A novel gaussian particle swarms optimized particle filter algorithm for the state of charge estimation of lithium-ion batteries.

WANG, X., WANG, S., HUANG, J., FERNANDEZ, C., ZHOU, Y. and CHEN, L.

2020

Int. J. Electrochem. Sci., 15 **(2020)** 10632 – 10648, doi: 1020964/2020.10.21

International Journal of
ELECTROCHEMICAL
SCIENCE
www.electrochemsci.org

A Novel Gaussian Particle Swarms optimized Particle Filter Algorithm for the State of Charge Estimation of Lithium-ion Batteries

Xinyang Wang[1], *Shunli Wang*[1,*], *Junhan Huang*[1], *Carlos Fernandez*[2], *Yicong Zhou*[1], *Lei Chen*[1]

[1] School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China;
[2] School of Pharmacy and Life Sciences, Robert Gordon University, Aberdeen AB10-7GJ, UK.
[*]E-mail: 497420789@qq.com

Received: 1 June 2020 / *Accepted:* 17 Jul 2020 / *Published:* 31 August 2020

A gaussian particle swarm optimized particle filter estimation method, along with the second-order resistance-capacitance model, is proposed for the state of charge estimation of lithium-ion battery in electric vehicles. Based on the particle filter method, it exploits the strong optimality-seeking ability of the particle swarm algorithm, suppressing algorithm degradation and particle impoverishment by improving the importance distribution. This method also introduces normally distributed decay inertia weights to enhance the global search capability of the particle swarm optimization algorithm, which improves the convergence of this estimation method. As can be known from the experimental results that the proposed method has stronger robustness and higher filter efficiency with the estimation error steadily maintained within 0.89% in the constant current discharge experiment. This method is insensitive to the initial amount and distribution of particles, achieving adaptive and stable tracking in the state of charge for lithium-ion batteries.

Keywords: Lithium-ion Battery; State of Charge; Particle Filter; Particle Swarm Optimization; Importance Resampling

1. INTRODUCTION

As the environmental issues have become more and more prominent in recent years, clean energy has greatly promoted the development of the new energy vehicle industry. The lithium-ion battery, with its incomparable advantages, has become the core power source for Electric Vehicles (EVs) [1]. The State of Charge (SOC) is an important monitoring parameter to the battery management system, which determines its safety and emerge distribution [3,4].

Since lithium-ion batteries exhibit a high degree of nonlinearity during actual use [5,6], it is unlikely to be able to directly monitor their SOC, and there is currently no mature and well-developed

SOC estimation method for the nonlinear system [7]. Nevertheless, Particle Filter (PF) is gradually becoming a research hotspot because of its unique advantage in handling nonlinear systems without the constraints of assumptions such as noise and models [8,9]. In PF algorithm, however, the problem of algorithm degradation [10] and sample depletion [11] can easily occur, resulting in large SOC estimation errors for lithium-ion batteries. In recent years, researchers have proposed many improved algorithms for SOC estimation to deal with the shortcomings of PF algorithm [12-24]. By improving the proposed distribution function, in reference [25-28], Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are incorporated into the PF algorithm to generate the proposed distribution function, which adaptively adjusts the particle weights according to the measured noise characteristics to achieve high-precision tracking effect. In reference [19], the estimation accuracy is improved by incorporating an adaptive genetic algorithm into the resampling process, which uses crossover and variation to reduce the number of resamples. In reference [29], a multi-strategy differential operation with the help of the cuckoo algorithm improves particle diversity and thus suppresses the problem of the algorithm. In addition, researchers also have incorporated artificial immunization [30], bat algorithms [31], and fish optimization algorithms [32] into the traditional resampling process of PF algorithms to improve filtering efficiency and state estimation accuracy, which achieves good tracking results as well.

In order to suppress the algorithmic degradation of the PF and improve the estimation accuracy of the SOC, a novel GPSO-PF method is proposed to realize the state monitoring of lithium-ion battery under complex current variation conditions. It mainly uses PSO to guide particles in the low likelihood region of PF moving towards the high likelihood region, and then iteratively updates the velocity and position of the particles with normally distributed decaying inertia weights. The particles are then filtered by resampling after updating, which obtains the minimum mean square estimation of the SOC. Finally, combined with the analysis of the influencing factors, it is experimentally verified that the improved iterative calculation algorithm can effectively improve the accuracy and robustness of lithium-ion battery.

2. MATHEMATICAL ANALYSIS

2.1. Equivalent circuit model

In the use of lithium-ion batteries, an accurate model is crucial for SOC estimation because of its complex internal physicochemical reactions. The electrochemical model obtains mathematical expressions of the battery's kinetic properties by experimentally analyzing its mechanism. Although the model is highly accurate and can represent the battery kinetic properties well, the model structure is complex and has many parameters, which is not suitable for real-time system [33]. While the Equivalent Circuit Model (ECM) stands out for its simple structure and fast implementation [34]. Considering the accuracy and complexity of the model [35], a more accurate and intuitive 2nd-order RC equivalent circuit model [36] is used in this paper to achieve real-time SOC estimation. Its model structure is shown in Figure 1.

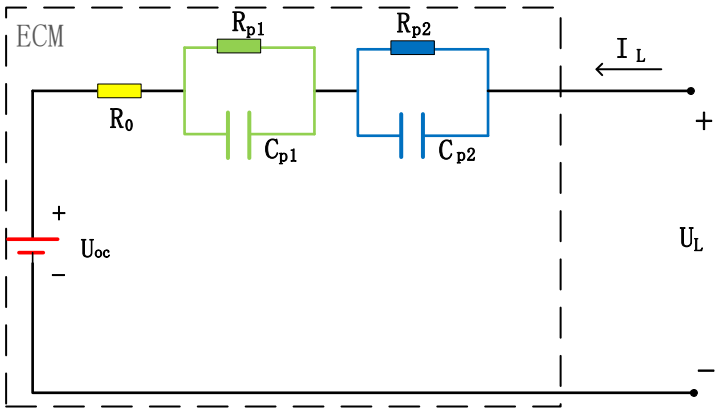


Figure 1. 2nd-order RC equivalent circuit model

As shown in the Figure 1, U_{OC} represents the open circuit voltage of the battery. I_L represents the inflow current value of the lithium-ion battery connected to the external circuit. R_0 represents the ohmic resistance of the battery, which characterizes the voltage drop at the moment of battery charging and discharging. The RC parallel circuit consisting of R_{p1} polarization resistance and C_{p1} polarization capacitance characterizes the electrochemical polarization effect. The RC parallel circuit consisting of R_{p2} diffusion resistance and C_{p2} diffusion capacitance characterizes the differential polarization effect. According Kirchhoff's law, the relationship between each component in the model and SOC can then be described.

