A hybrid algorithm based on beluga whale optimization-forgetting factor recursive least square and improved particle filter for the state of charge estimation of lithium-ion batteries.

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

2023

A hybrid algorithm based on beluga whale optimization-forgetting factor recursive least square and improved particle filter for the state of charge estimation of lithium-ion batteries

Xianfeng Shen[1], Shunli Wang[1,2*], Chunmei Yu[1], Chuangshi Qi[1], Zehao Li[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, Aberdeen, UK

[*] Shunli Wang wangshunli1985@qq.com

Abstract

Battery state of charge (*SOC*) is crucial in power battery management systems for improving the efficiency of battery use and its safety performance. In this paper, we propose a forgotten factor recursive least squares (FFRLS) method based on the beluga whale optimization (BWO) and an improved particle filtering (PF) algorithm for estimating the *SOC* of lithium batteries with ternary lithium batteries as the research object. Firstly, to address the accuracy deficiencies of the FFRLS method, the optimal parameter initial value and the forgetting factor value are optimized by using the BWO algorithm. Secondly, the adaptive simulated annealing algorithm (ASA) is introduced into the particle swarm optimization (PSO) to solve the sub-poor problem of traditional particle filtering. Experimental validation is performed by constructing complex working conditions, and the results show that the maximum error of parameter identification using the BWO-FFFRLS algorithm is stable within 2%. The *MAE* and *RMSE* are limited to within 2% when the ASAPSO-PF algorithm is applied to estimate the *SOC* estimation under Beijing Bus Dynamic Stress Test (BBDST), Dynamic Stress Test (DST), and Hybrid Pulse Power Characterization Test (HPPC) working conditions, indicating that the proposed algorithm has strong tracking capability and robustness for lithium battery *SOC*.

Keywords Lithium-ion battery ; Second-order RC-PNGV model ; BWO-FFRLS algorithm ; ASAPSO-PF algorithm ; State of charge

Introduction

The recent energy crisis and the effects of environmental pollution have sped up the development of electric vehicle research and application [1–3]. One of the crucial technologies for managing EV batteries and one of the essential factors of BMS is battery *SOC* estimation, which is crucial for thermal management, life prediction, and battery protection [4–6]. Since lithium-ion batteries have a high energy-to-weight ratio, a long life span, and a low white discharge rate, they are frequently utilized in electric cars [7–9]. The precise assessment of the *SOC* value is therefore crucial for both theoretical study and actual battery application [10–12].

The PF technique has received a lot of interest recently, both domestically and internationally [13, 14]. This method is frequently used in various disciplines, including visual tracking, robot localization, aircraft navigation, and defect detection, as it is simple to use, suitable for non-linear and non-Gaussian noise environments, and so on [15, 16]. PF has been used successfully by many academics to estimate the *SOC* of batteries [17, 18]. In non-linear systems, PF provides a good filtering effect; however, there is the issue of particle degradation [19, 20]. In this study, the sampling process is continuously improved using the PSO method by incorporating the most recent measurements into the sampling distribution and utilizing GPSO (Gaussian Particle Swarm Optimization), a technique for updating the particle's velocity based on a Gaussian distribution [21, 22]. This causes the sampling distribution to move toward the area with a higher posterior probability, thereby reducing the particle degradation phenomenon and enhancing accuracy [23, 24]. However, as more particles move toward that ideal particle, the velocity corresponding to each particle will become smaller and smaller [25–27]. Eventually, there is a greater likelihood that each particle is moving at the same velocity, which will inevitably bring about an approach to the local value. For this reason, we propose the algorithm of adaptive simulated annealing particle swarm optimization particle filtering [28, 29].

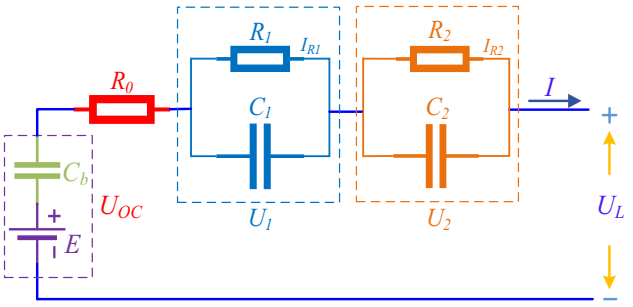
The ternary lithium battery is used as the research object in this paper, and the second-order RC-PNGV equivalent circuit model is chosen as the equivalent model of the lithium battery [30–32]. The RLS algorithm based on BWO and the forgetting factor is used as the model parameter identification algorithm. The problems of the FFRLS algorithm, multi-parameter identification of the battery model, slow convergence, and poor convergence accuracy of traditional identification algorithms were optimized. This algorithm has some benefits in terms of reliability and robustness, is straightforward to implement, and has higher accuracy. The ASAPSO-PF algorithm's guiding principles and stages for implementation are derived, and experiments are used to confirm the algorithm's accuracy. Under three complicated conditions: HPPC, BBDST, and DST, the PSO-PF, SAPSO-PF, and ASAPSO-PF algorithms show higher accuracy for *SOC* prediction compared to PF. To confirm the convergence, tracking effectiveness, and estimation accuracy of the algorithm under actual working conditions, *MAE* and *RMSE* evaluations are performed.

Mathematical analysis

Second-order RC-PNGV modeling

In this paper, the lithium battery is modeled with a modified PNGV model that substitutes a twin RC circuit for the original single RC circuit. In contrast to other models, which use a parallel circuit consisting of R1 and C1 with a smaller time constant to simulate the battery's rapid voltage change during a sudden change in current, and a parallel circuit consisting of R1 and C1 with a larger time constant to simulate the battery's rapid voltage change during the sudden change in current, these two RC links correspond to the two poles in the impedance spectrum of the lithium battery, which can more accurately describe the polarization [19, 33]. The enhanced PNGV circuit model can more accurately simulate the static circuit as well as characterize the battery's polarization characteristics. Given this, the parameter identification in this research uses the second-order RC-PNGV model, as illustrated in Fig. 1.

Fig. 1 Second-order RC-PNGV equivalent circuit model



In Fig. 1, E is an ideal voltage source, representing the open-circuit voltage of the power supply, which has a certain relationship with the *SOC*. The capacitance C_b describes the variation of the open-circuit voltage produced by the accumulation of current. E and C_b together represent the variation of the open-circuit voltage U_{oc}. R_0 is the ohmic internal resistance, and R_1 and R_2 represent the battery polarization resistance. C_1 and C_2 represent the battery polarization capacitance. I is the current flowing through the circuit, and U_L is the terminal voltage of the battery. Under the given constant temperature experiment, neglecting the effect of current, the constituent parameters of the model are functions of the battery *SOC*, and analyzing the constructed second-order RC- PNGV equivalent circuit model, the voltage and current expressions of the equivalent circuit can be obtained as shown in Eq. (1).

$$U_L = U_{OC} - R_O I - I_{R1}R_1 - I_{R2}R_2 \tag{1}$$

For the selected second-order equivalent model, $[SOC\ U_1\ U_2]^T$ is selected as the state variable, and the state space equation can be listed as shown in Eq. (2) by combining Eq. (1) and the definition of *SOC* after discretization.

$$\begin{cases} \begin{pmatrix} SOC_{k+1} \\ U_{1,k+1} \\ U_{2,k+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{\frac{-\Delta t}{\tau_1}} & 0 \\ 0 & 0 & e^{\frac{-\Delta t}{\tau_2}} \end{pmatrix} \begin{pmatrix} SOC_k \\ U_{1,k} \\ U_{2,k} \end{pmatrix} + \begin{pmatrix} -\frac{\eta \Delta t}{Q_N} \\ R_1\left(1 - e^{\frac{-\Delta t}{\tau_1}}\right) \\ R_2\left(1 - e^{\frac{-\Delta t}{\tau_2}}\right) \end{pmatrix} I_k \\ U_k = \begin{bmatrix} \frac{\partial U_{OC}}{\partial SOC} & -1 & -1 \end{bmatrix} \begin{bmatrix} SOC_k \\ U_{1,k} \\ U_{2,k} \end{bmatrix} - IR_0 \end{cases} \tag{2}$$

where Δt is the sampling interval, τ_1 and τ_2 are two parallel RC time constants, $\tau_1 = R_1C_{R1}$, $\tau_2 = R_2C_{R2}$, and Q_N is the battery capacity.

