20).

$$U_{OC,k} - U_{L,k} = k_1 \left[U_{L,k-1} - U_{OC,k-1} \right] + k_2 \left[U_{L,k-2} - U_{OC,k-2} \right] + k_3 I_k + k_4 I_{k-1} + k_5 I_{k-2} \tag{19}$$

$$\begin{cases} R_0 = k_5 / k_2 \\ R_1 = \left(\tau_1 c + \tau_2 R_0 - d \right) / \left(\tau_1 - \tau_2 \right) \\ R_2 = c - R_1 - R_0 \\ C_1 = \tau_1 / R_1 \\ C_2 = \tau_2 / R_2 \end{cases} \tag{20}$$

From the above expression, the basic mathematics of the AFFRLS algorithm can be understood in detail. Table 1 details the basic process of the AFFRLS algorithm, from which the computational steps of the AFFRLS algorithm can be clearly understood.

Table 1 Process of Adaptive forgetting factor least squares algorithm

Step 1, Parameter initialization.

Step 2, Input current and voltage data, that is, data vector $h(k) = [\, y(k-1), y(k-2), I(k), I(k-1), I(k-2)]^T$.

Step 3, Setting parameter groups, $\theta = (k_1, k_2, k_3, k_4, k_5)^T$ and forgetting factor λ.

Step 4, Calculate the least squares gain matrix K, calculate the least squares covariance matrix P.

Step 5, Updating the forgetting factor λ and parameter vector.

Step 6, Find the model parameters :R_0, R_1, R_2, C_1, C_2.

Theoretical analysis of SOC estimation methods

Particle swarm optimization algorithm

The basic definition of the particle swarm optimization (PSO) algorithm is as follows: in the particle swarm algorithm, each particle is looking for its best position and speed during flight, and there will be a fitness value after each flight, which is determined by the objective function [49]. When a particle finds its current best position, the position is called individual extreme value (*Pbest*). This position is called the global optimal solution (*Gbest*).

Suppose there is a population X of n particles in a D-dimensional space, where $X = (x_1, x_2, L, x_n)$. Let the ith particle in the population correspond to a vector $X_i = (x_{i1}, x_{i2}, L, x_{il})^T$ in the D-dimensional space. This vector is the position of the ith particle in D-dimensional space and the fitness value of the particle's position x_i needs to be calculated according to the objective function. Let the vector corresponding to the velocity of the ith particle be $V_i = (v_{i1}, v_{i2}, L, v_{il})^T$, then the vector of the extreme value of the individual particle can be represented as $P_i = (p_{i1}, p_{i2}, L, p_{il})^T$, and the vector of the extreme value of the population X can be represented as Pg = (pg1, pg2, L, pgl)T. The individual extremum and the population extremum are calculated by the following formula for the speed and position of the updating particle. As shown in eqs. (21) and (22).

$$v_{id}^{k+1}(t+1) = v_{id}^{k}(t) + c_1 r_1 \left(p_{id}^{k}(t) - x_{id}^{k}(t)\right) + c_2 r_2 \left(p_{gd}^{k}(t) - x_{gd}^{k}(t)\right) \qquad (21)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad (22)$$

Where, v_{id} is the current velocity; $d = 1, 2, L, D$; $i = 1, 2, L, n;$ The k is the number of iterations; The c_1, c_2 denote the acceleration factor and $c_1 > 0$, $c_2 > 0$; The r_1, r_2 denote the stochastic function with the value range of [0, 1]. The motion of the particle in the search space is shown in Fig. 2.

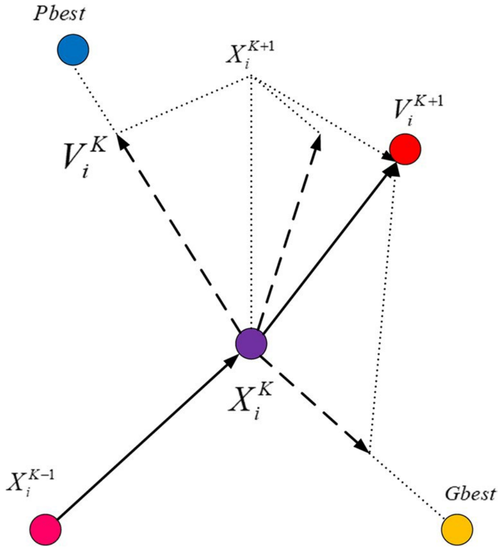


Fig. 2 Motion of a particle in a PSO

Variable weights shrinkage particle swarm optimization algorithm

The basic particle swarm algorithm is simple in principle and convenient in operation, but the optimization process is more dependent on control parameters such as population size, iteration number, and acceleration factor, so the convergence is slower during the calculation process, resulting in a limited searchability of the particle swarm and easy to fall into the local minimum.

In this paper, the basic particle swarm algorithm is improved by introducing the inertia weights while adding the contraction factor, to ensure that the particle swarm can better determine the search ratio of local optimum and global optimum during the optimization process and to find the balance point between the two to improve the search performance of the particle swarm, and the update formula of the particle velocity after the introduction of the inertia weights is shown in equations (23) and (24):

$$v_{\mathrm{id}}^{k+1}(t+1) = w v_{\mathrm{id}}^{k}(t) + c_1 r_1 \left(p_{\mathrm{id}}^{k}(t) - x_{\mathrm{id}}^{k}(t)\right) + c_2 r_2 \left(p_{gd}^{k}(t) - x_{gd}^{k}(t)\right) \qquad (23)$$

$$v_{\mathrm{id}}(t+1) = \begin{cases} -v_{d,\max}, v_{\mathrm{id}}(t+1) < -v_{d,\max} \\ v_{d,\max}, v_{id}(t+1) > -v_{d,\max} \end{cases} \qquad (24)$$

where w is the inertia weight and $v_{d,\ \max}$ is the maximum velocity of the particle swarm when it flies in the D-dimensional space. When the inertia weight increases, the global search performance of the particle swarm is enhanced and the local search performance is weakened; when the inertia weight decreases, the local search performance of the particle swarm is enhanced and the global search performance is weakened [50]. Reasonable selection of inertia weights can make the two balanced when the inertia weight is 1, that is, it becomes the basic particle swarm algorithm, which does not have the balance.

When the standard PSO algorithm is used for simulation, the inertia weight w is usually reduced linearly from 0.9 to 0.4, because the search range of the particle swarm is larger at the beginning of the iteration, and w is larger to lock the approximate location of the optimal solution quickly, and as the number of iterations increases, w is reduced so that the particle's speed becomes smaller, and a local search is carried out to look for the optimal solution within the approximate range of the optimal solution, which can show that the introduction of w is able to significantly improve the convergence speed of the algorithm. Where w is calculated as shown in eq. (25):

$$w(i) = w_{\max} - \frac{k}{\mathrm{gen}}\left(w_{\max} - w_{\min}\right) \tag{25}$$

where $w_{\max}$ is the maximum value of the inertia weights, $w_{\min}$ is the minimum value of the inertia weights, k is the current iteration number, and gen is the total number of iterations.

The addition of inertia weights improves the search performance of the particle swarm, at the same time, in order to eliminate the limitation of the particle velocity boundary and accelerate the convergence of the PSO algorithm, this paper introduces a contraction factor into the basic PSO algorithm, which can change the value of the acceleration factor c_1, c_2, thus changing the velocity of the particles. The updated equations for the particle velocity after the introduction of the shrinkage factor are shown in eqs. (26), (27), and (28):

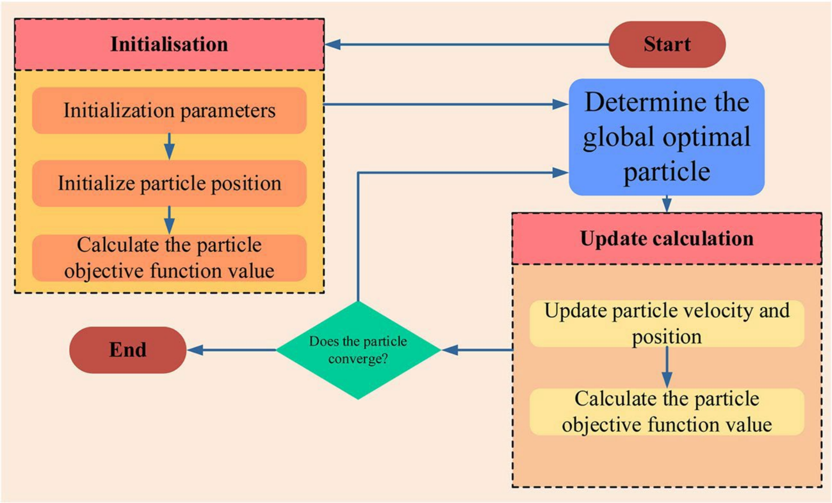
$$v_{\mathrm{id}}^{k+1}(t+1) = \vartheta\left(wv_{\mathrm{id}}^{k}(t) + c_1 r_1\left(p_{\mathrm{id}}^{k}(t) - x_{\mathrm{id}}^{k}(t)\right) + c_2 r_2\left(p_{gd}^{k}(t) - x_{gd}^{k}(t)\right)\right) \tag{26}$$

$$\vartheta = \frac{2K}{\left|\, 2 - \varnothing - \sqrt{\varnothing^2 - 4\varnothing}\,\right|} \tag{27}$$

$$K = 1 - \frac{k}{\mathrm{gen}} \tag{28}$$

where ϑ is the contraction factor, $\varnothing$ is the total acceleration factor with $\varnothing > 4$, and K is the contraction factor. The contraction coefficient is used to control the global search ability and local search ability of the particle swarm, K tends to 1 will make the particle speed larger, resulting in the expansion of the search range, a large number of global searches will make the particle swarm algorithm converge slowly, K tends to 0 will reduce the particle search speed, reduce the search range, and a large number of local searches will result in the convergence of the particle swarm algorithm too fast. The shrinkage factor K usually takes a fixed value, in this paper, in order to protect the particle swarm's ability to carry out global search in the early stage as well as local search in the later stage, so K is associated with the number of iterations. The algorithmic flow of PSO is shown in Fig. 3.

Fig. 3 The flowchart of the IPSO algorithm


Start
Initialisation
Initialization parameters
Initialize particle position
Calculate the particle objective function value
Determine the global optimal particle
Update calculation
Update particle velocity and position
Calculate the particle objective function value
Does the particle converge?
End

Unscented particle filtering algorithm

Currently, the main filtering algorithms used to solve non-linear system models are extended Kalman filtering (EKF), unscented Kalman filtering (UKF), particle filtering (PF), etc. The main idea of EKF is to approximate the nonlinear function linearly by ignoring or approximating the higher-order terms to solve the filtering problem. The main idea of the EKF algorithm is to linearize the approximation of nonlinear functions by ignoring or approximating the higher- order terms in order to solve the filtering problem of nonlinear systems. However, when the nonlinearity of the system is serious, ignoring the higher-order terms of the Taylor expansion will lead to a large linearization error, which results in an increase in the filtering error or even dispersion of the EKF. Secondly, the Jacobian matrix is computationally intensive. To overcome the shortcomings of EKF, a UKF algorithm based on unscented transform (UT) is proposed, which uses deterministic sampling to approximate the probability distribution of a nonlinear function, avoids the problem of computationally heavy computation for solving the Jacobian matrices, and the approximation accuracy of the Gaussian distribution can be up to the third order. However, for non-Gaussian and strongly nonlinear system models, it is difficult to guarantee filtering accuracy. PF approximates the probability density function by finding a set of random samples propagating in the state space and replaces the integral operation with the sample mean to obtain the minimum variance estimation of the state. PF is suitable for state estimation of nonlinear non-Gaussian systems, especially for the filtering problem of strongly nonlinear systems, which has a unique advantage and gets rid of the constraint that the random quantities must satisfy the Gaussian distribution when solving the nonlinear filtering problem. However, PF suffers from the problems of difficulty in selecting the important density function and lack of particles [51].

The particle filtering method implements cyclic Bayesian filtering through a nonparametric Monte Carlo simulation method. It applies to any nonlinear system that can be described by a state space model. Due to its nonparametric nature, it is free from the restriction that random quantities must satisfy Gaussian distribution when solving nonlinear filtering problems. Compared with the Gaussian model, particle filtering is more widely used and has better modeling capability. The core idea is to use some scattered random sampling points to approximate the posterior probability density function (PDF) of the state. Then the sampling average is utilized instead of integral calculation. Finally, a minimum variance estimate for the final condition can be obtained. The UPF algorithm utilizes the most recent observations to obtain the PDF and uses the UKF algorithm to generate the recommended distribution.

The PF algorithm in the UPF algorithm is optimized by taking advantage of the unscented transform algorithm. The UKF algorithm can theoretically compute the third-order squared difference with accuracy, which can be obtained by comparing the EKF algorithm based on the first-order Taylor expansion. The algorithm has high accuracy and the calculation results are valid.