2.2. State-space mathematical description

The state-space equation is the basis for SOC estimation. Set the current direction to be positive when the lithium-ion battery is discharged. According to the equivalent circuit model as shown in Figure 1, combined with Kirchhoff's law can be obtained formula (1):

$$U_L(t) = U_{OC}(t) - i(t)R_0(t) - U_{p1}(t) - U_{p2}(t) \tag{1}$$

When charging and discharging the lithium-ion battery, the RC parallel circuit is used to characterize the polarization effect, which is essentially represented by the zero state and zero input response process of the RC parallel circuit. The formula calculating the voltage U_{p1} and U_{p2} of the RC parallel circuits is shown in formula (2).

$$\begin{cases} \dfrac{dU_{p1}}{dt} = -\dfrac{U_{p1}}{R_{p1}C_{p1}} + \dfrac{i}{C_{p1}} \\ \dfrac{dU_{p2}}{dt} = -\dfrac{U_{p2}}{R_{p2}C_{p2}} + \dfrac{i}{C_{p2}} \end{cases} \tag{2}$$

Then, based on the definition of the battery's SOC, also known as the Ampere hour (Ah) integral method, the remaining capacity of the battery is calculated by accumulating the incoming and outgoing charges in real time. The ratio between the remaining capacity and the nominal capacity of the battery is the SOC. Its mathematical expression is shown in formula (3).

$$SOC_t = SOC_0 - \frac{1}{C_N}\int_0^t \eta I(t)dt \tag{3}$$

SOC_0 is the initial charge of the battery, SOC_t is the remaining charge of the battery at time t; η is the charge/discharge efficiency; C_N is the nominal capacity of the battery at room temperature; I(t) is the discharge current. According to formula (3), let τ to be the time constant of the RC circuit, where $\tau=RC$. Then combine equations (1), (2) and (3), the state and observation equations of the lithium-ion battery are obtained after discretization, as shown in formula (4) and formula (5).

$$\begin{bmatrix} SOC_k \\ U_{p1,k} \\ U_{p2,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{-T/\tau_1} & 0 \\ 0 & 0 & e^{-T/\tau_2} \end{bmatrix} \begin{bmatrix} SOC_{k-1} \\ U_{p1,k-1} \\ U_{p2,k-1} \end{bmatrix} + \begin{bmatrix} -\frac{\eta T}{C_n} \\ R_{p1}(1-e^{-T/\tau_1}) \\ R_{p2}(1-e^{-T/\tau_2}) \end{bmatrix} I_{k-1} \tag{4}$$

$$U_{L,k} = U_{OC(SOC,k)} - U_{p1,k} - U_{p2,k} - R_0 I_k \tag{5}$$

T_s is the integral sampling time interval. The $U_{OC(SOC,k)}$ is the open circuit voltage in the model at time k. Each parameter in the model is a function of the SOC, which is identified by HPPC experiments. The mainstream estimation methods for lithium-ion batteries are based on the Ah integral method. However, the Ah integral method depends on the initial SOC value and the calculation of the power consumption, which is known from formula (3), in terms of noise, temperature, instantaneous high current and power outages, it is not possible to accurately estimate SOC in practice. In this paper, the GPSO method is used to improve the accuracy and robustness of the SOC estimation by addressing the nonlinearity and the effect of the initial SOC values of lithium-ion batteries.

3. GPSO-PF BASED SOC ESTIMATION

3.1. PF based SOC estimation

The PF algorithm is an algorithm that implements Recursive Bayesian Theorem (RBT) based on a non-parametric Monte Carlo (MN) method simulation [37]. The basic idea is to use a large number of discrete random samples to approximate the probability function of the system's random variables according to the Law of Large Numbers (LOLN), substituting the sample mean value for the integral operation. It firstly generates a set of random particles $p(x_0)$ through the empirical distribution of the system state, and obtains the a priori probability density distribution $p(x_k^i|x_{k-1}^i)$ after importance sampling. The weights w_k^i and positions of each particle are then continuously adjusted according to the Bayesian filtering principle. Combined with the latest observations y_k^i, the true prior probability density of the state variables is approximated by resampling. The PF flowchart is shown in Figure 2.

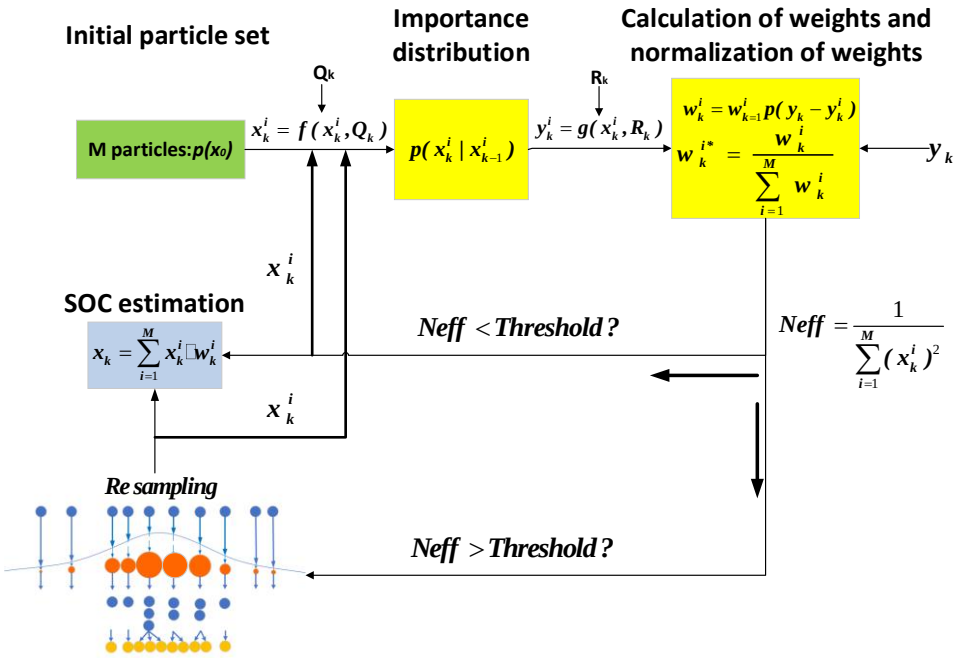


Figure 2. PF algorithm flowchart

In the traditional PF algorithm, its estimation accuracy is often affected by the initial particle distribution of the system. And in the resampling process, it selects and replicates high-weighted particles multiple times, while ignoring low-weighted particles, resulting in the problem of impoverished particle diversity and algorithmic degradation arises after several resamplings [38]. In order to alleviate the algorithm degradation and improve the robustness and accuracy of SOC estimation for nonlinear systems, the PSO algorithm is introduced to improve the resampling process of PF algorithm.