Online parameters identification based on improved FFRLS

FFRLS algorithm

RLS is characterized as one of the most commonly used methods in parameter identification, with good real-time performance, unique optimal solutions, and good analytical performance. However, the RLS method still has its drawbacks, such as being greatly perturbed by outliers and having an insufficient ability to filter Gaussian noise [34, 35]. In this regard, the RLS method with the forgetting factor introduces the forgetting factor λ based o n RLS, which can solve the problem of data saturation in the RLS algorithm. The discrete equation of the model to be identified and the corresponding difference equation are obtained by the Laplace transform, as shown in Eq. (3):

$$\begin{cases} G(z) = \dfrac{y(z)}{u(z)} = \dfrac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + \cdots + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \cdots + a_n z^{-n}} \\ y_k = -\sum_{i=1}^{n} a_i y_{k-1} + \sum_{i=1}^{n} b_i u_{k-1} + v_k \end{cases} \tag{3}$$

Therefore, it is widely used for battery model parameter identification. The improved algorithm formulation of FFRLS is shown in Eq. (4):

$$\begin{cases} \hat{\theta}_k = \hat{\theta}_{k-1} + K_k \left[y_k - h_k^T \hat{\theta}_{k-1} \right] \\ K_k = P_{k-1} h_k \left[\lambda + h_k^T P_{k-1} h_k \right]^{-1} \\ P_k = \frac{1}{\lambda} \left[1 - K_k h_k^T \right] P_{k-1} \end{cases} \tag{4}$$

Among them, $h_k = [y_{k-1}, y_{k-2}, -u_k, -u_{k-1}, -u_{k-2}]^T$.

However, for the FFRLS algorithm, the value of the forgetting factor λ has a great influence on the algorithm. If the value of λ is close to 1, the algorithm will not change with the change of model parameters; if the value of λ is close to 0, the algorithm will rely too much on the current data for recognition. Therefore, this paper uses an intelligent algorithm, the beluga whale optimization algorithm, to select the optimal values of the initial values of the optimal parameters and the forgetting factor values in the FFRLS method to improve its accuracy. The flow chart of the FFRLS algorithm is shown in Fig. 2.

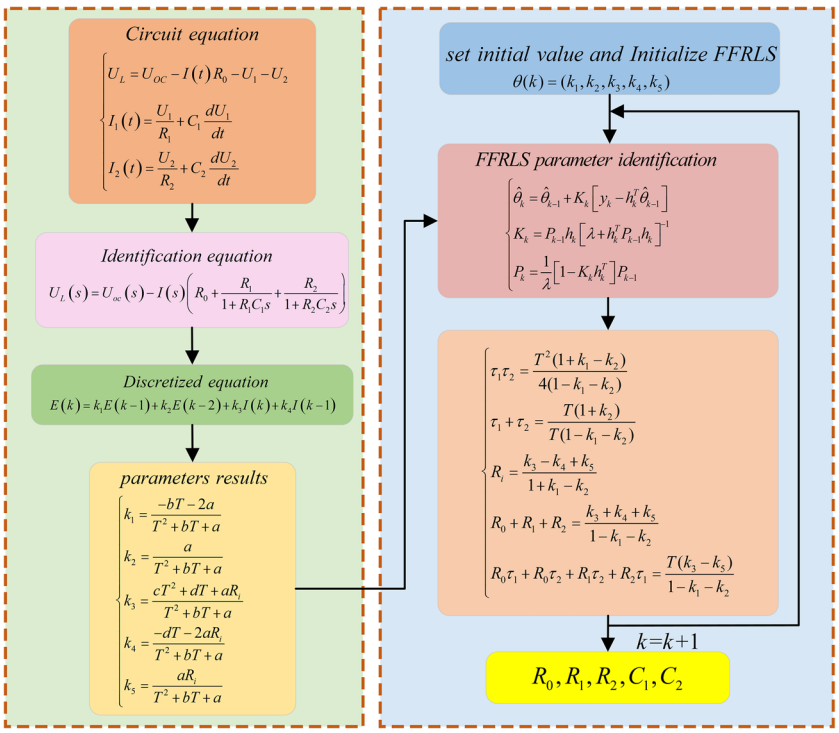


Fig. 2 Flow chart of FFRLS steps

BWO algorithm

BWO algorithm is a swarm intelligence algorithm compared with genetic algorithm (GA), simpler, less calculation formula, and very easy to implement; according to the equilibrium factor B_f to achieve the transition of the population from exploration to exploitation, the equilibrium factor is calculated as shown in Eq. (5):

$$B_f = B_0 \bullet \left(1 - \frac{t}{T}\right) \tag{5}$$

where t is the number of current iterations and T is the total number of iterations. The parameter B_0 is a random value in the range of (0, 1) that changes with each iteration. When the balance factor $B_f > 0.5$, it is the exploration phase; when the balance factor $B_f < 0.5$, it is the development phase. As the number of iteration t increases, the fluctuation range of B_f decreases from (0, 1) to (0, 0.5), indicating that the probabilities of the exploitation stage and exploration stage change significantly, and the probability of the exploitation stage is gradually larger with the increase in the number of iterations. Belugas can perform social behaviors in different poses, such as two belugas swimming closely together in a synchronized or mirror-image manner. Therefore, the position update of belugas is shown in Eq. (6):

$$\begin{cases} x_{i,j}^{t+1} = x_{i,p}^{t} + \left(x_{r,p}^{t} - x_{i,p}^{t}\right)(1 + r_1) \bullet \sin\left(2\pi r_2\right), j \text{ is an even number} \\ x_{i,j}^{t+1} = x_{i,p}^{t} + \left(x_{r,p}^{t} - x_{i,p}^{t}\right)(1 + r_1) \bullet \cos\left(2\pi r_2\right), j \text{ is an odd number} \end{cases} \tag{6}$$

where $x_{i,j}^{t+1}$ denotes the position of the ith individual on the jth dimension at the next iteration. P is a random integer, assuming that the problem dimension is D and P is a random integer in the range [1, D]. Thus, $x_{i,p}^{t}$ denotes the position of the ith individual on the random dimension P under the current iteration. r also denotes a random integer, and assuming that the population size is N, r is a random integer in the range [1, N]. Thus, $x_{r,p}^{t}$ denotes the position of a random individual r on the random dimension P under the current iteration which are random numbers between r_1, $r_2 \in (0, 1)$.