Singular value decomposition unscented particle filtering algorithm based on improved particle swarm optimization algorithm

As the particles collected by the particle filtering algorithm have a great impact on the accuracy of the calculation, the main impact is divided into: 1. the imbalance phenomenon brought about by the different weights of the particles; 2. if the number of particles with small weights is large, a large amount of computational time will be used to compute the smaller particles with small weights, which greatly reduces the computational efficiency and accuracy, i.e., there exists a degradation phenomenon of the particle filtering algorithm.

Although resampling can reduce the particle degradation phenomenon and improve the calculation accuracy, it also increases the calculation amount of the algorithm, and the calculation efficiency of the algorithm decreases, at the same time, too many times of resampling will make a large number of particles eliminated, and the particle library greatly reduced, and seriously even appeared in the particle library depletion. Therefore, in order to ensure that the number of effective particles will not be depleted because of the massive reduction caused by resampling, the application can be used to set a reasonable density function of the proposal, so that the likelihood function and the a priori distribution of the basic match, the particle set of samples to a greater extent are transferred to the region covered by the likelihood function. In this paper, the improved particle swarm algorithm optimizes the UKF algorithm of singular value decomposition to generate particles.

In this paper, the real-time tracking of changes in the statistical properties of noise in the UKF algorithm is realized by using the neo-interest sequence and the modified PSO algorithm to track the observation noise and process noise, respectively. The neo-interest e_k is defined as shown in equation (29):

$$e_k = y_k - g\left(\hat{x}_k, u_k\right) \qquad (29)$$

Also the real-time estimated covariance based on the new interest $\hat{H}_k$ and the observation noise covariance Rk can be estimated online by eqs. (30) and (31):

$$\hat{H}_k = e_k e_k^T \tag{30}$$

$$\hat{R}_k = \left\{ \hat{H}_k + \sum_{i=0}^{2n} w_c^{(i)} \left[\chi_{i,k|k-1} - y_k \right] \left[\chi_{i,k|k-1} - y_k \right]^T \right\} / 2 \tag{31}$$

From the above equation, the estimation of the observation noise covariance is updated based on the observations of the current step and the information from the previous iterations. In this case, if the process noise Q_k does not change, the above estimation of the covariance of the observation noise based on the hibernation sequence is accurate; however, when the two types of noise change at the same time, other methods are needed to make the estimation separately. In this paper, based on the online estimation of the observation noise using the new interest series, the process noise Q_k is tracked in real time by the modified PSO algorithm. The key point of combining the UKF filter and the modified PSO algorithm is to choose a reasonable fitness function. Considering that the covariance matrix of the new hibernation contains two kinds of noise information at the same time, the fitness function is defined as shown in equation (32) here:

$$F = \frac{tr\left(\hat{H}_k\right)}{tr\left\{ 2\hat{R}_k - \sum_{i=0}^{2n} w_c^{(i)} \left[\chi_{i,k|k-1} - y_k \right] \left[\chi_{i,k|k-1} - y_k \right]^T \right\}} \tag{32}$$

The UKF algorithm in this paper uses the improved PSO algorithm to adjust the noise based on UKF, which can improve the accuracy and real-time performance in the estimation process through the real-time update of the noise matrix. Whether in the UKF algorithm or the improved UKF algorithm, it uses the unscented transform to deal with the nonlinear problem, but the UT transform has a limitation, when the error covariance is singular then it will lead to the UT transform can't be carried out, which will result in the instability of the system. To address this problem, this paper adopts the UKF algorithm based on SVD decomposition as a way to realize the estimation of SOC, and the SVD decomposition is shown in equation (33).

$$P = U\Lambda G^T = U \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} G^T \tag{33}$$

P is a matrix of order $m \times n$, $m \geq n$, where U is a matrix of order $m \times m$, Λ is a semi-positive definite matrix of order $m \times n$, G is a matrix of order $n \times n$, where the matrix $S = \text{diag}(s_1, s_2, s_3, \ldots, s_r)$ within Λ is a diagonal matrix consisting of the singular values of the matrix P and r is the rank of the matrix P.

Similarly, in using the SDUKF algorithm for achieving accurate estimation of SOC, the state equation and measurement equation of the lithium battery are similar to eq, (2). Here, for easy distinction, let uk = I_k, $y_k = U_{L,k}$. Then, the new equation of state and measurement equation can be obtained as shown in eq, (34). Among them, W_k is the state error, V_k is the measurement error.

$$\begin{cases} x_{k+1} = f\left(x_k, u_k\right) + W_k = A_k x_k + B_k u_k + W_k \\ y_{k+1} = g\left(x_k, u_k\right) + V_k = \left[\frac{\partial U_{OC}}{\partial SOC} \; -1 \; -1 \right] \begin{bmatrix} SOC_k \\ U_{1,k} \\ U_{2,k} \end{bmatrix} - IR_0 + V_k \end{cases} \tag{34}$$

After describing the IPSO-SDUKF algorithm in detail, this paper utilizes the IPSO-SDUKF algorithm to improve the PF algorithm by implementing the sampling of the mean and variance of each particle it produces. The detailed principle step flow is shown in Fig. 4. In Fig. 4, P denotes the covariance matrix, w_j^m, w_j^c are the weights of the mean and variance, α is the scale parameter, usually 1e^{-4} $\leq \alpha \leq 1$, and λ is the scale parameter, $\lambda = 3 \times \alpha^2 - n_a$, $q\left(\overline{X}_k^i | X_{0:k-1}^i, Z_{1:k}\right)$ stands for the proposed density function, W_k^i is the particle weights, $\overline{X}_k$ represents the estimated value.

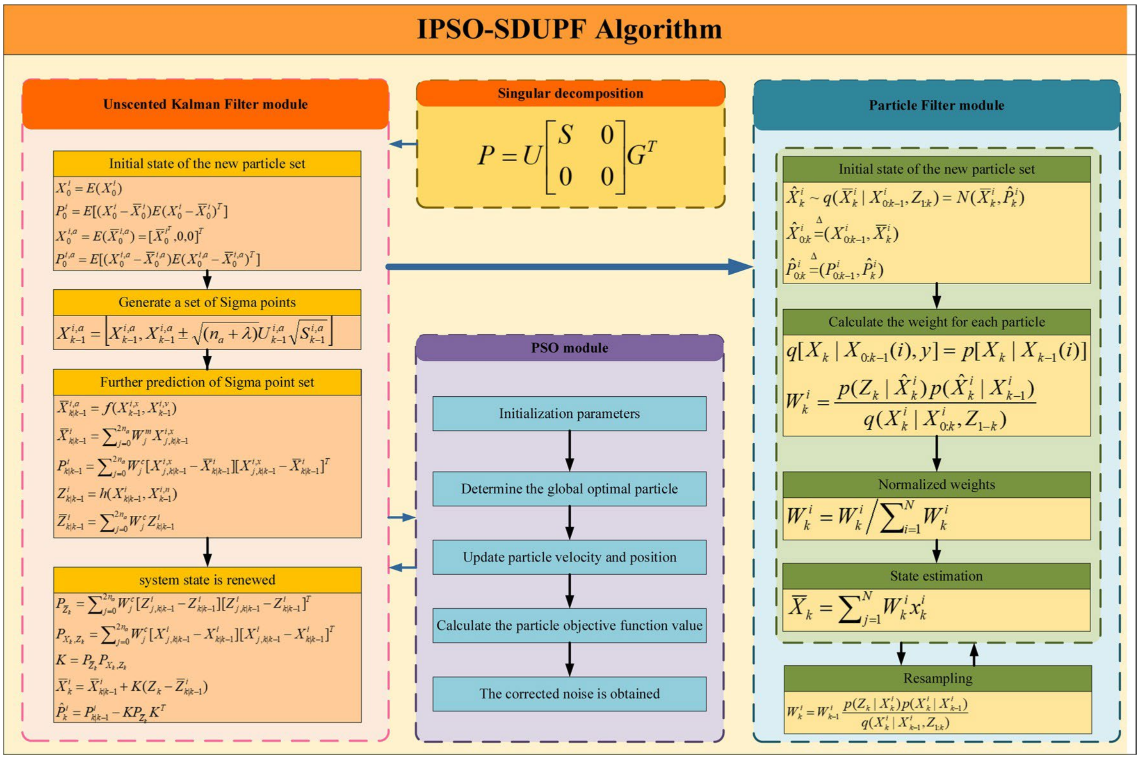

IPSO-SDUPF Algorithm
Unscented Kalman Filter module
Initial state of the new particle set
$X_0^i = E(X_0^i)$
$P_0^i = E[(X_0^i - \bar{X}_0^i)E(X_0^i - \bar{X}_0^i)^T]$
$X_0^{i,a} = E(\bar{X}_0^{i,a}) = [\bar{X}_0^{i^T}, 0, 0]^T$
$P_0^{i,a} = E[(X_0^{i,a} - \bar{X}_0^{i,a})E(X_0^{i,a} - \bar{X}_0^{i,a})^T]$
Generate a set of Sigma points
$X_{k-1}^{i,a} = \left[X_{k-1}^{i,a}, X_{k-1}^{i,a} \pm \sqrt{(n_a+\lambda)}U_{k-1}^{i,a}\sqrt{S_{k-1}^{i,a}}\right]$
Further prediction of Sigma point set
$\bar{X}_{k|k-1}^{i,a} = f(X_{k-1}^{i,x}, X_{k-1}^{i,v})$
$\bar{X}_{k|k-1}^i = \sum_{j=0}^{2n_a} W_j^m X_{j,k|k-1}^{i,x}$
$P_{k|k-1}^i = \sum_{j=0}^{2n_a} W_j^c [X_{j,k|k-1}^{i,x} - \bar{X}_{k|k-1}^i][X_{j,k|k-1}^{i,x} - \bar{X}_{k|k-1}^i]^T$
$Z_{k|k-1}^i = h(X_{k|k-1}^i, X_{k-1}^{i,n})$
$\bar{Z}_{k|k-1}^i = \sum_{j=0}^{2n_a} W_j^c Z_{k|k-1}^i$
system state is renewed
$P_{\bar{Z}_k} = \sum_{j=0}^{2n_a} W_j^c [Z_{j,k|k-1}^i - Z_{k|k-1}^i][Z_{j,k|k-1}^i - Z_{k|k-1}^i]^T$
$P_{X_k,Z_k} = \sum_{j=0}^{2n_a} W_j^c [X_{j,k|k-1}^i - X_{k|k-1}^i][X_{j,k|k-1}^i - X_{k|k-1}^i]^T$
$K = P_{\bar{Z}_k} P_{X_k,Z_k}$
$\bar{X}_k^i = \bar{X}_{k|k-1}^i + K(Z_k - \bar{Z}_{k|k-1}^i)$
$\hat{P}_k^i = P_{k|k-1}^i - KP_{\bar{Z}_k}K^T$
Singular decomposition
$P = U\begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}G^T$
PSO module
Initialization parameters
Determine the global optimal particle
Update particle velocity and position
Calculate the particle objective function value
The corrected noise is obtained
Particle Filter module
Initial state of the new particle set
$\hat{X}_k^i \sim q(\bar{X}_k^i \mid X_{0:k-1}^i, Z_{1:k}) = N(\bar{X}_k^i, \hat{P}_k^i)$
$\hat{X}_{0:k}^i \overset{\Delta}{=} (X_{0:k-1}^i, \bar{X}_k^i)$
$\hat{P}_{0:k}^i \overset{\Delta}{=} (P_{0:k-1}^i, \hat{P}_k^i)$
Calculate the weight for each particle
$q[X_k \mid X_{0:k-1}(i), y] = p[X_k \mid X_{k-1}(i)]$
$W_k^i = \dfrac{p(Z_k \mid \hat{X}_k^i)p(\hat{X}_k^i \mid X_{k-1}^i)}{q(X_k^i \mid X_{0:k}^i, Z_{1-k})}$
Normalized weights
$W_k^i = W_k^i \Big/ \sum_{i=1}^N W_k^i$
State estimation
$\bar{X}_k = \sum_{j=1}^N W_k^i x_k^i$
Resampling
$W_k^i = W_{k-1}^i \dfrac{p(Z_k \mid X_k^i)p(X_k^i \mid X_{k-1}^i)}{q(X_k^i \mid X_{k-1}^i, Z_{1:k})}$

Fig. 4 IPSO-SDUPF algorithm flow chart

Online SOC estimation based on the AFFRLS-IPSO-SDUPF algorithm

In this paper, based on the state equations and observation equations of lithium-ion batteries, the joint estimation algorithm of SOC for lithium-ion batteries based on online parameter identification and IPSO-SDUPF is realized by combining the AFFRLS algorithm and IPSO-SDUPF algorithm. It also reduces the impact on the SOC estimation accuracy due to the fixation of model parameters in the offline parameter identification method and thus the SOC estimation accuracy, in which the flow chart of the SOC estimation algorithm for lithium-ion batteries, as shown in Fig. 5.