3.2. Gaussian particle swarm optimization algorithm theory

The particle swarm algorithm is an intelligent optimization algorithm [39] that simulates the movement of a bird flock. The basic idea is to iteratively update the position of the particle swarm according to the position, velocity and fitness between each particle, to achieve the purpose of optimization. Wherein, the position of the particle determines the direction of the particle motion and the velocity magnitude determines the distance of the particle motion. In the basic particle swarm optimization algorithm, as the particles move towards the optimal solution, the phenomenon of local non-convergence tends to occur [40]. In order to improve the convergence of PSO, a GPSO method is adopted in this paper that uses Gaussian function distribution, localization and other features to adjust the inertia weights nonlinearly, and its convergence is better than the PSO algorithm [41], the mathematical expression of which is shown in equation (6) and equation (7).

$$v_k^i = |rand|(p_{best} - x_k^i) + |Rand|(g_{best} - x_k^i) \quad (6)$$

$$x_k^i = x_k^i + v_k^i \quad (7)$$

where v_k^i indicates the speed of movement and x_k^i indicates the position of movement. p_{best} is

the particle individual optimum of the iteration of particles, and g_{best} is the particle set optimum. $|rand|$ and $|Rand|$ are positive Gaussian distributed random numbers. In this paper, the sampling process of PF was optimized using PSO and the resampling process is shown in Figure 3.

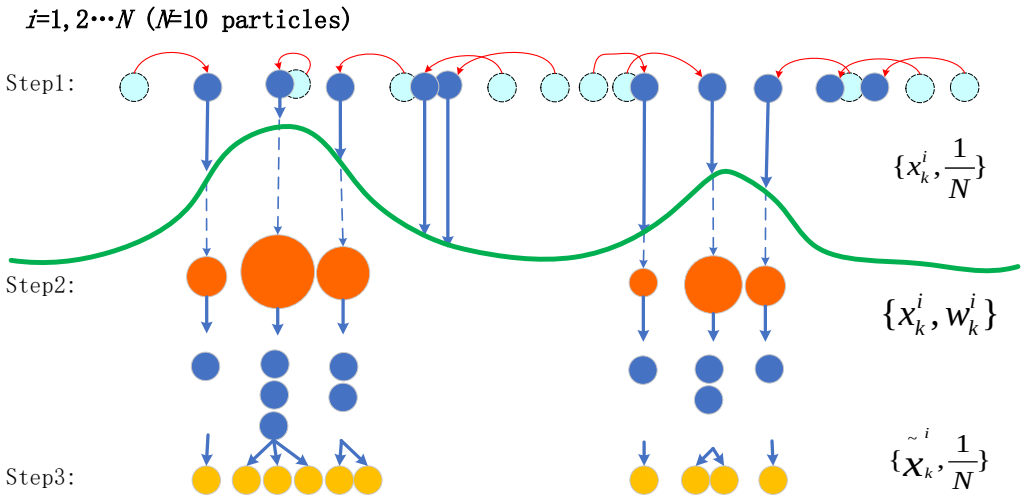


Figure 3. Sampling process diagram of particle swarm optimized particle filter

In the particle swarm algorithm, the offset occurs whenever there is a difference in fitness between the particles. If the set of particles is concentrated near the true value, this offset will instead reduce the diversity of particles. After applying the GPSO algorithm, the particles with less fitness move over a larger distance, and the particles with more fitness move over a smaller distance. And keeping a certain relative relationship between the particles, iterations into the update are repeated to guide the particles towards the state true value with a higher probability of occurrence. The region moves, and if the optimization threshold is reached, or if the set of particles is close to the real state, the optimization is stopped.

3.3. GPSO-PF based SOC estimation iterate calculation

First, the equation of the state-space of the model is obtained on the basis of the constructed equivalent circuit model. An initial set of particles is generated from the empirical distribution, importance sampling is performed by the state transfer equation, and then the Gaussian particle population is optimized for the importance distribution and resampled when the effective particle number threshold is not met. The process is as Figure 4.

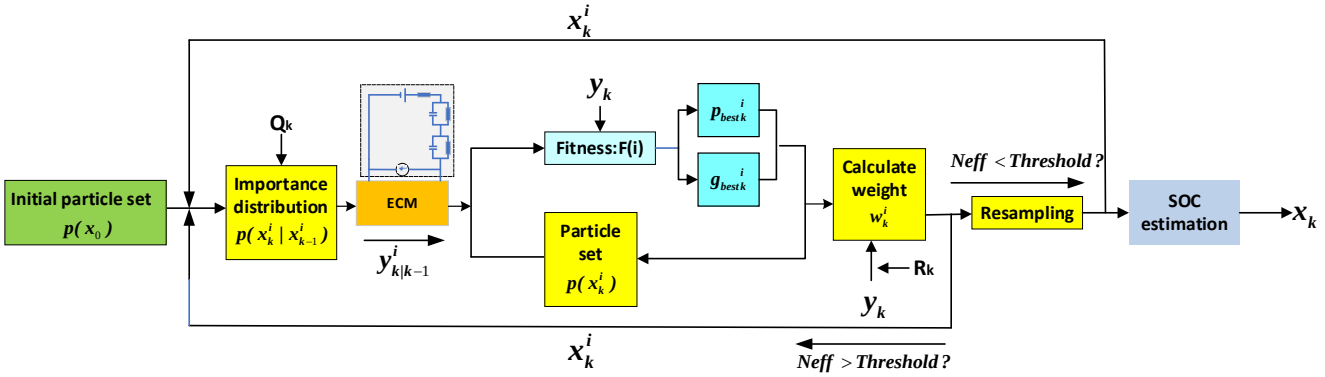


Figure 4. Framework of particle filtering algorithm for particle swarm optimization

The combination of the equation (4) and the equation (5) uses $[SOC\ ;\ U_{p1};\ U_{p2}]$ as the state variable x_k and U_L as the observation variable y_k. The equations of state and observation of the algorithm can be expressed as the equation (8).

$$\begin{cases} \boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, u_k, Q_k) \\ \boldsymbol{y}_k = h(\boldsymbol{x}_k, u_k, R_k) \end{cases} \tag{8}$$

Where Q_k is process noise; R_k is observation noise; u_k is current input. $f(x_{k-1},\ u_k,\ Q_k)$ is the state transfer equation; $h(x_k,\ u_k,\ R_k)$ is the observational equation. The specific algorithm steps for the SOC state estimation of lithium-ion batteries are as follows.

（1） Define the fitness function.

$$Fit[i] = \exp[-\frac{1}{2R_k}(y_k - y_{k|k-1}^i)^2] \tag{9}$$

y_k is the true end voltage measurement at moment k; $y_{k|k-1}^i$ is the algorithm prediction measurement.

（2） Initialization: k=0.

M particles $\{\boldsymbol{x}_0^i\}_{i=1}^M$ in the state space according to the empirical condition distribution of the system state vector p(x_0), with all particle weights of $\{w_0^i\}_{i=1}^M = \frac{1}{M}$.

（3） Algorithmic loop process: make k=1, 2. . .