The development phase of the BWO algorithm was inspired by the feeding behavior of beluga whales. Belugas can move and forage based on the location of nearby belugas. Belugas hunt by sharing each other's position information, so the influence of the best individual and other individuals on position updates is considered. Levy flight strategy is introduced in the development phase of the BWO algorithm to enhance convergence. It is assumed that belugas can capture prey using the Levy flight strategy, whose mathematical model is expressed as Eq. (7):

$$x_i^{t+1} = r_3 \bullet x_{\text{best}}^{t} - r_4 \bullet x_i^{t} + C_1 \bullet L_F \bullet \left\{x_r^{t} - x_i^{t}\right\} \tag{7}$$

where $C_1 = 2 \bullet r_4 \bullet \left(1 - \frac{t}{T}\right)$ represents the random jump degree, which is used to measure the intensity of Levy flight, and r_3 and r_4 are random numbers between (0,1). L_F is a random number consistent with the Levy distribution, calculated by Eqs. (8), (9), and (10).

$$L_F = 0.05 \times \left(\mu \bullet \sigma / |v|^{\frac{1}{\beta}}\right) \tag{8}$$

$$L_F = 0.05 \times \left(\mu \bullet \sigma / |v|^{\frac{1}{\beta}}\right) \tag{9}$$

$$\Gamma(x) = (x - 1)! \tag{10}$$

where μ and v are random numbers subject to a normal distribution, $(\mu, v) \sim N(0, 1)$, and β is a constant, set to 1.5.

To keep the population size constant, the current position of the beluga whale and its falling step length are used to establish the position updating Eqs. (11) and (12).

$$x_i^{t+1} = r_5 \bullet x_i^t - r_6 \bullet x_r^t + r_7 \bullet x_{\text{step}}^t \tag{11}$$

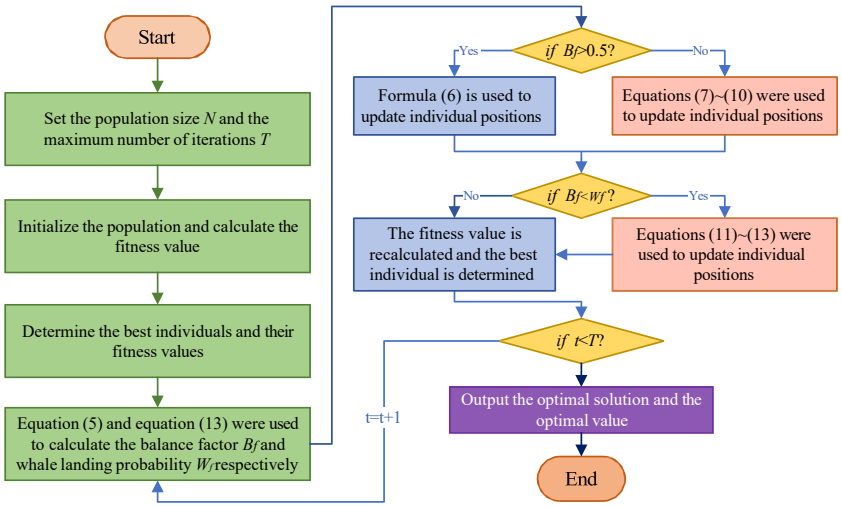
$$x_{\text{step}}^t = e^{\frac{-C_2 \bullet t}{T}} \bullet (Ub - Lb) \tag{12}$$

where r_5 , r_6 , and r_7 are random numbers of (0,1) and Ub and Lb are the lower and upper bounds of the optimization problem, respectively. $x_r{}^t$ is the position of individual r randomly selected from the current population and $x_{\text{step}}$is the falling step length of the whale. C_2 is a step factor, which is related to the probability of whale landing and population size: $C_2 = W_f \times N$. In this model, the probability W_f of a whale falling is designed as a linear function:

$$W_f = 0.1 - 0.05 \bullet \frac{t}{T} \tag{13}$$

The chance of whale fall dropped from 0.1 in the first iteration to 0.05 in the final iteration, demonstrating that the closer the beluga whale is to the food source, the lower the risk of the beluga whale in the optimization process. Figure 3 depicts the flow chart of the BWO method.

Fig. 3 Step flow chart of BWO algorithm



BWO-FFRLS algorithm

The issue of "data saturation" in a time-varying system of an equivalent model can be significantly improved by FFRLS. To acquire correct results, however, the fundamental question of the algorithm is how to select the best initial parameter value and forgetting factor. This is a challenge that we are currently facing. In this study, the BWO technique is used to discuss this issue. In contrast to other swarm intelligence algorithms like the GA method, which requires a lot of processing and data, the BWO approach is straightforward to use. It features high-efficiency traits and a good optimization effect at the same time. Overall, the BWO algorithm is capable of producing the desired outcomes. As a result, we decided to employ the BWO method in this study as an optimization approach to improve the FFRLS algorithm. The terminal voltage error probability was specified, the BWO algorithm was used, and the best starting parameter values and forgetting factor values were chosen in real time.

Figure 4 depicts the process flow chart for applying the BWO algorithm to enhance the FFRLS algorithm. Although the BWO-FFRLS algorithm is sophisticated, it still offers certain advantages over other algorithms in terms of the accuracy and stability of evaluation findings.