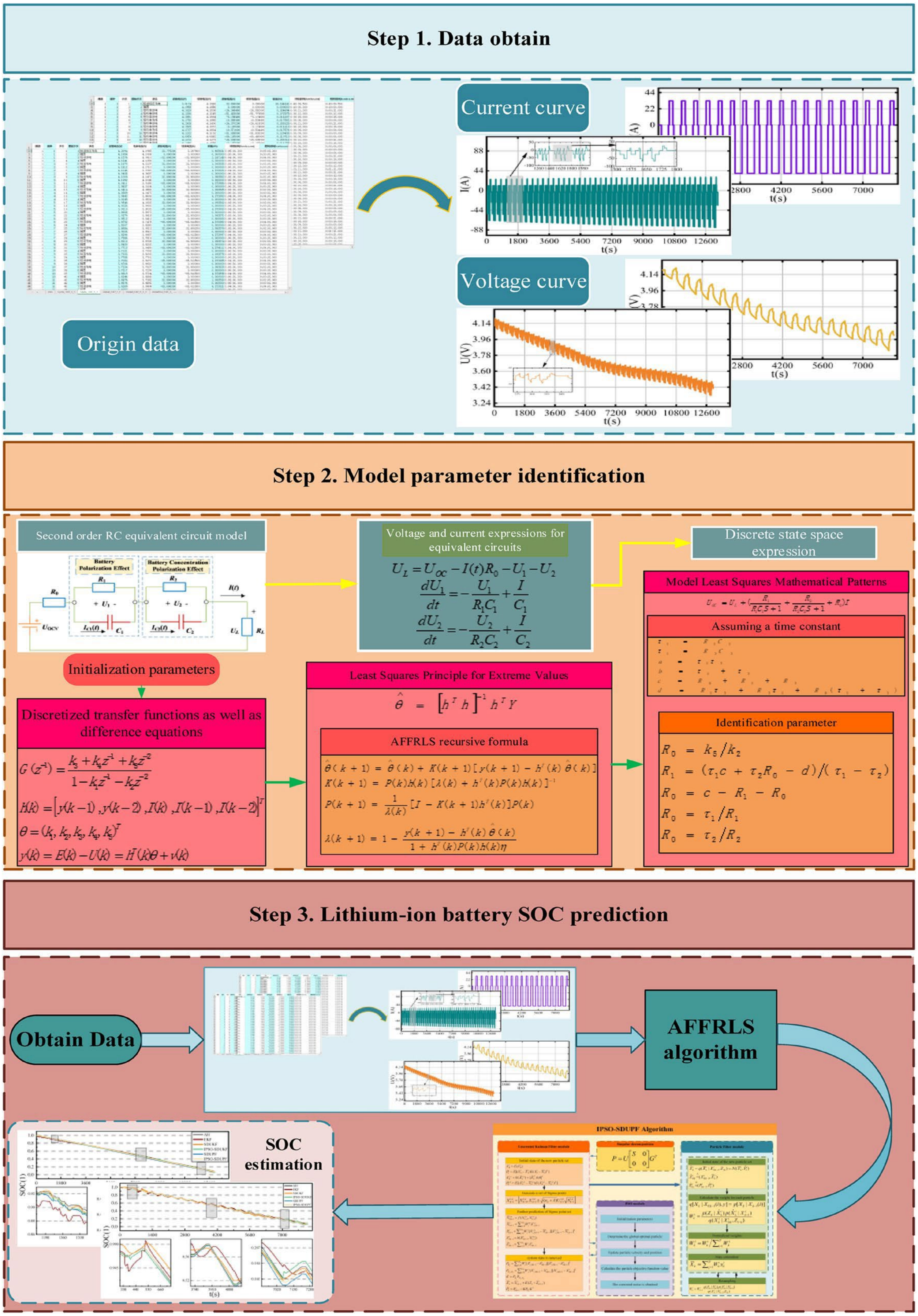

Step 1. Data obtain
Origin data
Current curve
Voltage curve
I(A)
U(V)
t(s)
Step 2. Model parameter identification
Second order RC equivalent circuit model
Battery Polarization Effect
Battery Concentration Polarization Effect
R_0
R_1
R_2
U_{OCV}
U_1
U_2
C_1
C_2
$I(t)$
U_L
R_L
Voltage and current expressions for equivalent circuits
$U_L = U_{oc} - I(t)R_0 - U_1 - U_2$
$\frac{dU_1}{dt} = -\frac{U_1}{R_1C_1} + \frac{I}{C_1}$
$\frac{dU_2}{dt} = -\frac{U_2}{R_2C_2} + \frac{I}{C_2}$
Discrete state space expression
Model Least Squares Mathematical Patterns
$U_{oc} = U_L + (\frac{R_1}{R_1C_1S+1} + \frac{R_2}{R_2C_2S+1} + R_0)I$
Assuming a time constant
$\tau_1 = R_1C_1$
$\tau_2 = R_2C_2$
$a = \tau_1\tau_2$
$b = \tau_1 + \tau_2$
$c = R_0 + R_1 + R_2$
$d = R_1\tau_2 + R_2\tau_1 + R_0(\tau_1 + \tau_2)$
Initialization parameters
Discretized transfer functions as well as difference equations
$G(z^{-1}) = \frac{k_3 + k_4z^{-1} + k_5z^{-2}}{1 - k_1z^{-1} - k_2z^{-2}}$
$H(k) = [y(k-1), y(k-2), I(k), I(k-1), I(k-2)]^T$
$\theta = (k_1, k_2, k_3, k_4, k_5)^T$
$y(k) = E(k) - U(k) = H^T(k)\theta + v(k)$
Least Squares Principle for Extreme Values
$\hat{\theta} = [h^T h]^{-1} h^T Y$
AFFRLS recursive formula
$\hat{\theta}(k+1) = \hat{\theta}(k) + K(k+1)[y(k+1) - h^T(k)\hat{\theta}(k)]$
$K(k+1) = P(k)H(k)[\lambda(k) + h^T(k)P(k)H(k)]^{-1}$
$P(k+1) = \frac{1}{\lambda(k)}[I - K(k+1)h^T(k)]P(k)$
$\lambda(k+1) = 1 - \frac{y(k+1) - h^T(k)\hat{\theta}(k)}{1 + h^T(k)P(k)h(k)\eta}$
Identification parameter
$R_0 = k_5/k_2$
$R_1 = (\tau_1c + \tau_2R_0 - d)/(\tau_1 - \tau_2)$
$R_0 = c - R_1 - R_0$
$R_0 = \tau_1/R_1$
$R_0 = \tau_2/R_2$
Step 3. Lithium-ion battery SOC prediction
Obtain Data
AFFRLS algorithm
IPSO-SDUPF Algorithm
SOC estimation
SOC(1)
t(s)

Fig. 5 Flowchart of SOC estimation for AFFRLS-IPSO-SDUPF

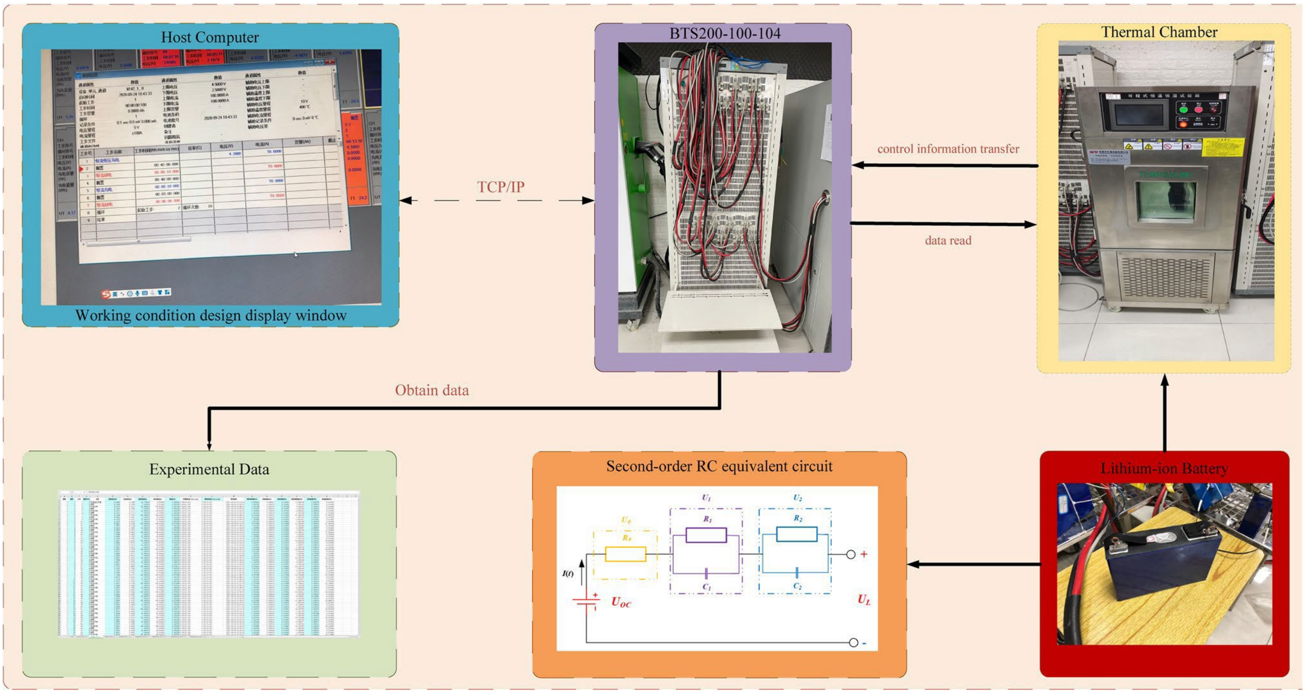


Fig. 6 Experimental platform

Validation and analysis of experimental results

Experimental platform

In this paper, a Li-ion ternary battery with a nominal capacity of 70Ah and an actual capacity of 68Ah was selected for testing. The experimental platform is shown in Fig. 6. Among them, we simulate the experimental conditions, simulating the BBDST condition and DST condition, respectively. We will conduct the experiments on the battery in a constant temperature box, and to avoid the variation of model parameters with temperature, the tests are conducted at 25 °C and 35 °C, respectively.

Parameter identification analysis

Currently, the application scenarios of lithium-ion batteries, as well as the operating conditions are very complex, and the variation of the operating current is large. In this paper, the joint algorithms based on AFFRLS and IPSO-SDUPF were tested using the Beijing Bus Dynamic Stress Test (BBDST) operating conditions and Dynamic Stress Test (DST) operating conditions at 25 °C ambient temperature conditions as well as at 35 °C high-temperature conditions, while ensuring a constant temperature environment. where the DST working condition is obtained by simplifying the U.S. Federal Urban Driving Schedule (FUDS). And the BBDST working condition is the Beijing Bus Dynamic Stress Test working condition. The experiments of the lithium-ion battery in these two conditions can be a good test of the accuracy of the algorithm. For the forgetting factor, the difference between FFRLS and AFFRLS algorithms lies in whether the forgetting factor can be adjusted adaptively or not. The FFRLS algorithm has a weak tracking ability when the current and voltage change is large due to the limitation of the forgetting factor during the online parameter identification, whereas the AFFRLS algorithm, which has a variable forgetting factor, can correct the forgetting factor automatically, and has a strong tracking ability. The following is an example to compare the tracking ability of FFRLS and AFFRLS on the real voltage curve with BBDST working conditions. When the temperature is set to 25 °C, as shown in Figs. 7, 8, and 9.

In Fig. 7, the forgetting factor λ was set to 0.99. At this point, it can be seen that both the AFFRLS algorithm and the FFRLS algorithm can track the real voltage better, but the FFRLS algorithm will still lead to a larger error at some moments due to the fast voltage change. In Fig. 8, the forgetting factor λ was set to 0.98. The FFRLS algorithm explicit error starts to increase due to the fact that the forgetting factor value at this time may differ from the optimal value, but since the AFFRLS algorithm can automatically adjust the forgetting factor value, adjusting the forgetting factor value to be close to the optimal value, the change of the initial value of the forgetting factor cannot have a big impact on the AFFRLS algorithm. Figure 9 is divided into two parts (a) and (b) and shows us the error curves of the FFRLS algorithm as well as the AFFRLS algorithm for the two cases of $\lambda = 0.99$ and $\lambda = 0.98$, respectively, and it can be seen that the maximum error of the FFRLS algorithm becomes bigger at $\lambda = 0.98$, which is not capable of estimating the real voltage well. Through the above comparison, it can be confirmed that the AFFRLS algorithm has better stability compared with the FFRLS algorithm, so this paper chooses to use the AFFRLS algorithm as the parameter identification method of the equivalent circuit model.