① M particles are randomly selected from the importance density function, which is the state transfer probability density function.

$$\boldsymbol{x}_k^i \square\ q(\boldsymbol{x}_k^i \mid \boldsymbol{x}_{k-1}^i, y_k) = p(\boldsymbol{x}_k^i \mid \boldsymbol{x}_{k-1}^i), i = 1, 2, \ldots, \mathrm{M} \tag{10}$$

② Particle swarm optimization.

a) The Fit[i] of each particle is obtained by calculating the fit value of each particle from the equation (9).

b) Compare each particle with its fit value Fit[i] and individual pole value $p_{best}(i)$, and if Fit[i] >$p_{best}(i)$, then replace $p_{best}(i)$ with Fit[i].

c) Compare each particle with its fit value Fit[i] and global pole value $g_{best}(i)$, and if Fit[i] > $g_{best}(i)$, then replace $g_{best}(i)$ with Fit[i].

d) Update particle position $\boldsymbol{x}_k^i$ and velocity $\boldsymbol{v}_k^i$ according to equation (6) and equation (7).

e) Whether the particle set is distributed and concentrated or whether the maximum number of iterations is reached, otherwise a) is returned.

③ Calculation and normalization of particle weights.

$$w_k^i = w_{k-1}^i p(y_k \mid \boldsymbol{x}_k^i) = w_{k-1}^i p(y_k - y_{k|k-1}^i), i = 1, 2\ldots, \mathrm{M} \tag{11}$$

$$w_k^i = w_k^i / \sum_{i=1}^{M} w_k^i \tag{12}$$

④ Resampling to calculate effective sample size.

$$N_{eff} = 1 / \sum_{i=1}^{M} (w_k^i)^2 \tag{13}$$

⑤ Status estimate.

$$\hat{\boldsymbol{x}}_k = \sum_{i=1}^{M} w_k^i \boldsymbol{x}_k^i \tag{14}$$

⑥ Determine if the cycle is over, and if it is not over k=k+1, continue the cycle.

4. EXPERIMENTAL ANALYSIS

4.1. Battery testing platform

The test platform for the battery is shown in Figure 5, which consists of the following components. (1) Battery. The positive and negative poles of the battery are connected separately through the battery test system, with black being the negative pole and red being the positive pole. (2) Host computer. The battery charge/discharge multiplier, duration and other process conditions are set by the host computer. (3) The battery test system NEWARE BTS-4000, which has 16 charge and discharge ports, transmits battery information to the host computer in real time during testing. (4) Thermostatic box. The temperature control panel sets specific operating temperature conditions for the battery and is suitable for experiments with temperature requirements.

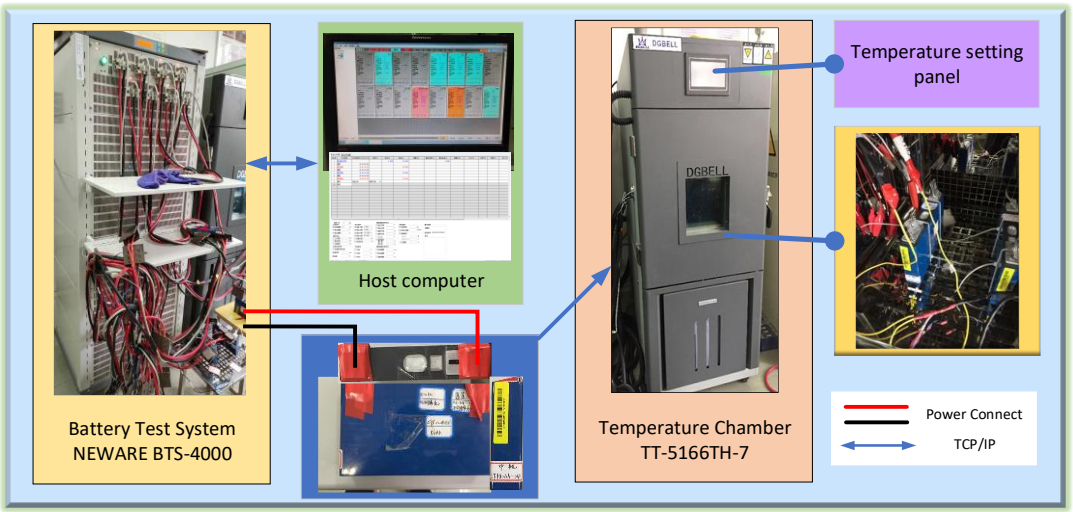


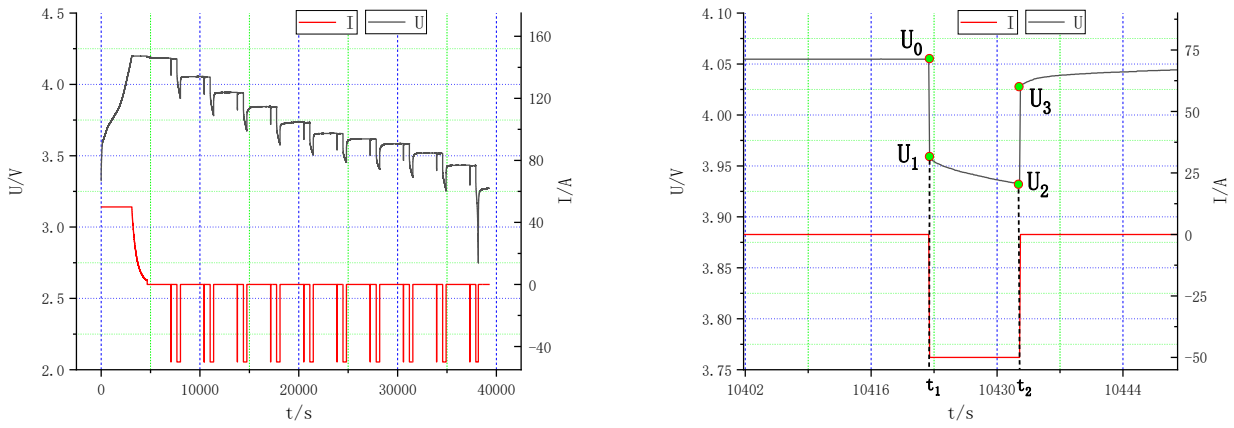
Figure 5. Battery test bench

During the experimental operation, the lithium-ion battery is first connected to the battery test system and then placed in the thermostat. The battery's operating steps, temperature and duration are set by the host computer. At the end of the experiment, the voltage, current, and charge/discharge multiplier of the battery can be seen in the history of the host computer during the whole operation.

4.2. Experimental validation and analysis

4.2.1. Parameter Recognition Experiment

In order to obtain the 2nd-order RC model parameters as a function of SOC, the lithium-phosphate battery LFP50Ah was tested at room temperature at 25°C by the BTS-4000 [42]. The experimental procedure was set up as follows: first, discharge the battery at 1C (50A current) for 10 seconds and then leave it alone for 10 minutes. Then, discharge the battery at 1C for 5 minutes and 50 seconds, and then leave it for 40 minutes. Cycle the above two steps until the SOC is 0. The HPPC experimental current and voltage diagram as shown in Figure 6.