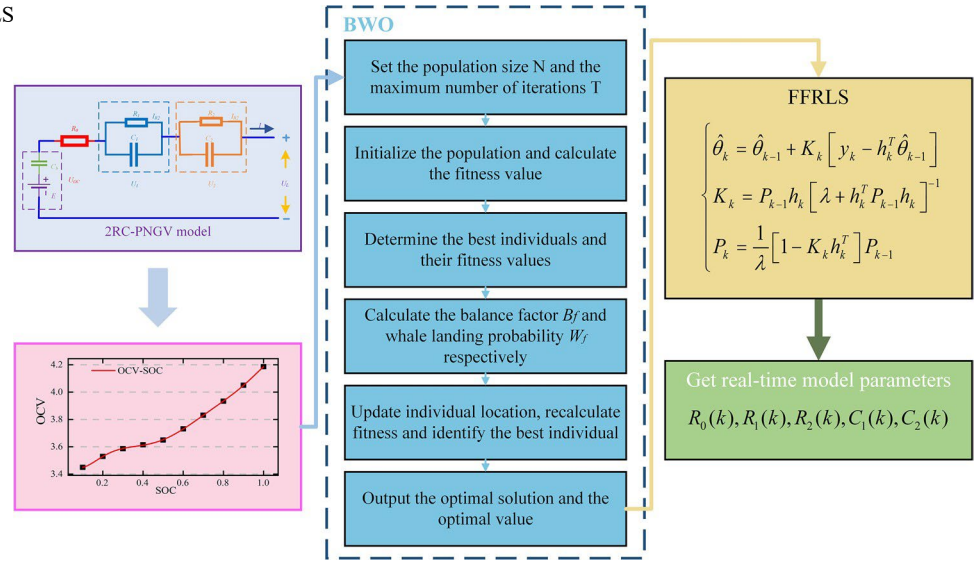


Fig. 4 Procedure flow of BWO- FFRLS algorithm

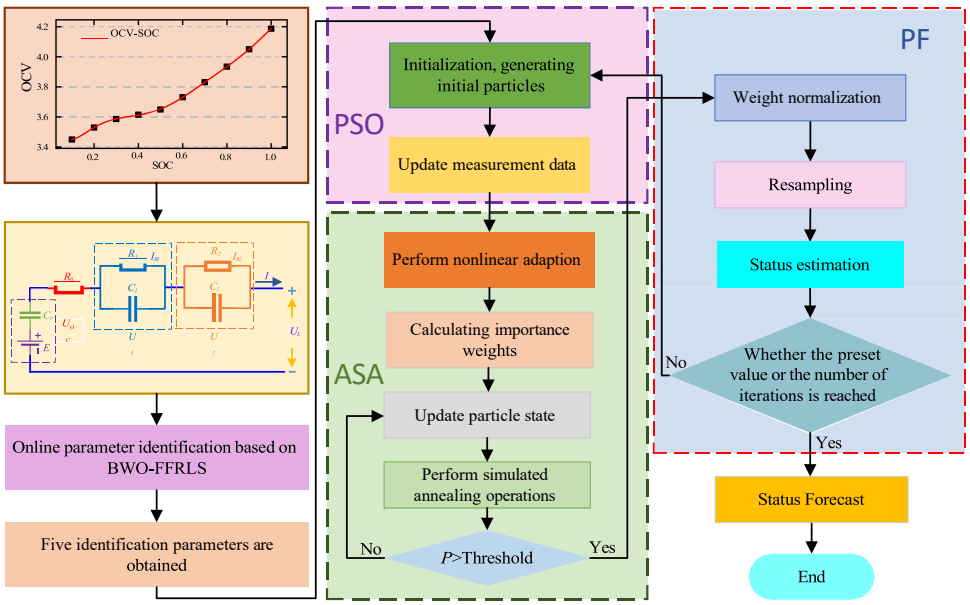


Fig. 5 The overall process framework of this paper

Improved particle filter algorithm

Figure 5 depicts this article's overall workflow. The weight of the majority of particles is very modest or even inconsequential during the process of particle weight correction, as a small number of particles gradually occupy the great majority of the particle weight. The accuracy of the PF algorithm is limited because a lot of computation work is applied to particles that are essentially unrelated to the predicted outcomes, which is the degradation problem of the particle filtering algorithm. To solve this issue, several academics suggested using the particle swarm optimization particle filtering technique (PSO-PF). The PSO algorithm increases the usage efficiency of particles by directing all particles to travel toward the area with a high probable probability to prevent particle degradation. This causes the particle swarm to move toward the ideal particle. By accepting a value that is worse than the current value with a given probability, we can update the particle based on the Gaussian distribution and broaden the scope of the global search. PSO and PF algorithms share a lot of commonalities. Although merging the two techniques results in a PF method that is more successful, the disadvantage of PSO being a local optimization problem has not yet been solved. To enhance the PSO-PF technique, the ASA algorithm is presented. The simulated

annealing operation is used to prevent the PSO algorithm from slipping into the local optimal. The population is steered to accept the different solutions with a specific probability following the Metropolis criterion and temperature, which are determined following the population's initial condition. The linear change strategy is used to control the social learning factors and the self-learning factors, and the hyperbolic tangent function is used to control the inertia weight coefficient to carry out non-linear adaptive changes, to achieve the goal of changing the optimization focus at various stages.

This paper's main premise is that when the PF method is optimized using PSO, the particle is chosen as the global ideal value when a new particle appears with a higher adaption value. The probability of simulated annealing is determined when the new particle fitness value does not satisfy this requirement. It is not acknowledged unless it exceeds the threshold, in which case it is the random number in the range (0, 1) and the particle is received.

Flow of the ASAPSO-PF algorithm

The ASAPSO-PF algorithm updates the particle position through the state model and then updates the particle weight. When the particles are not distributed at the true value, the weight of each particle is too low. At this time, the particle position is updated again by seeking the individual optimal value and the global optimal value, so that the particle keeps getting closer to the real value. The specific algorithm flow is as follows:

- **Step 1** Initialize parameters

Set the *SOC* initial value, $k = 0$, weight w_m initialization, variance R and Q, and then randomly generate N initial random particles with a Gaussian distribution, expressed as x_0^i, $i = 1, 2, 3 \ldots N$, the total number of particle swarm M, evolutionary algebra, and set the maximum and minimum of self-learning factor C_1 and social learning factor C_2.

- **Step 2** Status update

$$\begin{cases} \hat{x}_k = f\left(\hat{x}_{k-1}, I_{L,k-1}, w_k\right) \\ \hat{z}_k = g\left(\hat{z}_{k-1}, I_{L,k-1}, v_k\right) \end{cases} \tag{14}$$

The N particle states optimized at the last moment are substituted into the state equation and measurement equation of the lithium battery to estimate the N particle states of the battery at the next moment and the voltage at both ends of the battery. In this step, each predicted value is assigned to the particle swarm as the basic particle of the particle swarm.

- **Step 3** Update measurement data

Calculation error:

$$Error_k^i = Y_k - \hat{Y_k} \tag{15}$$

Fitness function:

$$Fitness = \exp\left[-\frac{1}{2R_k}\left(Y_k - \hat{Y}_k\right)^2\right] \tag{16}$$

Type: $\hat{Y_k}$ for the k moment, the ith particle calculated by the state equation and measurement equation, the estimated value; Y_k for the moment, the measured voltage value, the PSO algorithm, by calculating the fitness value of all the particles, moves to the optimal particle.

- **Step 4** Carry out non-linear adaptive changes

The negative hyperbolic tangent curve between [−4, 4] is selected to control the change in the inertia weight coefficient. The hyperbolic tangent curve is a non-linear control strategy, and its decline rate is slow at the

beginning of the search, which gives sufficient time for the particles to conduct a large range of global searches and reduces the situation of falling into local optimal. In the medium term, approximate linear decline gradually strengthens the ability of local search. In the later period, the rate of change decreases again, and a detailed local search is emphasized to accurately determine the initial state value of the global optimal solution.

Inertia weight function:

$$w = (w_{max} + w_{min})/2 + \tanh(-4 + 8 \times (k_{max} - k)/k_{max})(w_{max} - w_{min})/2 \tag{17}$$

Self-learning factor:

$$C_1 = C_{1\max} - k(C_{1\max} - C_{1\min})/k_{max} \tag{18}$$

Social learning factor:

$$C_2 = C_{2\min} - k(C_{2\min} - C_{2\max})/k_{max} \tag{19}$$

Take $C_{1\max} = 2.50$, $C_{2\max} = 1.25$, $C_{1\min} = 1.25$, and $C_{2\min} = 2.5$. With the increase in the number of iterations k, C_1 decreases linearly from 2.50 to 1.25, while C_2 increases linearly from 1.25 to 2.50. In this way, the ergodic requirement of particles in space is satisfied, and the global search ability is enhanced. When the number of iterations is more than half, $C_1 < C_2$, and the gap is larger and larger, the local search ability is also enhanced.

- **Step 5** Update particle status

According to the individual optimal value and global optimal value, Eq. (20) is used to update the state of each particle, including velocity and position, so that the particle approaches the real value. For each particle, its current adaptation value is compared with the best adaptation value experienced. If the current adaptive value is better than the best adaptive value, the current adaptive value is taken as the individual optimal value of the particle, and its position is the individual optimal position. The individual's optimal value is compared with the best adaptive value experienced within the group. If the individual optimal value is better that the best adaptive value within the group, it is regarded as the global optimal value.

$$\begin{cases} v_{k-1}^i = wv_{k-1}^i + C_1\text{rand}(P_{best} - x_{k-1}^i) + C_2\text{rand}(P_{gbest} - x_{k-1}^i) \\ x_k^i = x_{k-1}^i + v_{k-1}^i \end{cases} \tag{20}$$

Type, |rand n| and |Rand n| is a Gaussian random number, can use abs[$N(0, 1)$]; P_{best} is the individual optimal value; and $P_{g\text{best}}$ is the global optimal value.