Fig. 7 Comparison of voltage at BBDST operating conditions when $\lambda = 0.99$

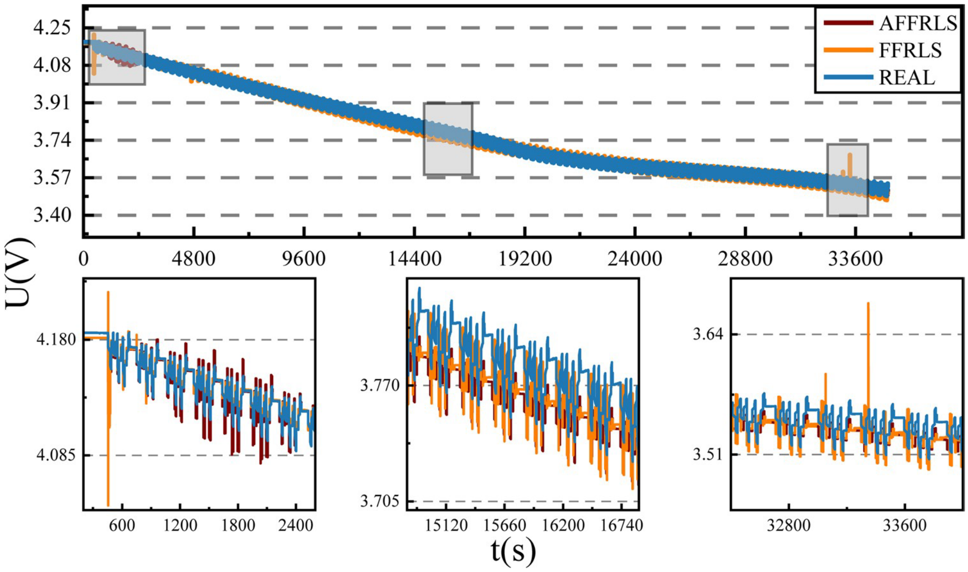
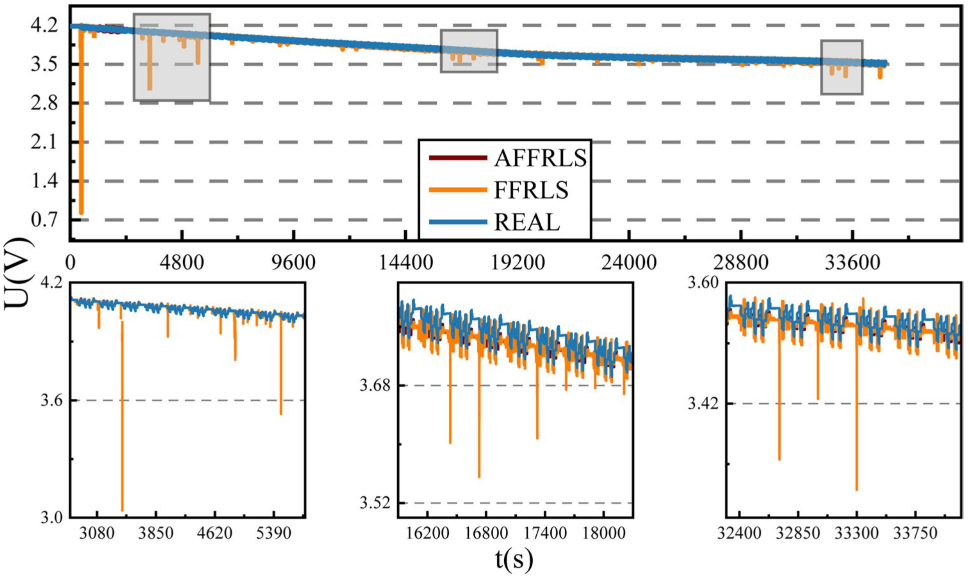
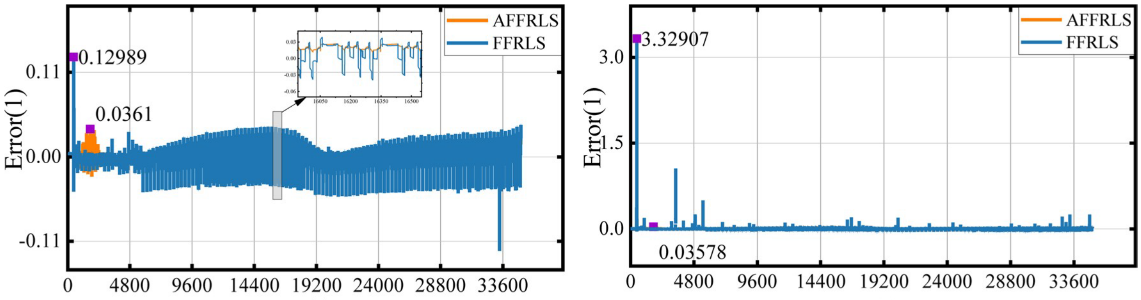


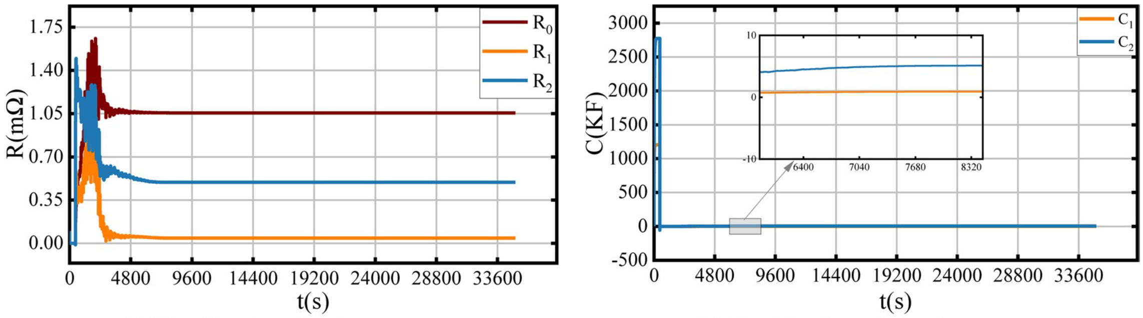
Fig. 8 Comparison of voltage at BBDST operating conditions when $\lambda = 0.98$





(a) $\lambda = 0.99$ (b) $\lambda = 0.98$

Fig. 9 Comparison of error measurements at BBDST operating conditions



(a) Identification of resistance parameters (b) Identification of capacitance parameters

Fig. 10 Resistance and capacitance identification results under BBDST operating conditions

As shown in Fig. 10, in part (a) are our real-time estimates of the three resistor values for the equivalent circuit model, and in part (b) are our real-time estimates of C1 and C2 within the equivalent circuit.

Due to the large influence of temperature on lithium batteries, the equivalent circuit model parameter identification for lithium batteries under BBDST condition is set here at 35 °C constant temperature. Different forgetting factor values are also set to validate the AFFRLS algorithm and FFRLS algorithm. As shown in Figs. 11, 12, and 13.

Here, all the parameters are the same as at 25 °C except for the temperature. It can be seen that the AFFRLS algorithm still works well for accurate estimation of the equivalent circuit model. Meanwhile, under different forget- ting factors, the forgetting factor can be adjusted adaptively, which is an obvious improvement compared with the FFRLS algorithm. Next, in Fig. 14, we will show the parameter identification results.

Fig. 11 Comparison of voltage at BBDST operating conditions when $\lambda = 0.99$

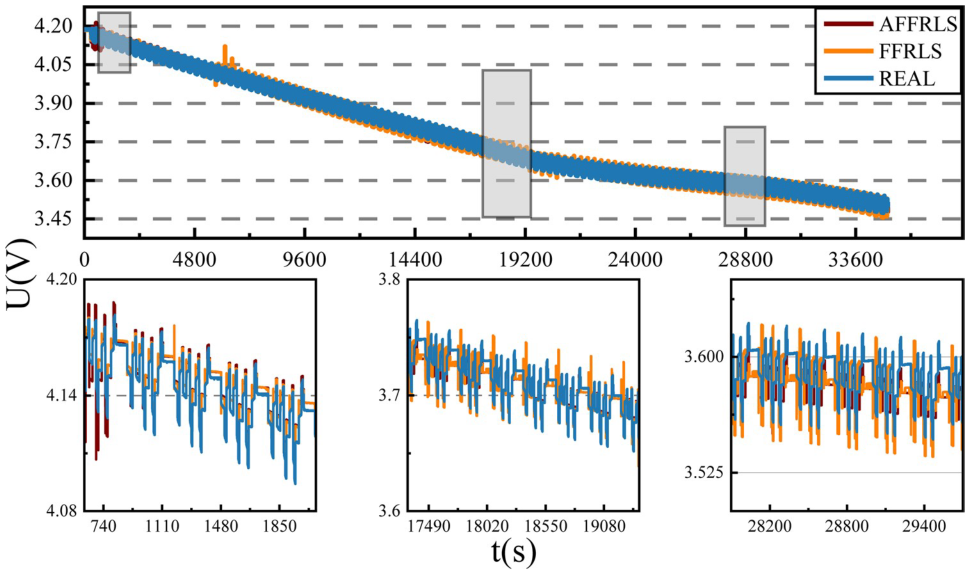
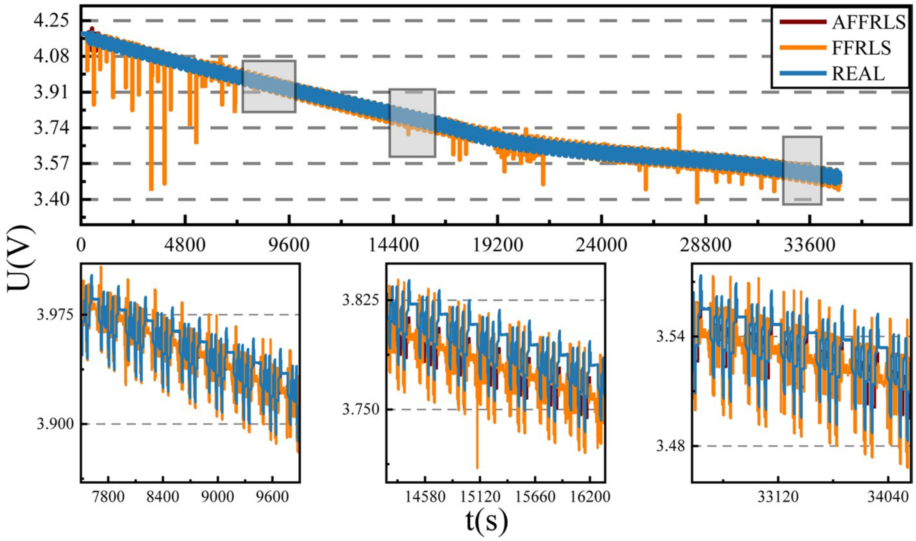
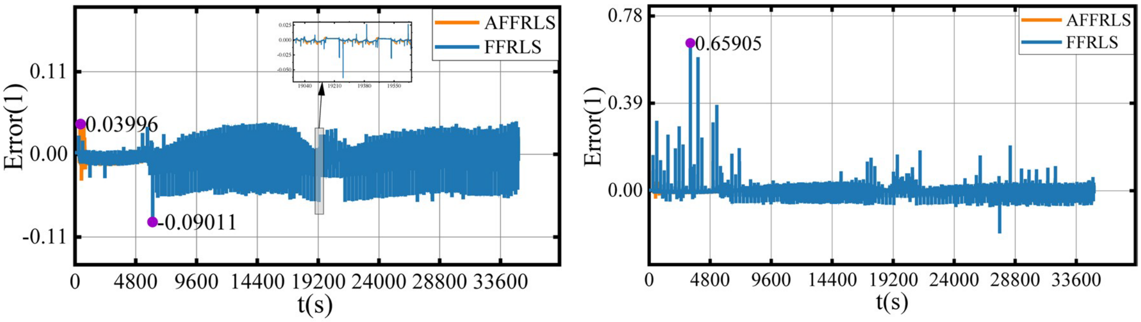


Fig. 12 Comparison of voltage at BBDST operating conditions when $\lambda = 0.98$

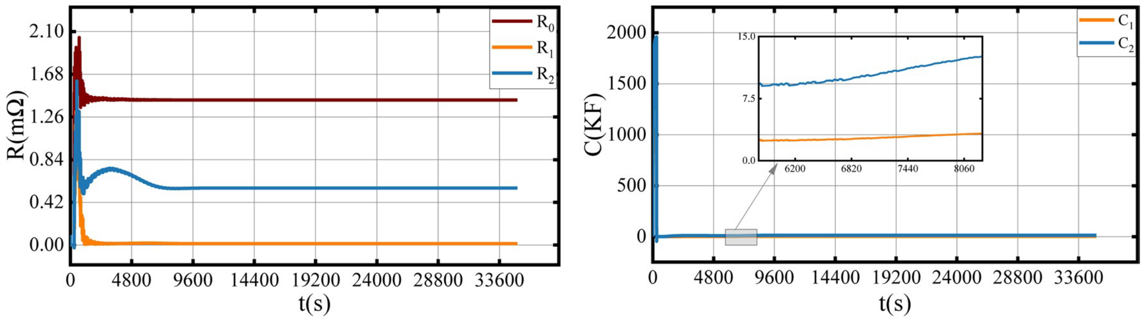




(a) $\lambda = 0.99$

(b) $\lambda = 0.98$

Fig. 13 Comparison of error measurements at BBDST operating conditions



(a) Identification of resistance parameters

(b) Identification of capacitance parameters

Fig. 14 Resistance and capacitance identification results under BBDST operating conditions

In this paper, mean absolute error (MAE) and root mean squared error (RMSE) are used as our evaluation metrics for the estimation accuracy of the equivalent circuit model using AFFRLS as well as FFRLS algorithms.