(a) HPPC test voltage and current diagram (b) SOC=0.9

Figure 6. HPPC test

Figure 6 (a) shows the voltage and current variation curves through the HPPC test. Figure 6 (b) shows the pulse discharge voltage curve at SOC=0.9. After 40 minutes of resting, of which the battery is almost stable internally, the voltage U_0 is the open circuit voltage U_{OC} [43]. The fit function of U_{OC} is the equation (15), a polynomial function relationship between the model parameters and the SOC was fitted by MATLAB, with a goodness-of-fit greater than 0.999.

$$\begin{aligned} U_{OC}(SOC) = &-535.7*SOC^9 + 233.5*SOC^8 - 431.3*SOC^7 + 446.4*SOC^6 \\ &-290.7*SOC^5 + 126.6*SOC^4 - 371.2*SOC^3 + 67.6*SOC^2 \\ &-5.7*SOC + 3.6 \end{aligned} \tag{15}$$

At the moment of t_1 and t_2, it is the voltage mutation at the end of the lithium-ion battery caused by the ohmic resistance R_0, which is calculated as shown in formula (16). According to the polynomial fitting, the fitting function of R_0 can be obtained as shown by the equation (17).

$$\boldsymbol{R_0} = \frac{(\boldsymbol{U}_0 - \boldsymbol{U}_1) + (\boldsymbol{U}_3 - \boldsymbol{U}_2)}{2 * \boldsymbol{I}} \tag{16}$$

$$\begin{aligned} R_0(SOC) = &-0.0077 * SOC^6 + 0.01797 * SOC^5 - 0.0095 * SOC^4 - 0.0065 * SOC^3 \\ &+0.0085 * SOC^2 - 0.0031 * SOC + 0.0023 \end{aligned} \tag{17}$$

In the period from t_1 to t_2, the terminal voltage of the lithium-ion battery slowly decreases from U_1 to U_2. This is the process by which the discharge current charges the polarized capacitor, which is the zero-state response phase of the dual RC circuit. The terminal voltage equation is shown by the equation (18).

$$\boldsymbol{U_L = U_{OC} - R_0 I - (1 - e^{-t/\tau_1}) R_{p1} I - (1 - e^{-t/\tau_2}) R_{p2} I} \tag{18}$$

In the above equation, τ represents the time constant, τ=RC. The battery terminal voltages from t_1 to t_2 are collected by MATLAB, and then the values of R_{p1}, R_{p2}, C_{p1} and C_{p2} are identified by cftool toolbox. After the above steps, the values of each parameter under different SOC can be identified [44], as shown in

Table **1**.

Table 1. Parameter identification values under different SOC states

SOC/100%	U_{OC}/V	R_0/mΩ	C_{p1}/kF	C_{p2}/kF	R_{p1}/mΩ	R_{p2}/mΩ
1.0	4.184	1.147	55.628	28.413	0.332	0.254
0.9	4.052	1.136	47.944	25.724	0.148	0.276
0.8	3.936	1.141	32.500	34.599	0.244	0.227
0.7	3.830	1.147	55.448	22.099	0.138	0.346
0.6	3.724	1.151	46.985	25.260	0.158	0.295
0.5	3.649	1.174	42.297	45.495	0.169	0.157
0.4	3.615	1.202	58.314	36.132	0.129	0.208
0.3	3.589	1.236	49.825	32.671	0.137	0.209
0.2	3.537	1.306	4.687	24.426	0.079	0.571

Similarly, the equations of each parameter regarding SOC are fitted by MATLAB and brought into the 2-order model to obtain the observed and measured equations for the SOC of the lithium-ion battery model. The fitted parameter equations are shown in equations (19), (20), (21) and (22).

$$\begin{aligned} R_{p1}(SOC) = &-0.0442 * SOC^7 + 0.1728 * SOC^6 - 0.2511 * SOC^5 + 0.1641 * SOC^4 \\ &-0.0458 * SOC^3 + 0.0057 * SOC^2 - 0.00173 * SOC + 7.20 \end{aligned} \tag{19}$$

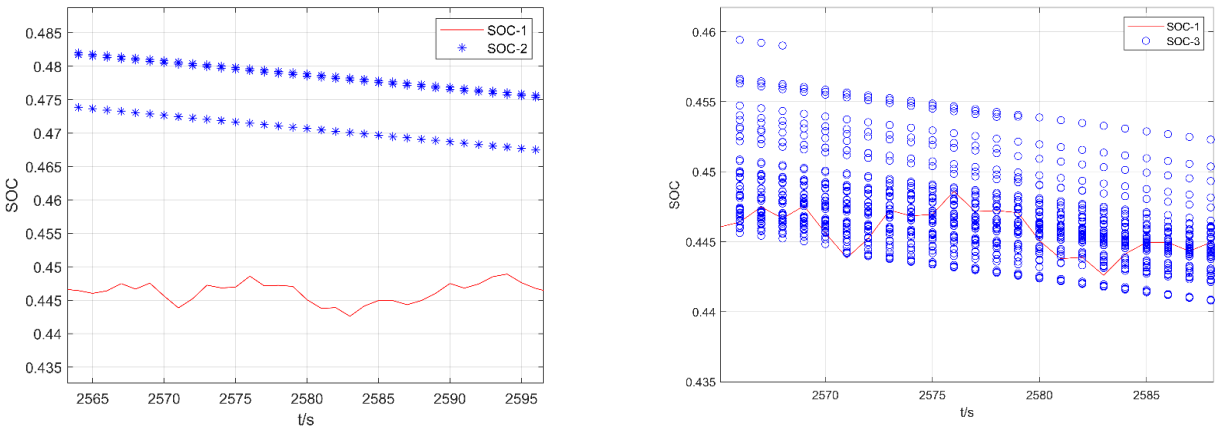
$$\begin{aligned} R_{p2}(SOC) = &0.5429 * SOC^8 - 2.8102 * SOC^7 + 6.0693 * SOC^6 - 7.1011 * SOC^5 \\ &+4.8842 * SOC^4 - 2.0010 * SOC^3 + 0.4701 * SOC^2 - 0.0566 * SOC \\ &+0.0027 \end{aligned} \tag{20}$$

$$\begin{aligned} C_{p1}(SOC) = &8.6976 * 10^6 * SOC^7 - 2.9999 * 10^6 * SOC^6 + 3.8799 * 10^7 * SOC^5 \\ &-2.2329 * 10^7 * SOC^4 + 4.7136 * 10^6 * SOC^3 + 2.3046 * 10^5 * SOC^2 \\ &-1.1586 * 10^5 * SOC + 1.7701 * 10^4 \end{aligned} \tag{21}$$

$$\begin{aligned} C_{p2}(SOC) = &-1.9877 * 10^8 * SOC^8 + 9.8351 * 10^8 * SOC^7 - 2.0425 * 10^9 * SOC^6 \\ &+2.3101 * 10^9 * SOC^5 - 1.5431 * 10^9 * SOC^4 + 6.1685 * 10^8 * SOC^3 \\ &-1.4211 * 10^8 * SOC^2 + 1.6887 * 10^7 * SOC - 1.7701 * 10^5 \end{aligned} \tag{22}$$