- **Step 6** Simulated annealing operation is carried out on the optimal position

The Metropolis criterion in the simulated annealing algorithm is introduced into each iteration. The initial temperature is set according to the initial particle optimal value and attenuates with a certain cooling coefficient μ after each iteration. Specific operations are shown in Eq. (21) below.

$$T(k) = \begin{cases} \frac{E(g_{best})}{\log(0.2)} & k = 1 \\ T(k-1)\mu & k > 1 \end{cases} \tag{21}$$

where T is the initial temperature. After each iteration, the difference between the internal energy (fitness) of the updated updated position and the optimal internal energy of the population is calculated. The probability calculated according to Eq. (22) is compared with rand() to determine whether to accept the poor solution. Take the cooling coefficient $\mu = 0.95$. The proposed ASAPSO algorithm adaptively changes three important parameters of the PSO algorithm and adds a simulated annealing operation, which increases the precision and speed of optimization.

$$\begin{cases} 1 & E_i(k) \geq E_g \\ \exp\left(-\frac{E_i(k)-E_g}{T_i}\right) & E_i(k) \leq E_g \end{cases} \tag{22}$$

- **Step 7** Calculate the simulated annealing probability P

The equilibrium state distribution of finite conversions at a temperature T_i is as follows: the simulated annealing probability p is calculated. When P is greater than threshold r , the particle is received and the probability is recalculated by going to Eq. (23); otherwise, the optimal position is selected for the simulated annealing operation by going back to Eq. (20).

$$P_i(k) = \exp\left(\frac{-E_i(k-1)}{T}\right) / \sum_{j \in S} \exp\left(\frac{-E_i(k)}{T_i}\right) \tag{23}$$

$$P_i = \begin{cases} \frac{1}{|S_{min}|} & x_i \in |S_{min}| \quad and \quad \sum_{x_i \in S_{min}} P_i = 1 \\ 0, & other \end{cases} \tag{24}$$

$S_{\min}$ is a search space of optimal values. When the temperature drops to 0, the distribution of P_i is shown in Eq. (24). When the temperature drops, accompanied by a large number of state transitions, the probability of finding the global optimal is 1 when the state of thermal equilibrium is reached.

- **Step 8** Calculate particle importance weight:

The PSO is re-assigned to the observed value, and the weight of each particle is calculated

$$w_k^i = \frac{1}{\sqrt{2\pi R}} \exp\left[-\frac{1}{2R}\left(error_k^i\right)^2\right] \tag{25}$$

- **Step 9** Normalize the weight

$$w_k^i = w_k^i / \sum_{i=1}^{N} w_k^i \tag{26}$$

- **Step 10** Resampling

In the resampling process, calculate the effective number of particles $N_{eff} = 1/\sum_{i=1}^{N}(W_k^i)^2$; when $N_{eff}<N_{th}$ is defined as the effective number of particle threshold, resampling will be a weighted sample $\{x_{0:k}^i, w_k^1\}^{N}{}_{i=1}$ mapping for samples such as the right $\{x_{0:k}^i, \frac{1}{N}\}^{N}{}_{i=1}$.

- **Step 11** State estimation

$$\hat{x} = \sum_{i=1}^{N} w_k^i x_k^i \tag{27}$$

Variance estimation

$$P_k = \sum_{i=1}^{N} w_k^i \left(x_k^i - \hat{x}_k\right)\left(x_k^i - \hat{x}_k\right)^T \tag{28}$$

- **Step 12** State prediction

Predict the unknown state parameter x_{k+1}^{i} from Eq. (29) of the state

$$x_{k+1}^{i} = f\left(x_k^i, v_k\right) \qquad i = 1, 2, \ldots N \tag{29}$$

Analysis of experimental results

Experimental platform construction

The experimental platform architecture is shown in Fig. 6. In this paper, the rated capacity of lithium ions produced by Delipu Battery Technology Co., Ltd. is selected, and the actual capacity is 39.9667 Ah of ternary lithium batteries. The battery test system for Shenzhen Yakoyuan Technology Co., Ltd., provides BTS200-100-104 as lithium-ion battery charging and discharging test equipment. An incubator (TT-5166-7) at room temperature (25,852 °C) was used to provide room temperature for the experimental battery. In this paper, the experimental data of HPPC, BBDST, and DST under three conditions are obtained by using the lithium-ion battery test platform.

Fig. 6 Lithium battery experimental platform architecture

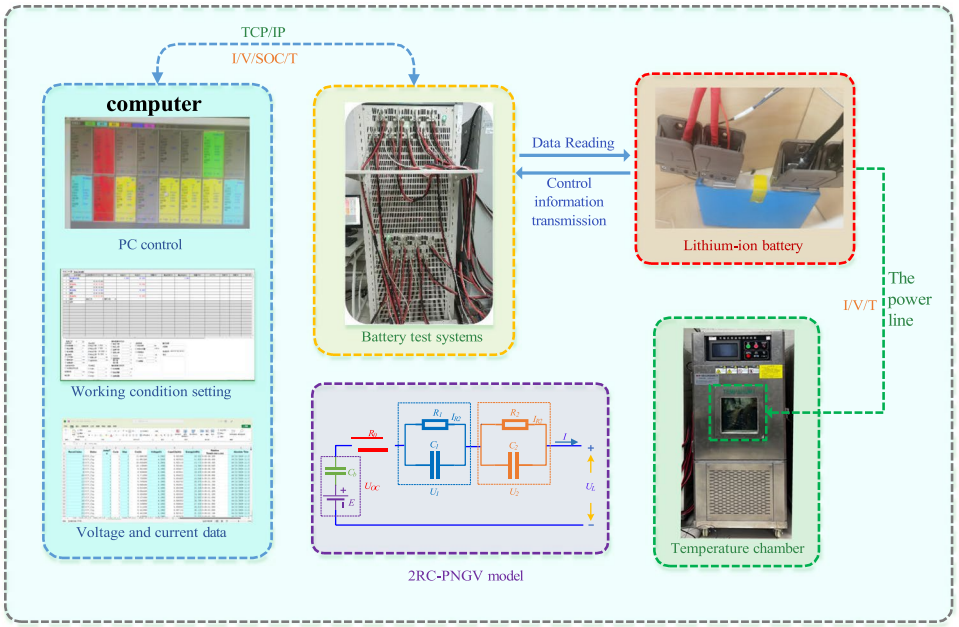
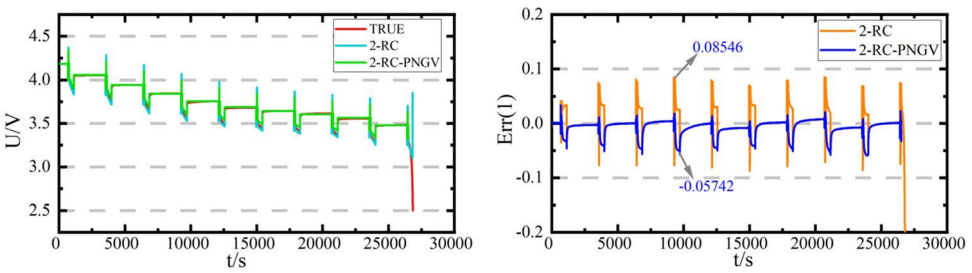


Fig. 7 Voltage curves and errors under different models



(a) Voltage under different models

(b) Voltage error under different models

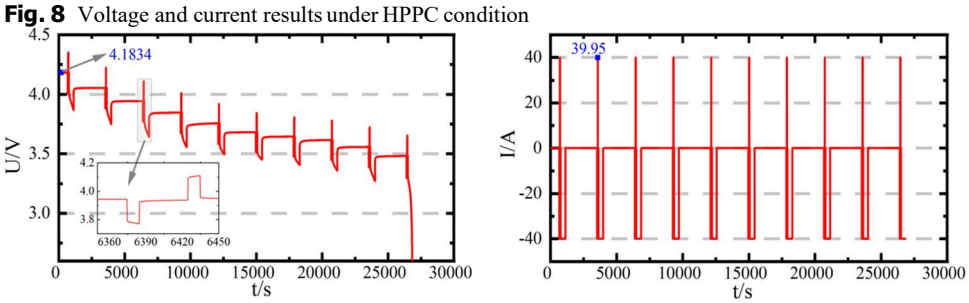
Circuit model comparison experiment

This work compares the second-order RC equivalent circuit model with the second-order RC-PNGV equivalent circuit model experimentally, as shown in Fig. 7, to ascertain if the effect of circuit model selection is appropriate. The voltage output curve and error under identical conditions for the two distinct models are shown in Fig. 7(a) and (b), respectively. The second-order RC- PNGV equivalent circuit model offers more accuracy, a better identification effect, and may produce more accurate data returns when compared to the second-order RC circuit model.