MAE and RMSE are calculated as shown in Eq. 35 and Eq. 36, respectively. $\hat{x_k}$ represents the estimated value, and x_k represents the true value; MAE and RMSE are the two most common metrics regarding continuous variables.

$$\mathrm{MAE} = \frac{1}{N}\sum\nolimits_{k=1}^{N} \left|\hat{x}_k - x_k\right| \tag{35}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum\nolimits_{k=1}^{N} \left(\hat{x}_k - x_k\right)^2} \tag{36}$$

In Table 2, it can be learned in detail that the value of the forgetting factor has a great influence on the FFRLS algorithm, and the inappropriate forgetting factor directly affects the estimation accuracy of the FFRLS algorithm, which further illustrates the dependence of the FFRLS algorithm on the forgetting factor, whereas, in the AFFRLS algorithm, the AFFRLS algorithm can still be estimated accurately, because the forgetting factor can be adjusted adaptively and the inappropriate value of the forgetting factor can be corrected automatically in the iteration. Meanwhile, experiments on lithium-ion batteries at two temperatures, 25 °C and 35 °C, show that the temperature change has an effect on the equivalent resistance and equivalent capacitance of the equivalent circuit model of the battery, but the estimation accuracy of the AFFRLS algorithm for the equivalent circuit model of lithium-ion batteries still maintains a high level.

Table 2 Error analysis of parameter identification at $\lambda = 0.99$ and $\lambda = 0.98$

Temperature	Identification method	AFFRLS ($\lambda = 0.99$)	AFFRLS ($\lambda = 0.98$)	FFRLS (λ=0.99)	FFRLS ($\lambda = 0.98$)
25 °C	MAE	0.00984	0.00984	0.01238	0.02318
	RMSE	0.01211	0.01212	0.01448	0.02313
35 °C	MAE	0.01011	0.01012	0.01381	0.03156
	RMSE	0.01213	0.01211	0.01503	0.03881

Lithium-ion battery SOC joint algorithm estimation

In this paper, the AFFRLS-IPSO-SDUPF joint algorithm will be used to estimate the SOC of Li-ion batteries under BBDST and DST operating conditions, while the AFFRLS algorithm will be used as a parameter identification method and EKF, SDUKF, IPSO-SDUKF, and SDUPF joint algorithms, respectively, will be used for estimation of SOCs of Li-ion batteries. as a comparison. Here, in this paper, the initial process noise of all five algorithms is set to $Q = 10^{-5} \times E_{3\times3}$, while the initial sampling noise of all five algorithms is set to $R = 0.02$. In addition, the initial error covariance of all five algorithms is set to $P = 0.8 \times E_{3\times3}$.

The experiments in Figs. 15 and 16 were all performed at 25 °C. In Figs. 15 and 16, the estimation curves of the five algorithms for the actual SOC in the BBDST condition can be seen, Fig. 15 demonstrates the estimation curves of the five algorithms for the SOC under the BBDST condition, and Fig. 16 kind of demonstrates the estimation errors of the five algorithms under the BBDST condition. Among them, the error of the EKF algorithm fluctuates the most, and the estimation error of the SDUKF algorithm is improved compared with the EKF algorithm, it is worth mentioning that the estimation error of the IPSO-SDUKF algorithm has a large improvement in estimation accuracy after the adaption of noise by IPSO algorithm. Meanwhile, in the SDUPF algorithm, since the PF directly uses the particles generated by the SDUKF algorithm, it can be seen that compared to the SDUKF algorithm, not only is the error reduced, but the error fluctuation is also reduced at the same time. Finally, it can be seen that the IPSO-SDUPF algorithm has the smallest error fluctuation and the smallest estimation error among the five algorithms.

Fig. 15 SOC estimation under BBDST working conditions

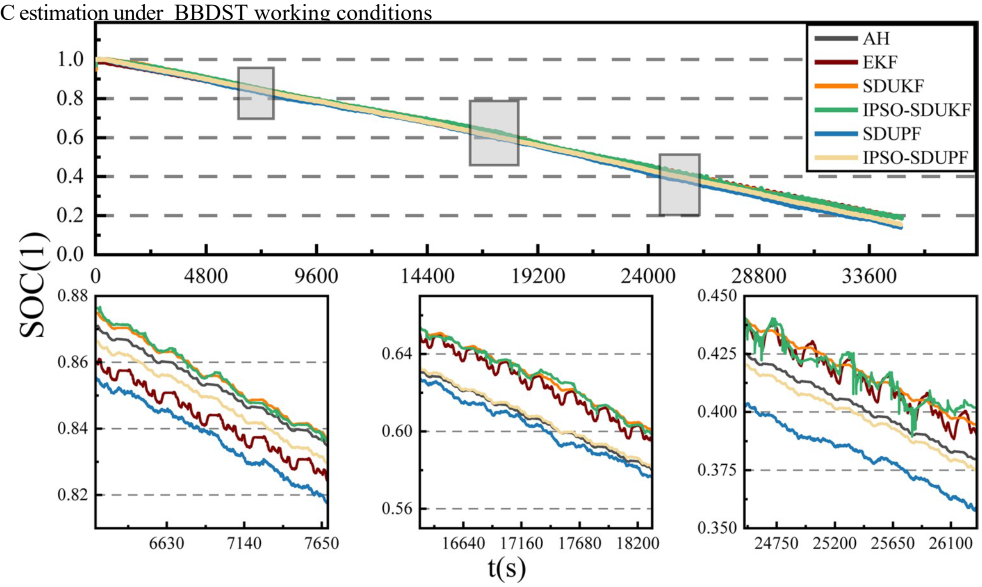
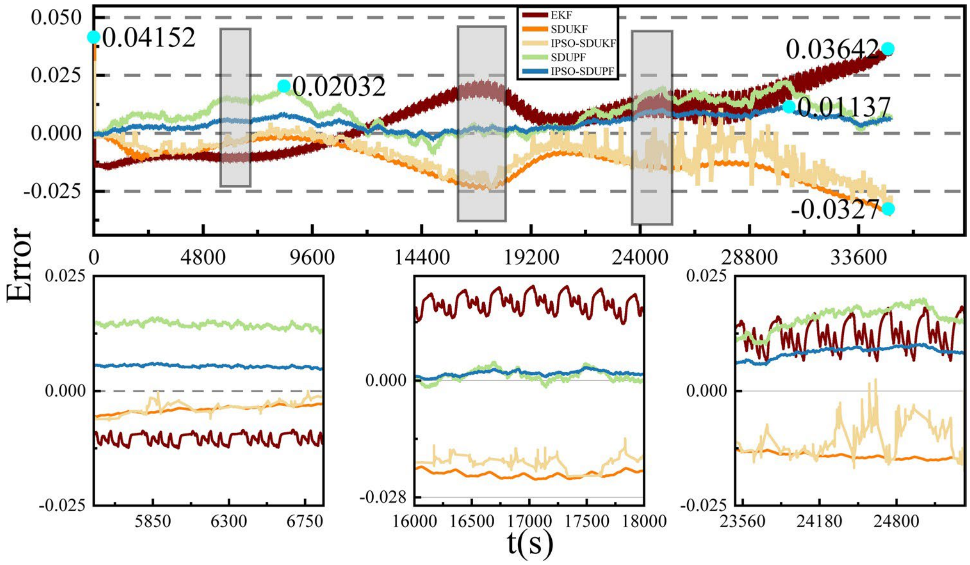


Fig. 16 SOC estimation error curve under BBDST working conditions



As shown in Fig. 17 and 18, it can be seen that the difference in estimation error of SOC by different algorithms is more obvious in the DST condition; also, EKF algorithm has the worst performance among all five algorithms, with large fluctuation of error, and it cannot estimate the SOC very well, similarly, SDUKF algorithm also has a large estimation error of SOC, both IPSO-SDUKF algorithm and SDUPF algorithm have a smaller estimation error, but the SDUPF algorithm has smoother fluctuations in estimation error. Finally, from the error curves and the SOC estimation curves, it can be seen that the IPSO-SDUPF algorithm also has the best estimation performance in the DST condition.

In Fig. 19, the ability of the five algorithms to estimate SOC will be measured using the mean absolute error (MAE), root mean square error (RMSE), and MAX error, respectively. It can be seen that part (a) in Fig. 19 represents the BBDST working condition. Among them, under the MAE metric, the EKF algorithm performs the worst and the IPSO-SDUPF algorithm performs the best. Under the RMSE metric, the EKF algorithm performs the worst, and it is worth noting that the performance of the IPSO-SDUKF and IPSO-SDUPF algorithms are close to each other, which shows that the IPSO optimization algorithm is effective in improving the accuracy of SOC estimation. However, under the MAX error metric, the IPSO-SDUPF algorithm has superior SOC estimation performance.

In part (b) of Fig. 19, the five algorithms are shown for the DST working condition with three evaluation criteria. Among them, unsurprisingly, it is still the EKF algorithm that has the worst estimation performance, where the IPSO-SDUPF algorithm is still improved compared to the IPSO-SDUKF algorithm under the two major evaluation criteria of MAE and RMSE, but the IPSO-SDUPF has the best performance under MAX error. The advantages of the IPSO-SDUPF algorithm are further illustrated in Table 3. It performs well under both MAE, RMSE, and MAX error evaluation criteria.

Above, the IPSO-SDUPF algorithm performs well in the testing of Li-ion batteries under BBDST condition and DST condition in a constant temperature environment at 25 °C, as well as in the on-line estimation of SOC of Li-ion batteries. Next, we will test the Li-ion batteries again under BBDST and DST conditions in a constant temperature environment at 35 °C to evaluate the performance of the IPSO-SDUPF algorithm in performing SOC on line estimation. Here, the noise parameters within each algorithm are unchanged compared to the experiments in the 25 °C constant temperature environment.

Fig. 17 SOC estimation under DST working conditions

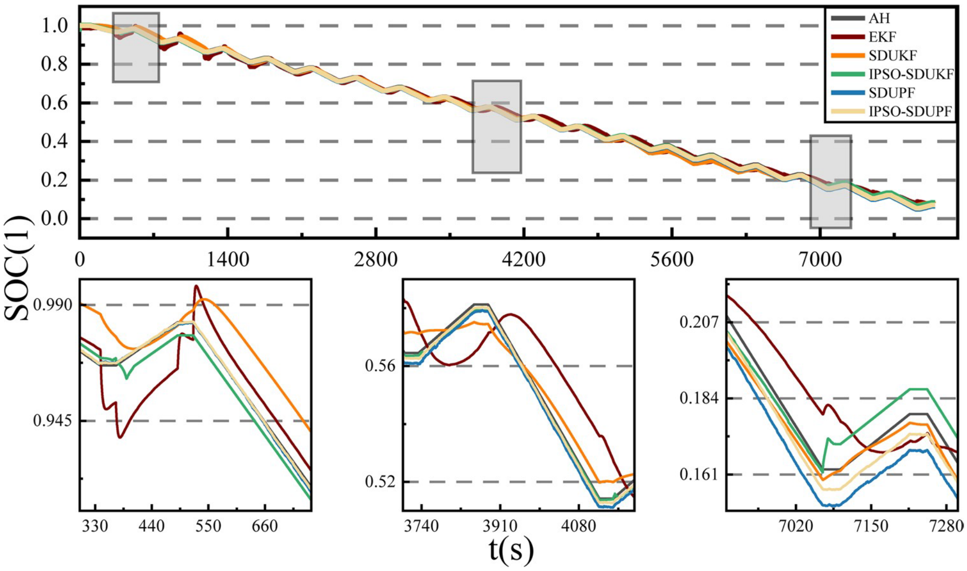
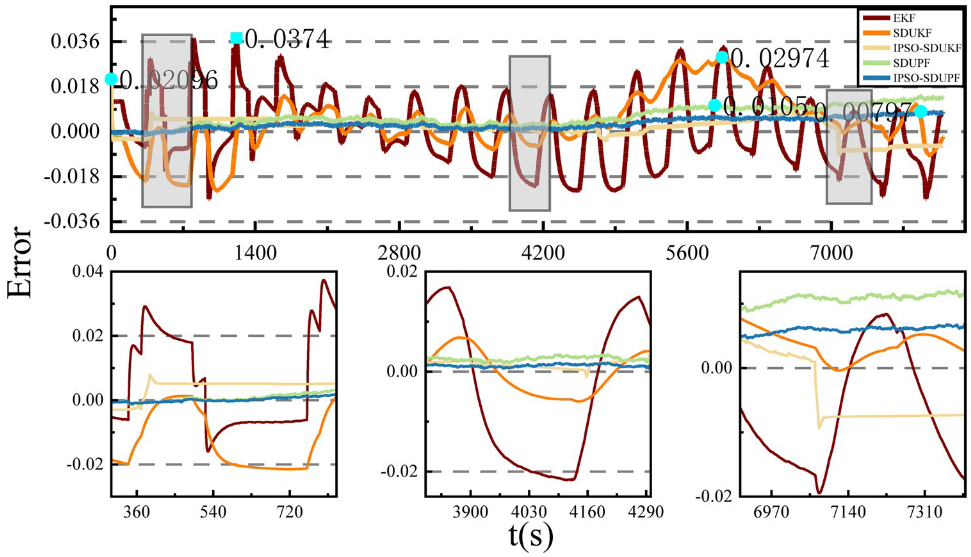
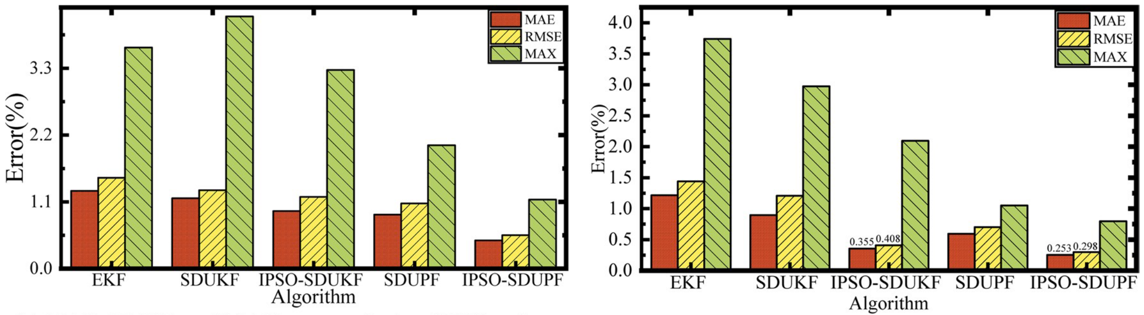