4.2.2. Constant current exchanger experiment

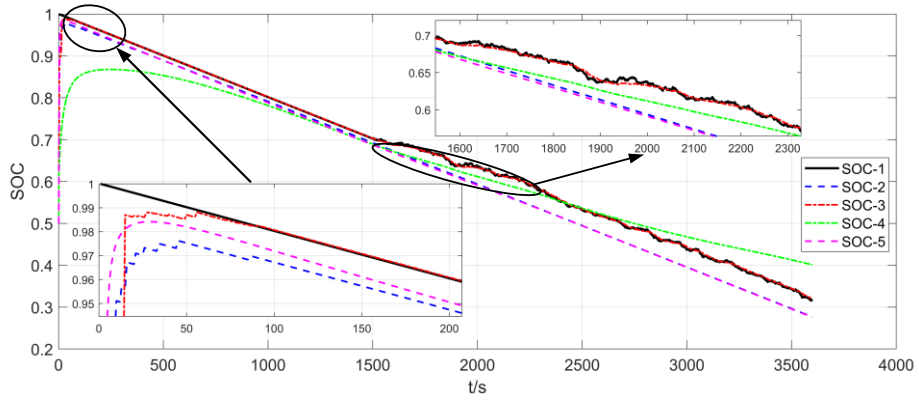
To verify the superiority of GPSO-PF, this paper utilizes CPU Intel(R) Celeron(R) and memory 1.10Ghz of Windows 10 system for simulation experiment verification. Based on the equivalence circuit model, the SOC value obtained by the Ah integral method is used as the reference value according to the state transfer equation (4) and the observation equation (5). In order to illustrate how well the algorithm tracks at arbitrary initial values and under system mutations, a random mutation in the current was added to the moments after 1500 s of the constant current discharge experiment. The initial state of PF and PSO-PF algorithm is set to [0.5 0 0 0]. The sampling interval is 0.1s. The number of particles is 50. The effective particle threshold is 2Neff/3. The number of iterations of the particle cluster optimization algorithm is 20, and when the maximum number of iterations or the optimal solution of the particle cluster is updated 10 times, it is determined that the particle set is distributed near the real state and exit optimization. Figure 7 (a) and Figure 7 (b) show the true reference state distribution and resampling particle set distribution of PF and PSO algorithms, respectively. SOC-1 is the standard SOC value. SOC-2 is the estimated SOC value by PF. SOC-3 is the estimated SOC value by PSO-PF.



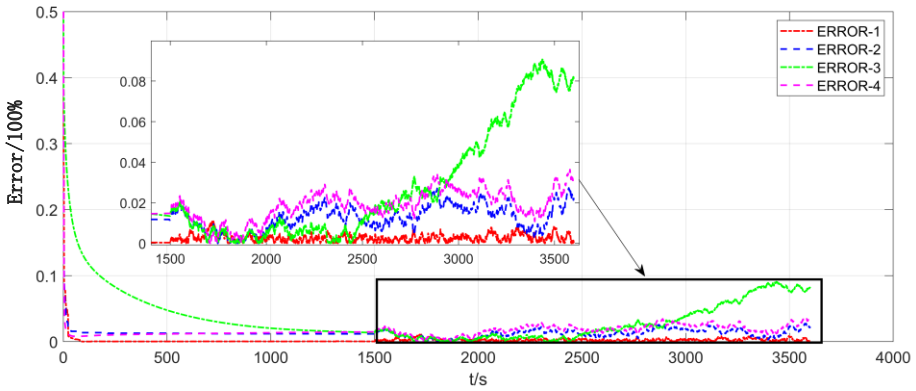
(a) Particle distribution by PF (b) Particle distribution by GPSO-PF

Figure 7. Particle distribution after resampling by two algorithms

As can be seen from Figure 7 (a), the algorithm degradation of PF is evident. The particle diversity of the PF algorithm is low and there is no dispersion of particles that can represent the true state value again through iteration, with almost only three particles involved in the estimation process of the system state, resulting in a gradual increase in the estimation error over time. However, as can be seen from Figure 7 (b) that the overall particle set of the GPSO-PF algorithm is uniformly distributed in the same interval. There are still particles that can participate in the expression as the state of the system changes abruptly with the change. The improved algorithm is able to drive the particle set to the high likelihood region. To illustrate the superiority of the improved algorithm, EKF, UKF and PF are used as comparisons in the constant current discharge experiments. Figure 8(a) shows the SOC estimates of the PSO algorithm and other algorithms, and Figure 8(b) shows the SOC errors of these algorithms. SOC-4 is the estimated SOC value by EKF. SOC-5 is the estimated SOC value by UKF.



(a) SOC estimation



(b) SOC estimation error

Figure 8. SOC estimation and its error via different algorithms

ERROR-1 is the estimation error of PSO-PF. ERROR-2 is the estimation error of PF. ERROR-3 is the estimation error of EKF. ERROR-4 is the estimation error of UKF. It can be seen from Figure 8(a) that the improved algorithm has a higher estimation accuracy and convergence speed, which is not affected by the number of particles, the initial value of the system and the state mutations. Compared to other algorithms, PF leads to over-frequent resampling and severe algorithm degradation due to the small number of particles and large initial value deviations, which makes it impossible to track the system state. Although EKF is able to gradually converge to the standard SOC values, it is the slowest to converge, and after adding the amount of state mutations, the error accumulates [45] and does not converge in later tracking. UKF has higher convergence speed than PF, yet it cannot be used in non-Gaussian distributed systems [46] and cannot track the real state of the SOC later. As can be seen in Figure 8(b), the error of the improved algorithm has been maintained at a smooth level with a maximum estimated deviation of less than 0.89%. Despite the fact that EKF can achieve good tracking results in the early stages, the error accumulates more and more in the later stages when the system state changes abruptly. PF has higher accuracy than UKF, but neither algorithm can accurately track the SOC state at the later stage.

4.2.3. Dynamic current experiment

To further verify the filtering performance of the improved PSO-PF algorithm and to simulate the SOC estimation under frequent charge/discharge conditions, a dynamic variable current test was performed on the lithium-ion battery. The experimental procedure was set up as follows: first, discharge the battery at 0.5C for 6 minutes and stand still for 3 minutes. And then, in each dynamic cycle test, charge the battery at 0.5C for 1 minute and stand still for 3 minutes. After that discharge the battery at 1C for 6 minutes and leave it alone for 3 minutes. Go through several cycles until the battery reaches the protection voltage. The current is shown in Figure 9.