Parameter identification result

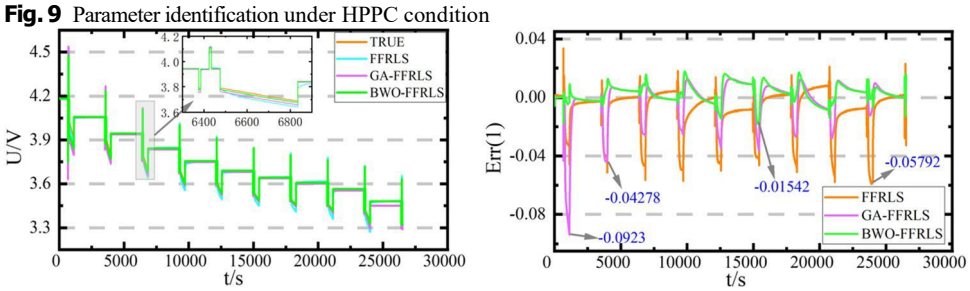
This paper verifies the parameter results under HPPC conditions to obtain the changes in internal lithium-ion parameters at various times and uses the resulting data as the simulation voltage in the model. This allows us to compare and analyze the simulation voltage and the actual measured voltage and, ultimately, confirm the accuracy of the parameter identification. The findings of the real voltage and current data under HPPC working conditions are shown in Fig. 8(a) and (a), respectively; Fig. 9(a) and (b) present, respectively, the outcomes of parameter identification and data errors under HPPC working conditions for three algorithms.

Fig. 8 Voltage and current results under HPPC condition



(a) Voltage under HPPC condition (b) Current under HPPC condition

Fig. 9 Parameter identification under HPPC condition



(a) Voltage under different algorithms (b) Voltage error under different algorithms

To verify the high-precision selection of the pro- posed algorithm, parameter identification results under different algorithms were analyzed, as shown in Table 1 below:

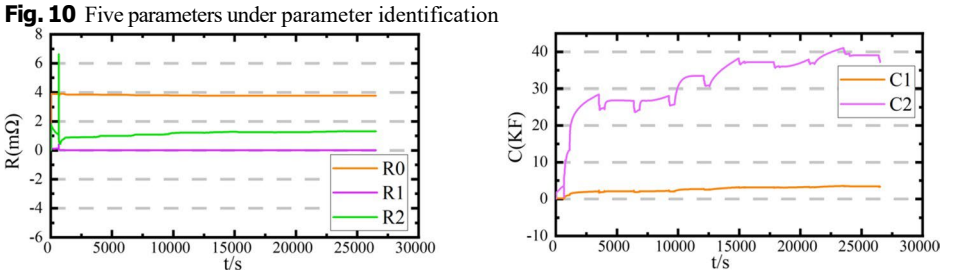
Table 1 Comparison of error data of parameter identification results under different algorithms

Algorithm	Maximum error	*MAE*	*RMSE*
FFRLS	0.05792	0.0115	0.0164
GA-FFRLS	0.0923	0.0074	0.0089
BWO-FFRLS	0.01542	0.0042	0.0068

As can be seen from the result comparison chart and the error data table comparison chart, the FFRLS optimized by BWO has the best effect. Although the GA algorithm also has certain optimality in terms of its complexity, under the premise of the same intelligent algorithm, the GA algorithm requires a large amount of data support and a large amount of computation, while the BWO algorithm has simple computation and good effects. In terms of comprehensive effect, choosing the BWO algorithm to optimize the FFRLS algorithm is more suitable to achieve the results we want, and the experimental result renderings also verify our idea.

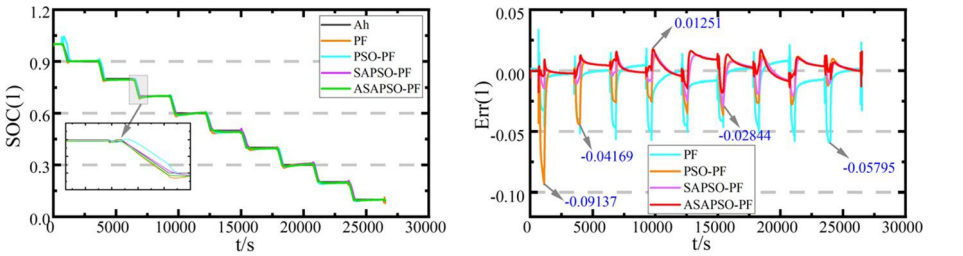
The five parameter graphs for R_0, R_1, R_2, C_1, and C_2 are shown in Fig. 10(a) and (b), respectively. The BWO-FFRLS algorithm identification results may be found to have a significant initial change fluctuation in the result graphs, which is caused by initialization, the passage of time, and the stability of the whale landing probability. Beluga optimization gradually stabilizes the identification results by iteratively choosing the best initial parameter value and forgetting the factor value of time k.

Fig. 10 Five parameters under parameter identification



(a) The value of R under parameter identification (b) The value of C under parameter identification

Fig. 11 *SOC* estimation results and errors under HPPC conditions



(a) Estimates under different algorithms (b) Error of estimated results under different algorithms

Verification of SOC estimation based on ASAPSO-PF algorithm

To verify the accuracy of the algorithm, mixed pulse power characterization (HPPC) data at room temperature (25 °C) was selected to estimate the model. The results of BWO-FFRLS parameter identification were applied to the ASAPSO- PF algorithm for *SOC* estimation, as shown in Fig. 11.

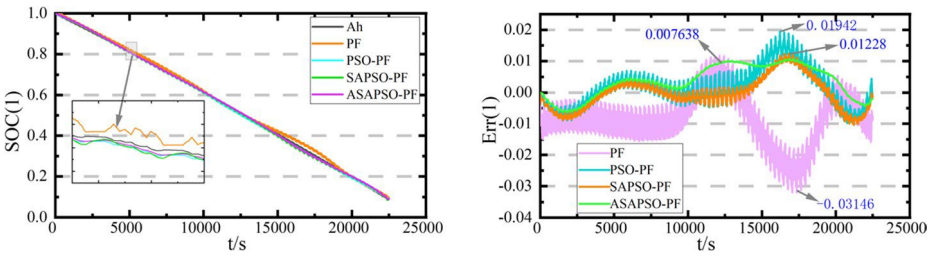
The four algorithms all exhibit a step-like declining tendency over time, as seen in Fig. 11(a). The error comparison chart shows that there is a clear deviation between the estimated value of the PF algorithm and the reference value. The PSO algorithm, which is based on the PF algorithm, decreases error loss. However, the PSO algorithm tends to easily enter the local optimal in the early stages of time. This causes a significant discrepancy between the result and the actual value, and the highest error can reach 9.137%. However, throughout time, the PSO algorithm has undergone some optimization for the PF child poverty problem. The PSO algorithm is optimized by the introduction of the simulated annealing process. The addition of simulated annealing can help with the issue that the PSO method is prone to slip into local optimality at the beginning, as can be seen from the error graph. The inertia weight coefficient is controlled by the hyperbolic tangent function, and the non-linear adaptive change based on simulated annealing reduces error and increases precision. The model was simulated using the BBDST experimental data to further test the reaction of the estimating technique to the level

of charge of lithium batteries under more complicated application situations. While the actual application conditions of high-power lithium batteries are complex and variable, using BBDST experimental data to verify the feasibility of the algorithm will be more convincing. BBDST includes the data of the car in starting, taxing, acceleration, rapid acceleration, and other operations with authenticity and dynamics. Figure 12(a) compares *SOC* estimation outcomes for various algorithms, while Fig. 12(b) compares estimate errors for various algorithms.