Fig. 18 SOC estimation error curve under DST working conditions





(a) MAE, RMSE, and MAX error analysis of SOC under BBDST condition

(b) MAE, RMSE, and MAX error analysis of SOC under DST condition

Fig. 19 MAE, RMSE, and MAX error analysis of SOC

Table 3 Comparison of MAE, RMSE, and MAX errors for BBDST and DST conditions at 25 °C

Working condition	Estimation method	EKF	SDUKF	IPSO-SDUKF	SDUPF	IPSO-SDUPF
BBDST	MAE	0.01281	0.01159	0.00949	0.00889	0.00465
	RMSE	0.01496	0.01293	0.01181	0.01074	0.00551
	MAX	0.03642	0.04152	0.0327	0.02032	0.01137
DST	MAE	0.01216	0.00896	0.00355	0.00594	0.00253
	RMSE	0.0144	0.01208	0.00408	0.00701	0.00298
	MAX	0.0374	0.02974	0.02096	0.0105	0.00797

In Figs. 20 and 21, comparing the experiments at 25 °C and 35 °C, the basic situation is the same in the validation of the five algorithms. Among them, EKF has the worst estimation performance, with large fluctuations in accuracy during estimation, and the SDUKF algorithm is improved compared to the EKF algorithm, but still has large error fluctuations. The SOC estimation of lithium-ion batteries by the IPSO-SDUKF algorithm has already had a significant improvement compared to the EKF and SDUKF algorithms, but the error fluctuations are still large. It can be seen that the estimation error curve of the SDUPF algorithm has smaller fluctuations compared to the IPSO-SDUKF algorithm. Finally, it can be seen that the estimation error curve of the IPSO-SDUPF algorithm has a maximum error of only 0.01015, and the overall estimation error is within a good range, which can be well realized for the online SOC estimation of lithium-ion batteries. Next, as shown in Figs. 22 and 23, in this paper, online SOC estimation is performed for Li-ion batteries under DST operating conditions in a constant temperature environment at 35 °C. In Figs. 22 and 23, similar to the test results of the five algorithms in the BBDST case, the IPSO-SDUPF algorithm undoubtedly achieved the best estimation results. Among them, the IPSO-SDUKF algorithm improves significantly compared to the EKF and SDUKF algorithms, and the IPSO-SDUPF algorithm also improves the estimation effect compared to the SDUPF algorithm. Meanwhile, the overall maximum error of the IPSO-SDUPF algorithm is only 0.01029.

Fig. 20 SOC estimation under BBDST working conditions

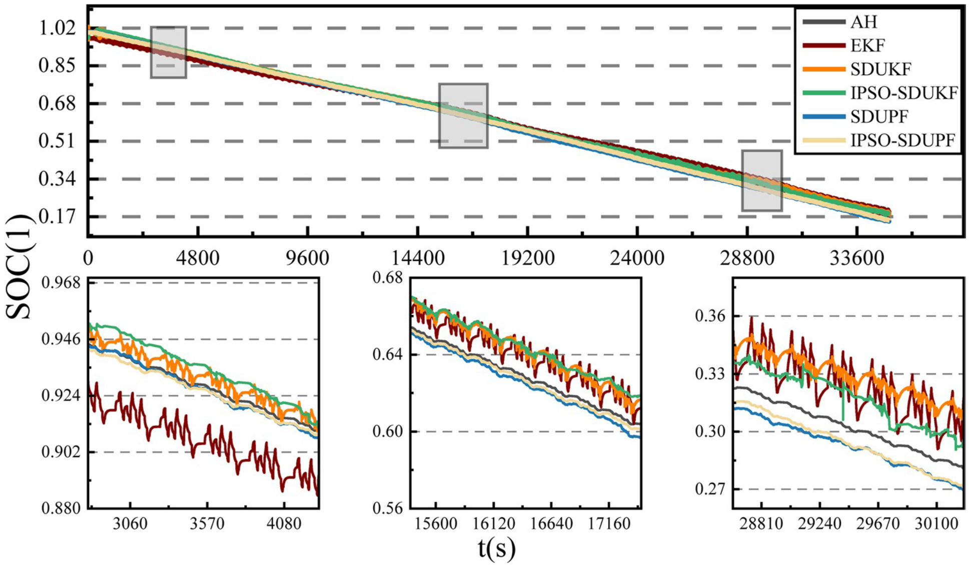


Fig. 21 SOC estimation error curve under BBDST working conditions

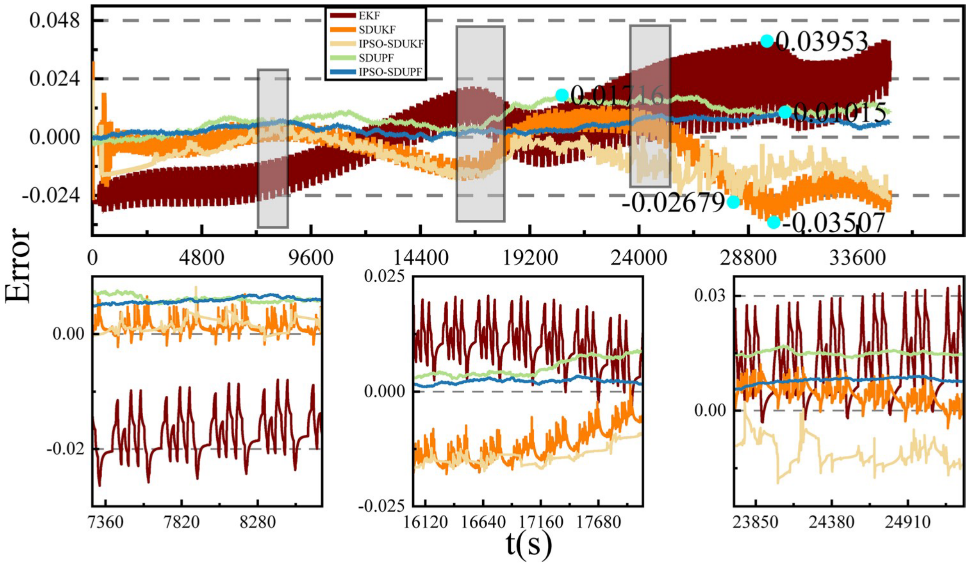


Fig. 22 SOC estimation under DST working conditions

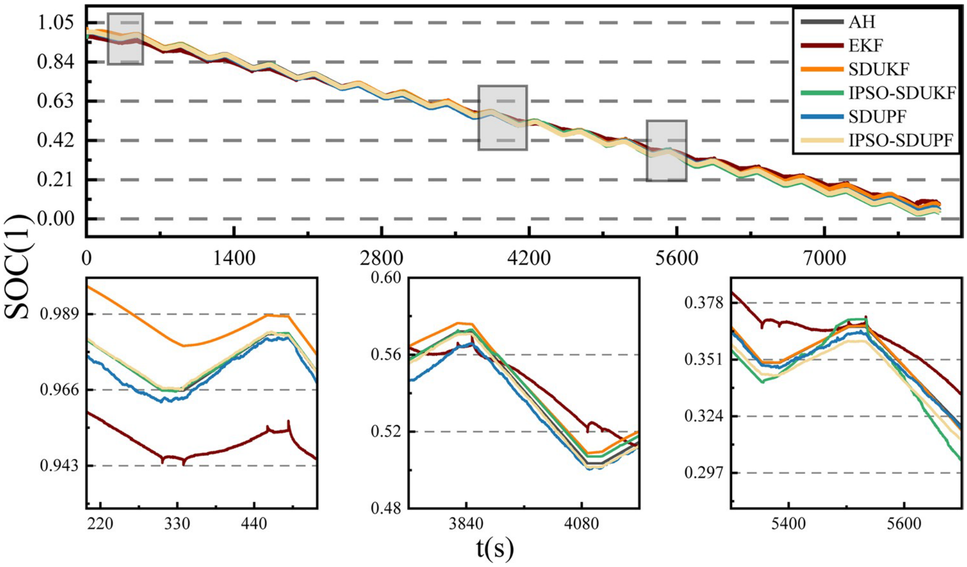
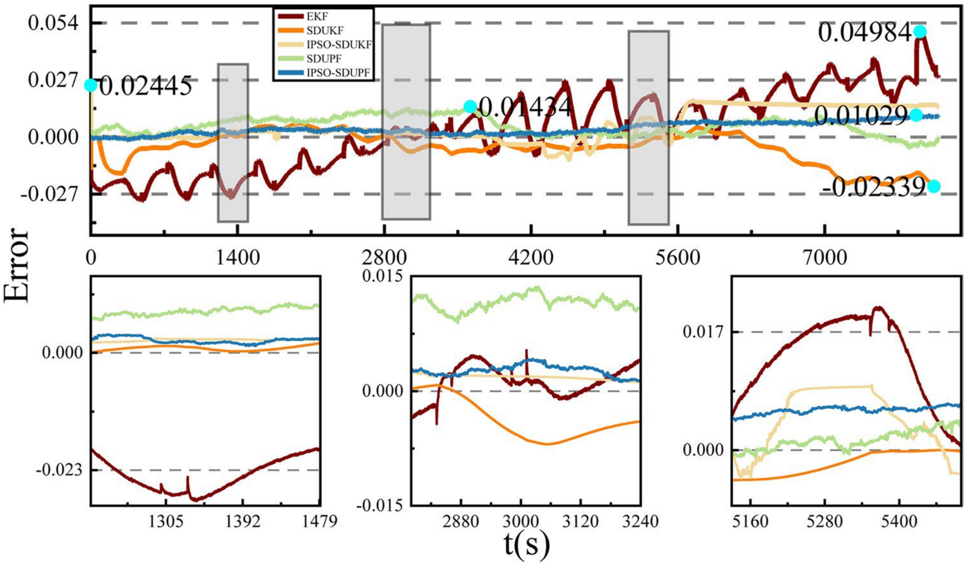


Fig. 23 SOC estimation error curve under DST working conditions

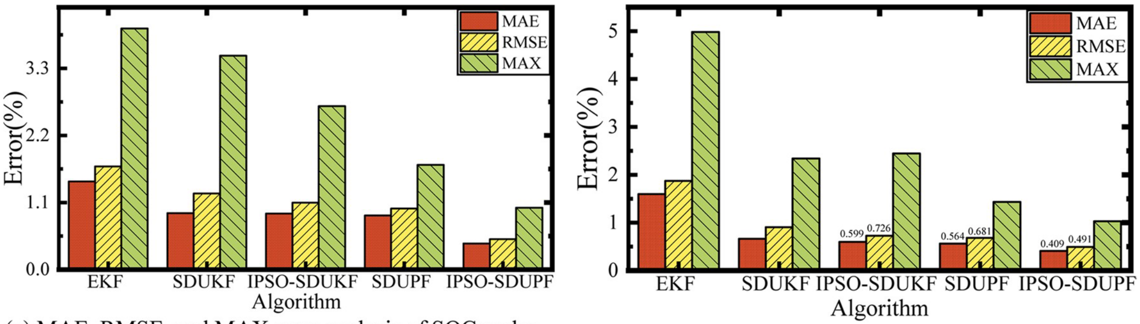


In Fig. 24, the five algorithms are represented in part (a) under the BBDST condition at 35 °C under the three evaluation metrics of MAE, RMSE, and MAX error, in which it can be seen that all the metrics of EKF are the worst, and the performance of the IPSO-SDUKF algorithm, SDUPF algorithm, and IPSO-SDUPF algorithm is progressively improved under these three evaluation metrics. In part (b), the performance of the five algorithms under the DST condition at 35 °C is represented under the three evaluation metrics of MAE, RMSE, and MAX Error. Similar to the BBDST condition, the IPSO- SDUPF algorithm has the best performance, where the MAE is 0.409% and the RMSE is 0.491%.