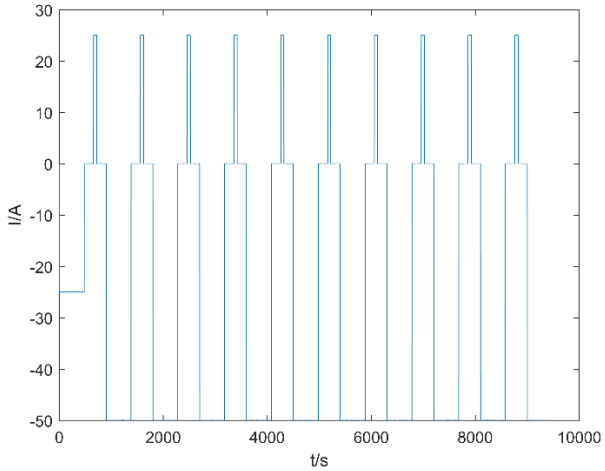
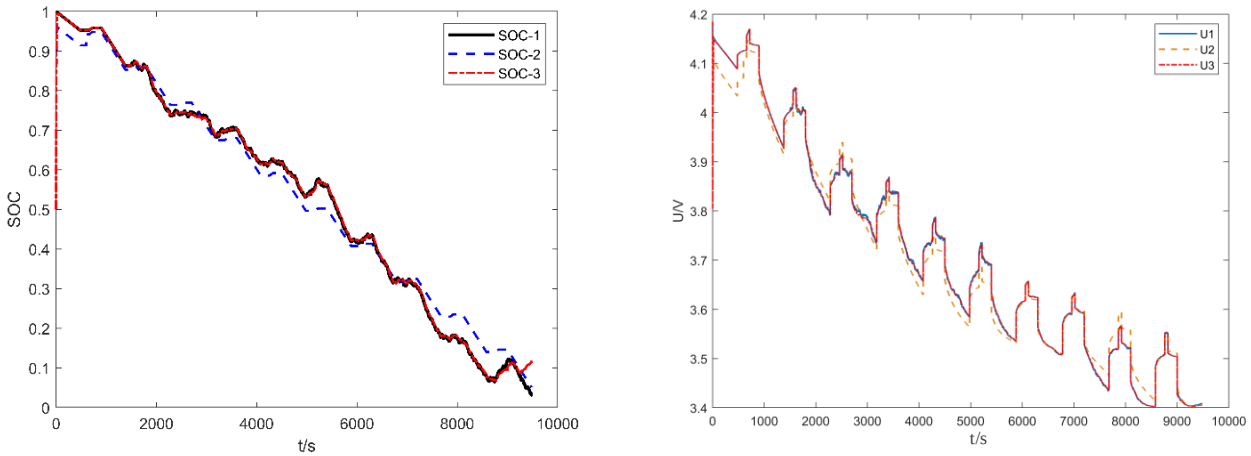


Figure 9. Dynamic current diagram

The initial settings are the same as for the constant current discharge experiment. Under the dynamic current test, the SOC estimation and voltage values obtained by the Ah integration method are used as reference. The SOC estimation is shown in Figure 10.



(a) SOC (b) Voltage

Figure 10. SOC estimation and voltage under dynamic current

It can be seen from Figure 10 (a) that the PF algorithm has caused a serious algorithm degradation problem when the system initial particle set deviation is 0.5, while the PSO algorithm has strong robustness and accuracy in the process of resampling near the true state value. In the Figure 10 (b), it can be seen that the model voltage estimated by GPSO-PF varies with the input current, and its operating voltage value is not affected by the measured noise at all. In order to quantitatively compare the filtering performance of PF and PSO-PF, the root mean square error of the experiment is defined as formula (23).

$$MSE=\frac{1}{\mathrm{T}}\sum_{k=1}^{T}(x_k^*-x_k) \tag{23}$$

T represents the total time step of an experiment; x_k represents the estimated SOC state at the kth moment; x_k^* represents the true SOC state value at the k moment. And the filtering efficiency η is defined by the formula (24), taking into account the effect of particle number and estimation time on the filtering performance.

$$\eta=1/(MSE*M) \tag{24}$$

The formula (24) shows that the higher the particle filtering efficiency, the higher the filtering accuracy (smaller MSE) from a smaller number of particles. The initial value of the setting algorithm is [0.5 0 0], and 10 independent sets of dynamic variable current experiments under the same working conditions are performed. The estimates of the average elapsed time, root mean square error, and filtering efficiency of the two algorithms for different particle counts are calculated, respectively, as shown in Table 2.

Table 2. Comparison of filtering performance under different particle Numbers

Algorithm	particles	MSE	Efficiency η/%	Averge time/s
PSO-PF	30	0.0586	56.797	345.9
	50	0.0564	35.436	407.5
	100	0.0574	17.403	880.1
PF	50	8.57	0.233	14.4
	500	3.28	0.0610	132.7
	1000	0.00587	17.020	285.2

As shown in Table 2, the filtering efficiency and error of PF is much worse than PSO-PF at M = 50. Although the RMS error of the PF algorithm gradually decreases when increasing the number of particles. However, at M=500, the increase in the number of particles instead reduces the filtering efficiency by nearly three times. In addition, at M=1000, the filtering efficiency of PF only reaches 17.403%, which brings nearly 20 times the time consumption.

To achieve more accurate estimates, too many particles in the initial setup can lead to excessive time consumption, while a smaller number of particles can lead to excessive accuracy errors. Compared to PSO-PF, the particle swarm optimization improves the effectiveness of each particle. The prior distribution of the particles dynamically adjusts their position and velocity from the current measured

value to achieve more accurate SOC estimation. For example, at M = 30, the filtering efficiency can reach 56.797% with a RMS error of 0.0586. However, as the number of particles increases, the root-mean-square error and the filtering performance may also decrease, partly due to the oscillation of the PSO near the global optimum at a later stage, resulting in a longer computation time. On the other hand, it is due to the fact that the PSO algorithm is insensitive to particle size [47], when the population size decreases, it has little effect on the filtering performance, which is mainly related to the particle position and motion velocity at each iteration.

5. CONCLUSION

With the rapid development of socio-economic, the applications of lithium-ion batteries are becoming more and more widespread. In order to ensure the safety of lithium-ion batteries, higher requirements are placed on the SOC estimation of lithium-ion batteries. In this paper, the state space expression of the lithium-ion battery is obtained on the basis of 2-order RC model. The GPSO-PF method effectively suppresses the degradation of the PF algorithm and improves the problem of large SOC estimation errors under fault conditions. Finally, it is experimentally shown that the proposed algorithm, even at small numbers of particles and large initial deviations, is still able to maintain high estimation accuracy, filtering efficiency and strong robustness.

NOMENCLATURE

The symbols used in this research can be described as shown in Table 3.