It is clear from Fig. 12 that the estimation outcomes of the three algorithms are in line with the functionality of HPPC. The PF algorithm has the highest variation in *SOC* estimating error, as can be observed from the error result chart, and the contrast between the earlier and later fluctuations is somewhat mild. The implementation of PSO can minimize significant PF volatility in the future, but it also has drawbacks of its own. The SA algorithm, which is based on PSO-PF, reduces the experiment's maximum error from 1.942 to 1.228%, and non-linear adaptive modifications bring the estimated value closer to the actual value. Compared with PF, PSO-PF, and SAPSO-PF algorithms, the error curve of the ASAPSO-PF algorithm is relatively stable, and the maximum error is 0.7638%. Therefore, the rationality of the proposed algorithm can also be verified.

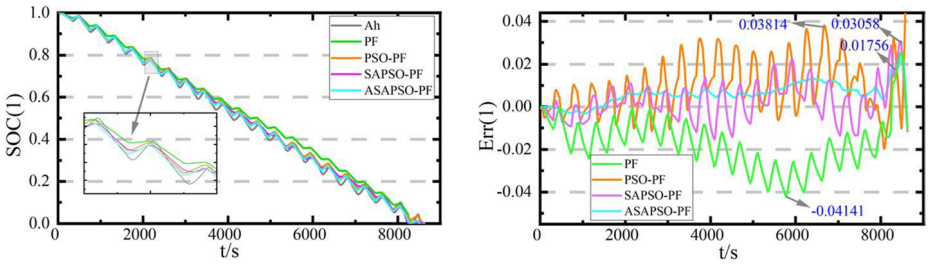
The robustness of our approach during the DST is extremely important for our examination of experimental results. The error curve in Fig. 13 shows that the PF, PSO- PF, and SAPSO-PF algorithms have considerable error variations under DST conditions. Their maximum error is all greater than 3%; however, the fluctuation of the ASAPSO-PF algorithm is minimal, with a maximum error of only 1.756%, which is significantly lower than the estimation of other algorithms and thus validates the algorithm in this work.

Fig. 12 *SOC* estimation results and errors under BBDST conditions



(a) Estimates under different algorithms (b) Error of estimated results under different algorithms

Fig. 13 *SOC* estimation results and errors under DST conditions



(a) Estimates under different algorithms (b) Error of estimated results under different algorithms

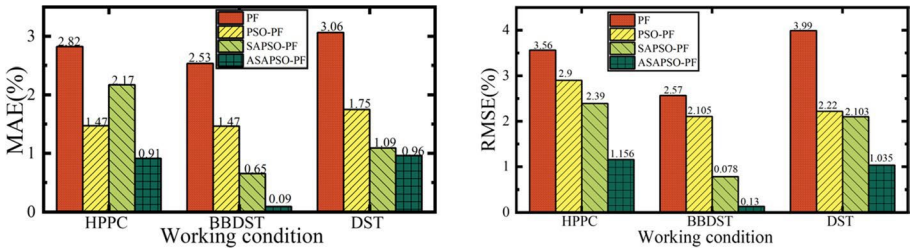
According to the research results of the above three complex working conditions, the ASAPSO-PF algorithm has a high-precision effect on *SOC* estimation. To further confirm that the algorithm in this paper is the correct choice, *MAE* and *RMSE* are chosen to further compare the calculation results. The detailed calculation formula is shown in Eqs. (30) and (31).

$$MAE = \frac{\sum_{i=1}^{N} \left| SOC'(t) - SOC(t) \right|}{N} \tag{30}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(SOC'(t) - SOC(t)\right)^2}{N}} \tag{31}$$

The mean absolute error (*MAE*) represents the mean of the absolute error between the predicted and measured values. The root mean square error (*RMSE*) represents the sample standard deviation of the difference between the predicted value and the measured value and represents the degree of dispersion of the sample, and the size of its value reflects the prediction ability of the algorithm. The results obtained are shown in Fig. 14. It can be seen from the data graph that under three different working conditions, the error result of ASAPSO-PF is the smallest, which verifies the feasibility of the proposed algorithm.

Fig. 14 Comparison of error results under different working conditions



(a) Comparison of MAE estimation results under different working conditions

(b) Comparison of RMSE estimation results under different working conditions

Conclusion

Accurate *SOC* estimation of lithium-ion batteries is the key and most difficult point of lithium-ion battery condition monitoring. Based on the second-order RC-PNGV equivalent circuit model to characterize the state and output characteristics of lithium batteries, a state of charge estimation algorithm based on the beluga optimization-forgetting factor recursive least squares method and an improved particle filter are proposed. Through the experimental analysis, the following conclusions are drawn:

(1) In the process of parameter identification, for the problem of multi-parameter identification of the battery model, compared with the traditional FFRLS algorithm, genetic optimization algorithm, and BWO algorithm combined with the introduction of FFRLS, it can be seen from the result comparison diagram that not only the results of online parameter identification are realized, but also the operation is simple, high precision, and not complicated in the whole identification process. Therefore, it can be concluded that the BWO algorithm combined with the FFRLS algorithm is an innovation for parameter identification.

(2) In the *SOC* estimation of batteries, the ASA algorithm is introduced into the PSO algorithm to solve the sub-poverty problem of traditional PF, and a series of problems such as PSO easy to fall into local optimal and slow convergence speed are well optimized. Under the same conditions and different complex working conditions, different algorithms were verified. From the verification results, we can see that the *SOC* error of the ASAPSO-PF estimation of batteries is controlled within 2%, which further verifies that the ASAPSO-PF algorithm has high accuracy and robust- ness in the *SOC* estimation of lithium batteries.

(3) Through the verification of the whole experiment process in this paper, it is known that it is feasible to combine the BWO algorithm with FFRLS for parameter identification, and the application of the ASAPSO-PF algorithm to estimate the *SOC* accuracy of the battery has also achieved a good optimization effect. Therefore, in future work, we can try to apply this method to other studies and have deeper thinking about the development of future lithium-ion batteries.