Here, the listed Table 4 further illustrates the advantages of the IPSO-SDUPF algorithm. Meanwhile, the IPSO- SDUPF algorithm performs best compared to the other four algorithms under the evaluation criteria of MAE, RMSE, and MAX errors at 35 °C.

In order to verify the stability of the IPSO-SDUPF algorithm for SOC estimation of lithium batteries, this paper will combine the UKF algorithm, SDUKF algorithm, IPSO-SDUKF algorithm, and IPSO-SDUPF algorithm based on the on line parameter identification of AFFRLS in the singularity of error covariance, i.e., the error covariance matrices $P_0 = -0.8 \times E_{3 \times 3}$. The rest of the parameters are kept constant to estimate the SOC of the Li-ion battery, where the SOC estimation curves and error curves at 25°C are shown in Figs. 25 and 26, and the SOC estimation curves and error curves at 35 °C are shown in Figs. 27 and 28.

As shown in Figs. 25, 26, 27, and 28, for the UKF algorithm, the SOC of the Li-ion battery cannot be correctly estimated because the initial error covariance is set to a singular matrix, but SDUKF, IPSO-SDUKF and IPSO-SDUPF all use singular value decomposition instead of the Cholesky decomposition, thus avoiding the adverse effect on the stability of the algorithms when the error covariance matrix is singular. stability of the algorithm when the error covariance matrix is singular. Meanwhile, to have a better test, we validate the algorithm at 25 °C and 35 °C. It can be seen that the use of SDUKF, IPSO-SDUKF, and IPSO-SDUPF algorithms is not affected when the error covariance matrix is singular. It can be seen that the use of singular value decomposition instead of Cholesky decomposition can be a good solution to the problem of instability caused by singular error covariance matrices in the iterative process.



(a) MAE, RMSE, and MAX error analysis of SOC under BBDST condition

(b) MAE, RMSE, and MAX error analysis of SOC under DST condition

Fig. 24 MAE, RMSE, and MAX error analysis of SOC

Table 4 Comparison of MAE, RMSE, and MAX errors for BBDST and DST conditions at 35 °C

Working condition	Estimation method	EKF	SDUKF	IPSO-SDUKF	SDUPF	IPSO-SDUPF
BBDST	MAE	0.01444	0.00926	0.00919	0.00886	0.00429
	RMSE	0.01691	0.01248	0.01098	0.01001	0.00501
	MAX	0.03953	0.03507	0.02679	0.01716	0.01015
DST	MAE	0.01596	0.00664	0.00599	0.00564	0.00409
	RMSE	0.01872	0.00904	0.00726	0.00681	0.00491
	MAX	0.04984	0.02339	0.02445	0.01434	0.01029

Fig. 25 Estimated battery SOC of singular matrix at 25 °C for BBDST condition

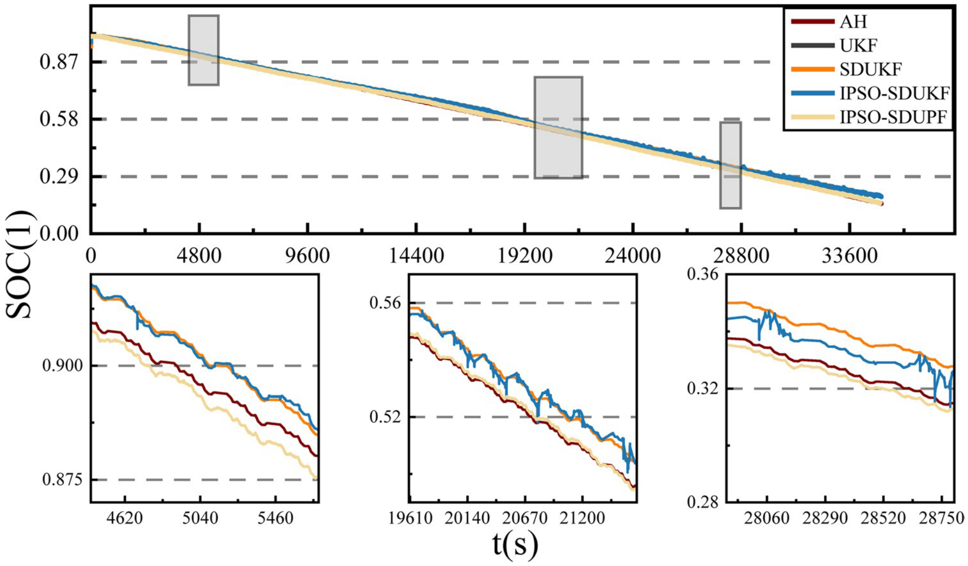


Fig. 26 Battery SOC estimation error of singular matrix at 25 °C for BBDST condition

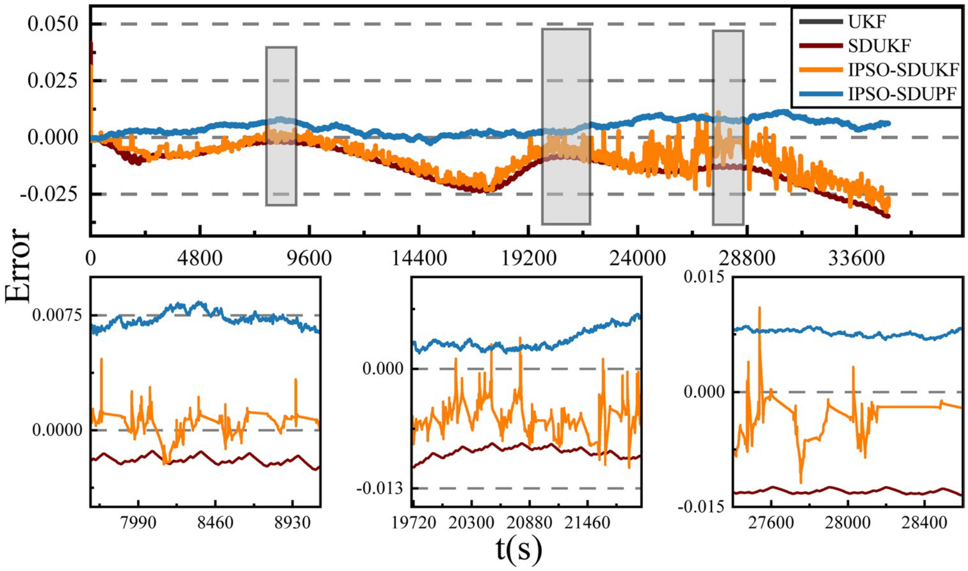


Fig. 27 Estimated battery SOC of singular matrix at 35 °C for BBDST condition

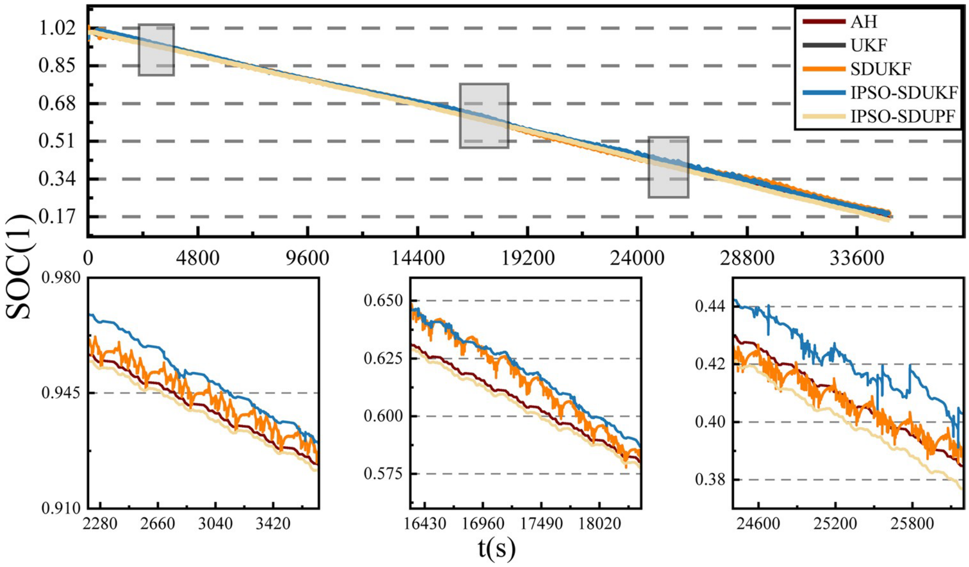
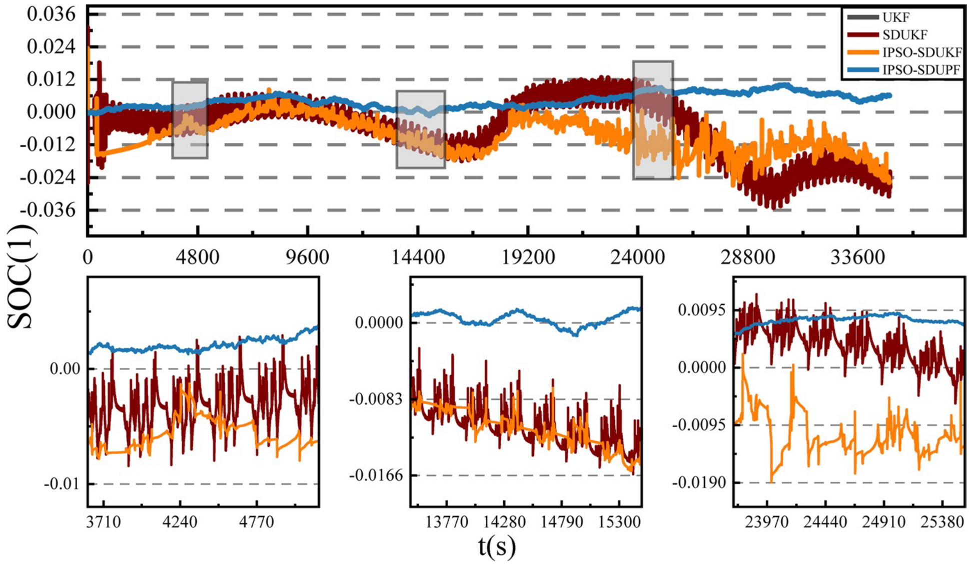


Fig. 28 Battery SOC estimation error of singular matrix at 35 °C for BBDST condition



Conclusion

In this paper, based on the second-order RC equivalent circuit model, the AFFRLS online parameter identification method is combined with the IPSO-SDUPF algorithm to characterize the state and output characteristics of lithium-ion batteries under different operating conditions at multiple temperatures. The basic contributions can be summarized in the following four points:

1. For the stable and accurate estimation of SOC, the core parameter of lithium-ion battery, an adaptive variable forgetting factor FFRLS model parameter identification method is proposed and combined with the IPSO-SDUPF algorithm to carry out a joint online estimation method of SOC for lithium-ion battery (AFFRLS-SDUPF). Good identification accuracy is guaranteed under two complex working conditions.

2. The AFFRLS online parameter identification method was combined with EKF, SDUKF, IPSO-SDUKF, SDUPF, and IPSO-SDUPF algorithms to estimate the SOC of lithium-ion batteries using five algorithms at the same time under the BBDST and DST conditions, and it was experimentally demonstrated that the IPSO-SDUPF algorithm had the highest estimation accuracy.
3. Parameter identification of Li-ion batteries was carried out at 25 °C and 35 °C and also tested under BBDST and DST conditions. The results show that temperature affects the internal resistance and capacitance of batteries, but the IPSO-SDUPF algorithm still performs well in the experiments in terms of SOC estimation accuracy of Li-ion batteries.
4. The singular value decomposition added to the UPF algorithm by the SDUPF algorithm is a good solution to the problem of system instability caused by the negative determination of the error covariance, and the addition of IPSO, which is self-adaptive to the noise, further improves the estimation accuracy of the system. The estimation of lithium-ion batteries by the IPSO-SDUPF algorithm combines both the stability and the accuracy and combines it with the AFFRLS algorithm to carry out the online estimation in real time. Combined with the AFFRLS algorithm for real-time online estimation of the SOC of lithium-ion batteries, the system can satisfy the various demands for SOC estimation.