Table 3. List of symbols

Symbol	Description	Symbol	Description
SOC	State of Charge	EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter	ECM	Equivalent Circuit Model
Ah	Ampere hour	EV	Electric Vehicle
RC	Resistance and Capacitance	UKF	Unscented Kalman Filter
PF	Particle Filter	MC	Monte Carlo
OCV	Open Circuit Voltage	LOLN	Law of Large Number

GPSO	Gaussian Particle Swarm Optimization	PSO	Particle Swarm Optimization
HPPC	Hybrid Pulse Power Characterization	RBT	Recursive Bayesian Theorem

ACKNOWLEDGMENTS
The work was supported by National Natural Science Foundation of China (No. 61801407) and the Undergraduate Innovation Fund Project (CX19-097).

References

1. M.S.H. Lipu, M.A. Hannan, A. Hussain, M.M. Hoque, P.J. Ker, M.H.M. Saad, and A. Ayob, *J. Cleaner Prod.,* 205 (2018) 115.
2. T.R. Hawkins, B. Singh, G. Majeau-Bettez, and A.H. Strømman, *J. Ind. Ecol.*, 17 (2013) 158.
3. M.A. Hannan, M.S.H. Lipu, A. Hussain, and A. Mohamed, *Renewable Sustainable Energy Rev.,* 78 (2017) 834.
4. J. Duan, X. Tang, H. Dai, Y. Yang, W. Wu, X. Wei, and Y. Huang, *Electrochem. Energ. Rev.,* 3 (2020) 1.
5. L.S. Guo, Z.R. Wang, J.H. Wang, Q.K. Luo, and J.J. Liu, *J. Loss Prev. Process Ind.,* 49 (2017) 953.
6. Y. Fu, S. Lu, K. Li, C. Liu, X. Cheng, and H. Zhang, *J. Power Sources,* 273 (2015) 216.
7. Q. Liu, S. Liu, H. Liu, H. Qi, C. Ma, and L. Zhao, *Energies,* 12 (2019) 2041.
8. S. Kim, and I.H. Jung, *Probab. Eng. Mech.,* 50 (2017) 9.
9. Z. Chen, H. Sun, G. Dong, J. Wei, and J. Wu, *J. Power Sources,* 414 (2019) 158.
10. P.L.T. Duong, and N. Raghavan, *Microelectron. Reliab.,* 81 (2018) 232.
11. M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, *IEEE Trans. Signal Process.*, 50 (2002) 174.
12. L. Li, A.A.F. Saldivar, Y. Bai, and Y. Li, *Energies,* 12 (2019) 2784.
13. Y. Ma, Y. Chen, X. Zhou, and H. Chen, *IEEE Trans. Control Syst. Technol.,* 27 (2019) 1788.
14. Pan C, *Int. J. Electrochem. Sci.,* 14 (2019) 9537.
15. K. Zhang, J. Ma, X. Zhao, D. Zhang, and Y. He, *IEEE Access.*, 7 (2019) 166657.
16. M. Ye, H. Guo, and B. Cao, *Appl. Energy,* 190 (2017) 740.
17. Y. Guo, Z. Zhao, and L. Huang, *Energy Procedia.,* 105 (2017) 4153.
18. L. Fan, B. Hongkui, X. Jiajun, Y. Chenlong, and L. Xuhui, *J. Syst. Eng. Electron.,* 29 (2018) 854.
19. C. Zhang, X. Wang, W. Qin, and Y. Wang, *Trans. Jpn. Soc. Aeronaut. Space Sci.,* 62 (2019) 162.
20. M. Abbasi, and M.R. Khosravi, *J. Grid Comput.,* 18 (2020) 305.
21. S.A. Alam, and O. Gustafsson, *J. Signal Process Syst.,* 92 (2020) 555.
22. B. Duan, Q. Zhang, F. Geng, and C.H. Zhang, *Int. J. Energy Res.,* 44 (2020) 1724.
23. Y.K. Fang, C. Wang, W. Yao, X.J. Zhao, H.J. Zhao, and H.B. Zha, *IEEE Trans. Intell. Transp. Syst.,* 20 (2019) 4538.
24. G. Xie, X. Peng, X. Li, X.H. Hei, and S.L. Hu, *Can. J. Chem. Eng.,* 98 (2020) 1365.
25. H. Zhang, Q. Miao, X. Zhang, and Z. Liu, *Microelectron. Reliab.,* 81 (2018) 288.
26. C. Zheng, Z. Chen, and D. Huang, *Energy.* 191 (2020) 116504.
27. J. Hou, Y. Yang, H. He, and T. Gao, *Appl. Sci.*, 9 (2019) 1726.
28. B.H. Zhou, X.M. Liao, and K. Wang, *Soft Comput.,* 23 (2019) 13067.
29. G.S. Walia, and R. Kapoor, *Expert Syst. Appl.,* 41 (2014) 6315.

30. Y.K. Qiao, Q. Zhang, and J.S, *Appl. Mech. Mater.,* 44 (2010) 3459.
31. Z. Chen, Y. Bo, M. Tian, P. Wu, and X. Ling, *J. Aerosp. Eng.*, 31 (2018) 1943.
32. L. Min, Z. Chen, and X. Shao, Iterated Unscented Kalman Particle Filter algorithm based on a variable-step adaptive artificial fish swarm optimization, *Chinese Control Conference*, Chengdu, China, 2016, 1934-1768
33. H. He, R. Xiong, and J. Fa, *Energies*, 4 (2011) 582.
34. H. He, R. Xiong, H. Guo, and S. Li, *Energy Convers. Manage.,* 64 (2012) 113.
35. S. Santhanagopalan, Q. Guo, and P. Ramadass, *J. Power Sources,* 156 (2006) 620.
36. L.J. Zhang, H. Peng, Z.S. Ning, Z.Q. Mu, and C.Y. Sun, *Appl Sci.,* 7 (2017) 1002.
37. J.S. Liu, and R. Chen, *J. Am. Stat. Assoc.,* 93 (1998) 1032.
38. A. Doucet, S. Godsill, and C. Andrieu, *Stat. Comput.,* 10 (2000) 197.
39. J. Kennedy, and R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, Australia, 1995, 1942-1948
40. I.C. Trelea, *Inf. Process. Lett.*, 85 (2003) 317.
41. R.A. Krohling, and D.S.C. Leandro, *IEEE Trans. Syst. Man Cybern. Part B Cybern.,* 36 (2006) 1407.
42. S. Schindler, M. Bauer, M. Petzl, and M.A. Danzer, *J. Power Sources.*, 304 (2016) 170
43. Y. Wang, C. Zhang, and Z. Chen, *J. Power Sources.*, 305 (2016) 80.
44. J. Su, M. Lin, S. Wang, J. Li, J. Coffie-Ken, and F. Xie, *Meas. Control.*, 52 (2019) 193.
45. L. Guo, J. Li, and Z. Fu, *Energy Procedia,* 158 (2019) 2599.
46. Q. Yu, R. Xiong, and C. Lin, *Energy Procedia,* 105 (2017) 2791.
47. P.P. Adam, J.N. Jaroslaw, and E.P Agnieszka, *Swarm Evol. Comput.*, 58 (2020) 100718