Author contributions All authors contributed to the conception and design of the study. Material preparation, data collection, and analysis were conducted by Xianfeng Shen, Zehao Li, and Chuangsi Qi. Shunli Wang and Chunmei Yu guided the construction of the paper. The first draft of the manuscript was written by Xianfeng Shen. Carlos optimized the grammatical format of the full paper. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

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

References

1. Marom R, Amalraj SF, Leifer N, Jacob D, Aurbach D (2011) A review of advanced and practical lithium battery materials. J Mater Chem 21:9938–9954. https://doi.org/10.1039/C0JM04225K
2. Scrosati B, Garche J (2010) Lithium batteries: status, prospects and future. J Power Sources 195:2419–2430. https://doi.org/10.1016/j.jpowsour.2009.11.048
3. 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
4. Huang B, Pan Z, Su X, An L (2018) Recycling of lithium-ion batteries: recent advances and perspectives. J Power Sources 399:274–286. https://doi.org/10.1016/j.jpowsour.2018.07.116
5. Ni Z, Li B, Yang Y (2023) Deep domain adaptation network for transfer learning of state of charge estimation among batteries. J Energy Storage 61:106812. https://doi.org/10.1016/j.est2023.106812
6. Zhang J, Ren H, Wang J, Qi J, Yu R, Wang D, Liu Y (2016) Engineering of multi-shelled SnO_2 hollow microspheres for highly stable lithium-ion batteries. J Mater Chem A 4:17673–17677. https://doi.org/10.1039/c6ta07717j
7. Bao X, Liu Y, Liu B, Liu H, Wang Y (2023) Multi-state online estimation of lithium-ion batteries based on multi-task learning. Energies 16:3002. https://doi.org/10.3390/en16073002
8. Lian G, Ye M, Wang Q, Wei M, Ma Y (2023) Noise-immune state of charge estimation for lithium-ion batteries based on optimized dynamic model and improved adaptive unscented Kalman filter under wide temperature range. J Energy Storage 64:107223. https://doi.org/10.1016/j.est.2023.107223
9. Sun S, Gao Z, Jia K (2023) State of charge estimation of lithium-ion battery based on improved Hausdorff gradient using wavelet neural networks. J Energy Storage 64:107184. https://doi.org/10.1016/j.est.2023.107184
10. Saqli K, Bouchareb H, M'sirdi NK, Bentaie MO (2023) Lithium- ion battery electrothermal modelling and internal states co-estimation for electric vehicles. J Energy Storage 63:107072. https:// doi.org/10.1016/j.est.2023.107072
11. Khalid A, Kashif SAR, Ul Ain N, Awais M, Smieee MA, Carreno JEM, Vasquez JCC, Guerrero JMM, Khan B (2023) Comparison of Kalman filters for state estimation based on computational complexity of Li-ion cells. Energies 16:2710. https://doi.org/10.3390/en16062710
12. Wang D, Yang Y, Gu T (2023) A hierarchical adaptive extended Kalman filter algorithm for lithium-ion battery state of charge estimation. J Energy Storage 62:106831. https://doi.org/10.1016/j.est.2023.106831
13. Hong S, Qin C, Lai X, Meng Z, Dai H (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
14. Wang J, Meng J, Peng Q, Liu T, Zeng X, Chen G, Li Y (2023) Lithium-ion battery state-of-charge estimation using electrochemical model with sensitive parameters adjustment. Batteries-Basel 9:180. https://doi.org/10.3390/batteries9030180
15. Hao X, Wang S, Fan Y, Xie Y, Fernandez C (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
16. Pang H, Geng Y, Liu X, Wu L (2022) A composite state of charge estimation for electric vehicle lithium-ion batteries using back-propaegation neural network and extended Kalman particle filter. J Electro- chem. Soc 169:110516. https://doi.org/10.1149/1945-7111/ac9f79
17. Xue A, Yang W, Yuan X, Yu B, Pan C (2022) Estimating state of health of lithium-ion batteries based on generalized regression neural network and quantum genetic algorithm. Appl Soft Comput 130:109688. https://doi.org/10.1016/j.asoc.2022.109688
18. Li H, Qu Z, Xu T, Wang Y, Fan X, Jiang H, Yuan C, Chen L (2022) SOC estimation based on the gas-liquid dynamics model using particle filter algorithm. Int J Energy Res 46:22913–22925. https:// doi.org/10.1002/er.8594
19. Geng Y, Pang H, Liu X (2022) State-of-charge estimation for lithium-ion battery based on PNGV model and particle filter algorithm. J Power Electron 22:1154–1164. https://doi.org/10.1007/s43236-022-00422-0
20. Chen D, Meng J, Huang H, Wu J, Liu P, Lu J, Liu T (2022) An empirical-data hybrid driven approach for remaining useful life prediction of lithium-ion batteries considering capacity diving. Energy 245:123222. https://doi.org/10.1016/j.energy.2022.123222
21. Particle Swarm Optimization (PSO) (2015) A tutorial. Chemometrics and Intelligent Laboratory Systems 149:153–165. https://doi.org/10.1016/j.chemolab.2015.08.020
22. Mei Y, Li B, Wang H, Wang X, Negnevitsky M (2022) Multi-objective optimal scheduling of microgrid with electric vehicles. Energy Rep 8:4512–4524. https://doi.org/10.1016/j.egyr.2022.03.131
23. Wen D, Shi C, Zhang Y, Liao K, Liu J, Luo B, Wang T (2023) A novel multi-strategy self-optimizing SAPSO algorithm for PMSM parameter identification. IET Power Electron 16:305–319. https://doi.org/10.1049/pel2.12385
24. Yi M, Xie W, Mo L (2021) Short-term electricity price forecasting based on BP neural network optimized by SAPSO. Energies 14:6514. https://doi.org/10.3390/en14206514
25. Zhu H, Xue X, Geng A, Ren H (2021) Matching sensor ontologies with simulated annealing particle swarm optimization. Mob Inf Syst 2021:5510055. https://doi.org/10.1155/2021/5510055
26. Yi M, Mo L, Shen Q (2020) Study on generation scheduling of cascade hydropower stations based on SAPSO. J Coast Res:371–378. https://doi.org/10.2112/JCR-SI104-066.1
27. Diao Y, Ma H, Wang H, Wang J, Li S, Li X, Pan J, Qiu Q (2022) Optimal flood-control operation of cascade reservoirs using an improved particle swarm optimization algorithm. Water 14:1239. https://doi.org/10.3390/w14081239
28. Lu J, Zhang Z (2021) An improved simulated annealing parti- cle swarm optimization algorithm for path planning of mobile robots using mutation particles. Wirel Commun Mob Comput 2021:2374712. https://doi.org/10.1155/2021/2374712
29. Wang E, Sun C, Wang C, Qu P, Huang Y, Pang T (2021) A satellite selection algorithm based on adaptive simulated annealing particle swarm optimization for the BeiDou navigation satellite system/global positioning system receiver. Int J Distrib Sens Netw 17:15501477211031748. https://doi.org/10.1177/15501477211031748

30. Li R, Yu J, Li J, Chen F (2015) Equivalent model and parameter identification of lithium-ion battery. In: Deng Z, Li H (eds) Proceedings of the 2015 Chinese Intelligent Automation Conference: Intelligent Technology and Systems, vol 338. Springer, New York, pp 29–39. https://doi.org/10.1007/978-3-662-46466-3_4
31. El Ghossein N, Salameh JP, Karami N, El Hassan M, Najjar MB (2015) Survey on electrical modeling methods applied on different battery types. In: Proceedings of the 2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (taeece). Ieee, New York, pp 39–44. https://doi.org/10.1109/TAEECE.2015.7113597
32. Xiong W, Mo Y, Zhang F (2019) Lithium-ion battery modeling and state of charge estimation. Integr Ferroelectr 200:59–72. https://doi.org/10.1080/10584587.2019.1592620
33. Liu D, Fan Y, Wang S, Xia L, Qiu J, Bobobee ED (2021) A novel fading memory recursive least square method (FMLS) for accurate state of charge estimation of lithiumion batteries combined with improved second order PNGV modeling. Int J Electrochem Sci 16:21097. https://doi.org/10.20964/2021.09.34
34. Ge C, Zheng Y, Yu Y (2022) State of charge estimation of lithium-ion battery based on improved forgetting factor recursive least squares-extended Kalman filter joint algorithm. J Energy Storage 55:105474. https://doi.org/10.1016/j.est.2022.105474
35. Ouyang T, Xu P, Chen J, Lu J, Chen N (2020) Improved parameters identification and state of charge estimation for lithium-ion battery with real-time optimal forgetting factor. Electrochimica Acta 353:136576. https://doi.org/10.1016/j.electacta.2020.136576