Accurate estimation of SOC of Li-ion batteries is the focus and difficulty of condition monitoring of Li-ion batteries. The SOC data of the battery is not only the core parameter of the BMS, but also very important to ensure the accurate estimation of SOC to ensure the stability and safe use of the battery. In order to ensure that the algorithm can still work well under various complex environments, the AFFRLS-IPSO-SDUPF algorithm proposed in this paper is validated under BBDST and DST conditions for the above three aspects. At the same time, after the experimental demonstration at both temperatures, the IPSO-SDUPF algorithm outperforms the EKF, SDUKF, IPSO-SDUKF, and SDUPF algorithms in terms of system stability and estimation accuracy under the premise that the AFFRLS algorithm performs the online parameter estimation of the lithium-ion battery. Meanwhile, when the initial error covariance is set to a singular matrix, the IPSO-SDUPF algorithm performs well in terms of both system stability and estimation accuracy, which ensures the accuracy of SOC estimation.

Funding The work was supported by the National Natural Science Foundation of China (Nos. 62173281 and 61801407).

References

1. Wang C, Wang S, Zhou J et al (2023) A novel back propagation neural network-dual extended Kalman filter method for state-of- charge and state-of-health co-estimation of lithium-ion batteries based on limited memory least square algorithm. J Energy Storage 59:106563. https://doi.org/10.1016/j.est.2022.106563
2. Li Z, Xiong R, Mu H et al (2017) A novel parameter and state- of-charge determining method of lithium-ion battery for electric vehicles. Appl Energy 207:363–371. https://doi.org/10.1016/j.apenergy.2017.05.081
3. Wang C, Wang S, Zhou J, Qiao J (2022) A novel BCRLS-BP-EKF method for the state of charge estimation of lithium-ion batteries. Int J Electrochem Sci:220431. https://doi.org/10.20964/2022.04.53
4. Chaoui H, Mandalapu S (2017) Comparative study of online open circuit voltage estimation techniques for state of charge estimation of lithium-ion batteries. Batteries 3:12. https://doi.org/10.3390/batteries3020012
5. Yu Q, Nie Y, Peng S et al (2023) Evaluation of the safety stand- ards system of power batteries for electric vehicles in China. Appl Energy 349:121674. https://doi.org/10.1016/j.apenergy.2023.121674
6. Qiao J, Wang S, Yu C et al (2023) A chaotic firefly- particle filtering method of dynamic migration modeling for the state-of-charge and state-of-health co-estimation of a lithium-ion battery performance. Energy 263:126164. https://doi.org/10.1016/j.energy.2022.126164
7. Duan W, Song S, Xiao F et al (2023) Battery SOH estimation and RUL prediction framework based on variable forgetting factor online sequential extreme learning machine and particle filter. J Energy Storage 65:107322. https://doi.org/10.1016/j.est.2023.107322

8. Chen X-K, Sun D (2015) Modeling and state of charge estimation of lithium-ion battery. Adv Manuf 3:202–211. https://doi.org/10.1007/s40436-015-0116-3
9. Fan B, Pu J (2014) State of charge prediction for lithium-ion batteries. Electr Power Compon Syst 42:464–470. https://doi.org/10.1080/15325008.2013.857368
10. Wang L, Han J, Liu C, Li G (2022) State of charge estimation of lithium-ion based on VFFRLS-noise adaptive CKF algorithm. Ind Eng Chem Res 61:7489–7503. https://doi.org/10.1021/acs.iecr.1c03999
11. Du J, Liu Z, Wang Y (2014) State of charge estimation for Li-ion battery based on model from extreme learning machine. Control Eng Pract 26:11–19. https://doi.org/10.1016/j.conengprac.2013.12.014
12. Duan J, Wang P, Ma W et al (2020) State of charge estimation of lithium battery based on improved correntropy extended Kalman filter. Energies 13:4197. https://doi.org/10.3390/en13164197
13. Liu Z, Dang X (2018) A new method for state of charge and capacity estimation of lithium-ion battery based on dual strong tracking adaptive H infinity filter. Math Probl Eng 2018:1–18. https://doi.org/10.1155/2018/5218205
14. Lao Z, Xia B, Wang W et al (2018) A novel method for lithium-ion battery online parameter identification based on variable forgetting factor recursive least squares. Energies 11:1358. https://doi.org/10.3390/en11061358
15. Li J, Ye M, Jiao S et al (2020) A novel state estimation approach based on adaptive unscented Kalman filter for electric vehicles. IEEE Access 8:185629–185637. https://doi.org/10.1109/ ACCESS.2020.3030260
16. Liu S, Cui N, Zhang C (2017) An adaptive square root unscented Kalman filter approach for state of charge estimation of lithium-ion batteries. Energies 10:1345. https://doi.org/10.3390/en10091345
17. Ma D, Gao K, Mu Y et al (2022) An adaptive tracking-extended Kalman filter for SOC estimation of batteries with model uncertainty and sensor error. Energies 15:3499. https://doi.org/10.3390/ en15103499
18. Zhang S, Guo X, Zhang X (2020) An improved adaptive unscented kalman filtering for state of charge online estimation of lithium- ion battery. J Energy Storage 32:101980. https://doi.org/10.1016/j. est.2020.101980
19. He H, Qin H, Sun X, Shui Y (2013) Comparison study on the battery SoC estimation with EKF and UKF algorithms. Energies 6:5088–5100. https://doi.org/10.3390/en6105088
20. Wang J, Meng J, Peng Q et al (2023) Lithium-ion battery state-of-charge estimation using electrochemical model with sensitive parameters adjustment. Batteries 9:180. https://doi.org/10.3390/batteries9030180
21. Guo X, Xu X, Geng J et al (2019) SOC estimation with an adaptive unscented Kalman filter based on model parameter optimization. Appl Sci 9:4177. https://doi.org/10.3390/app9194177
22. Wang H, Zheng Y, Yu Y (2021) Joint estimation of SOC of lithium battery based on dual Kalman filter. Processes 9:1412. https://doi.org/10.3390/pr9081412
23. Lv J, Jiang B, Wang X et al (2020) Estimation of the state of charge of lithium batteries based on adaptive unscented Kalman filter algorithm. Electronics 9:1425. https://doi.org/10.3390/elect ronics9091425
24. Zhang C, Li K, Pei L, Zhu C (2015) An integrated approach for real-time model-based state-of-charge estimation of lithium-ion batteries. J Power Sources 283:24–36. https://doi.org/10.1016/j. jpowsour.2015.02.099
25. Li B, Bei S (2019) Estimation algorithm research for lithium battery SOC in electric vehicles based on adaptive unscented Kalman filter. Neural Comput & Applic 31:8171–8183. https://doi.org/10.1007/s00521-018-3901-7
26. Meng J, Luo G, Gao F (2016) Lithium polymer battery state-of- charge estimation based on adaptive unscented Kalman filter and support vector machine. IEEE Trans Power Electron 31:2226– 2238. https://doi.org/10.1109/TPEL.2015.2439578
27. Biazi V, Moreira AC, Pinto JL et al (2023) A particle filter-based virtual sensor for estimating the state of charge and internal temperature of lithium-ion batteries: implementation in a simulated study case. J Energy Storage 61:106814. https://doi.org/10.1016/j.est.2023.106814
28. Xia B, Sun Z, Zhang R, Lao Z (2017) A Cubature particle filter algorithm to estimate the state of the charge of lithium-ion batteries based on a second-order equivalent circuit model. Energies 10:457. https://doi.org/10.3390/en10040457
29. Zhou D, Zhang K, Ravey A et al (2016) Online estimation of lithium polymer batteries state-of-charge using particle filter- based data fusion with multimodels approach. IEEE Trans Ind Appl 52:2582–2595. https://doi.org/10.1109/TIA.2016.2524438
30. Peng S, Sun Y, Liu D et al (2023) State of health estimation of lithium-ion batteries based on multi-health features extraction and improved long short-term memory neural network. Energy 282:128956. https://doi.org/10.1016/j.energy.2023.128956
31. Peng S, Zhang A, Liu D et al (2023) State-of-charge estimation of lithium-ion batteries based on dual-coefficient tracking improved square-root unscented Kalman filter. Batteries 9:392. https://doi.org/10.3390/batteries9080392
32. Hao X, Wang S, Fan Y et al (2023) An improved forgetting factor recursive least square and unscented particle filtering algorithm for accurate lithium-ion battery state of charge estimation. J Energy Storage 59:106478. https://doi.org/10.1016/j.est.2022.106478
33. Hong S, Qin C, Lai X et al (2023) State-of-health estimation and remaining useful life prediction for lithium-ion batteries based on an improved particle filter algorithm. J Energy Storage 64:107179. https://doi.org/10.1016/j.est.2023.107179
34. Zhang K, Ma J, Zhao X et al (2019) State of charge estimation for lithium battery based on adaptively weighting cubature particle filter. IEEE Access 7:166657–166666. https://doi.org/10.1109/ACCESS.2019.2953478
35. Li Y, Chen J, Lan F (2020) 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 456:227984. https://doi.org/10.1016/j.jpowsour.2020.227984
36. Guo X, Kang L, Yao Y et al (2016) Joint estimation of the electric vehicle power battery state of charge based on the least squares method and the Kalman filter algorithm. Energies 9:100. https://doi.org/10.3390/en9020100
37. Wei X, Yimin M, Feng Z (2019) Lithium-ion battery modeling and state of charge estimation. Integr Ferroelectr 200:59–72. https://doi.org/10.1080/10584587.2019.1592620
38. He H, Xiong R, Zhang X et al (2011) State-of-charge estimation of the lithium-ion battery using an adaptive extended Kalman filter based on an improved Thevenin model. IEEE Trans Veh Technol 60:1461–1469. https://doi.org/10.1109/TVT.2011.2132812
39. Wang Q, Gao T, Li X (2022) SOC Estimation of lithium-ion batery based on equivalent circuit model with variable parameters. Energies 15:5829. https://doi.org/10.3390/en15165829
40. Sarrafan K, Muttaqi KM, Sutanto D (2020) Real-time estimation of model parameters and state-of-charge of Li-ion batteries in electric vehicles using a new mixed estimation model. IEEE Trans Ind Appl 56:5417–5428. https://doi.org/10.1109/TIA.2020.3002977
41. Liu X, Zheng C, Wu J et al (2020) An improved state of charge and state of power estimation method based on genetic particle filter for lithium-ion batteries. Energies 13:478. https://doi.org/10. 3390/en13020478

42. Tian Y, Lu C, Wang Z, Tao L (2014) Artificial fish swarm algorithm-based particle filter for Li-ion battery life prediction. Math Probl Eng 2014:1–10. https://doi.org/10.1155/2014/564894
43. Li M, Zhang Y, Hu Z et al (2021) A battery SOC estimation method based on AFFRLS-EKF. Sensors 21:5698. https://doi.org/10.3390/s21175698
44. Sun X, Ji J, Ren B et al (2019) Adaptive forgetting factor recursive least square algorithm for online identification of equivalent circuit model parameters of a lithium-ion battery. Energies 12:2242. https://doi.org/10.3390/en12122242
45. Lai X, Yuan M, Tang X et al (2022) Co-estimation of state-of- charge and state-of-health for lithium-ion batteries considering temperature and ageing. Energies 15:7416. https://doi.org/10. 3390/en15197416
46. Long H-Y, Zhu C-Y, Huang B-B et al (2019) Model parameters online identification and SOC joint estimation for lithium-ion battery based on a composite algorithm. J Electr Eng Technol 14:1485–1493. https://doi.org/10.1007/s42835-019-00179-w
47. Duan W, Song C, Chen Y et al (2020) Online parameter identification and state of charge estimation of battery based on multitime-scale adaptive double Kalman filter algorithm. Math Probl Eng 2020:1–20. https://doi.org/10.1155/2020/9502605
48. Ouyang T, Xu P, Chen J et al (2020) Improved parameters identification and state of charge estimation for lithium-ion battery with real-time optimal forgetting factor. Electrochim Acta 353:136576. https://doi.org/10.1016/j.electacta.2020.136576
49. Chen L, Wang S, Jiang H et al (2020) Decreasing weight particle swarm optimization combined with unscented particle filter for the non-linear model for lithium battery state of charge estimation. Int J Electrochem Sci 15:10104–10116. https://doi.org/10.20964/2020.10.41
50. He G, Wang Z, Ma H, Zhou X (2023) Optimal capacity configuration of wind–solar hydrogen storage microgrid based on IDW-PSO. Batteries 9:410. https://doi.org/10.3390/batteries9080410
51. Xia G, Wang G (2016) INS/GNSS tightly-coupled integration using quaternion-based AUPF for USV. Sensors 16:1215. https://doi.org/10.3390/s16081215