

Research Article

Parameter Identification and State of Charge Estimation of NMC Cells Based on Improved Ant Lion Optimizer

Kai Zhang , Jian Ma , Xuan Zhao , Xiaodong Liu , and Yixi Zhang 

School of Automobile, Chang'an University, Xi'an, China

Correspondence should be addressed to Xuan Zhao; zhaoxuan@chd.edu.cn

Received 5 March 2019; Revised 13 June 2019; Accepted 17 June 2019; Published 16 July 2019

Academic Editor: Adrian Chmielewski

Copyright © 2019 Kai Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For lithium battery, which is widely utilized as energy storage system in electric vehicles (EVs), accurate estimating of the battery parameters and state of charge (SOC) has a significant effect on the prediction of energy power, the estimation of remaining mileage, and the extension of usage life. This paper develops an improved ant lion optimizer (IALO) which introduces the chaotic mapping theory into the initialization and random walk processes to improve the population homogeneity and ergodicity. After the elite (best) individual is obtained, the individual mutant operator is conducted on the elite individual to further exploit the area around elite and avoid local optimum. Then the battery model parameters are optimized by IALO algorithm. As for the SOC estimation, unscented Kalman filter (UKF) is a common algorithm for SOC estimation. However, a disadvantage of UKF is that the noise information is always unknown, and it is usually tuned manually by “trial-and-error” method which is irregular and time-consuming. In this paper, noise information is optimized by IALO algorithm. The singular value decomposition (SVD) which is utilized in the process of unscented transformation to solve the problem of the covariance matrix may lose positive definiteness. The experiment results verify that the developed IALO algorithm has superior performance of battery model parameters estimation. After the noise information is optimized by IALO, the UKF can estimate the SOC accurately and the maximum errors rate is less than 1%.

1. Introduction

With the increasingly serious problems of global energy crisis and environmental pollution, electric vehicles (EVs) have experienced explosive development and drawn increasing attention from institute and industry in the last decade [1–3]. The energy storage system is the most important part in EVs which is often comprised by different kinds of batteries, including lead-acid batteries, nickel-hydrogen batteries, and secondary lithium batteries. Lithium battery is the most popular energy source nowadays and occupies more than 90% of the market share due to the characteristic of high energy density, high efficiency, no memory effect, long cycle life, and environment-friendly performance [4–6]. Due to the flammability of liquid electrolyte and the lithium dendrite growth which may puncture battery separator and cause short-circuit, lithium battery may degrade rapidly even thermal runaway under abuse condition, like overcharge or over discharge. Therefore, lithium battery should be operated

stable and safe, which is achieved by advanced Battery Management System (BMS) [7]. One of the most important functions of BMS is state of charge (SOC) estimation, which can improve the control efficiency of energy and extend the usage life of the batteries [8]. However, SOC cannot be obtained directly by measuring and must be deduced from other measurable signals, like voltage and current. Also, several factors can influence the accuracy of SOC estimation and should be thoroughly investigated [9].

Many studies have focused on the research of SOC estimation strategies. The most widely used strategy is coulomb counting which is based on load current integration [10–13]. However, this strategy is open-loop without feedback and may easily cause cumulative calculation error due to inaccurate initial SOC value and stochastic disturbances. The other one popular technology is intelligent algorithm control strategies including neural network and machine learning. This method is not interested in the battery internal state and only cares about the relationship between the output

and input which can extract from experimental data set [14]. However, this method needs a larger amount of experimental data for algorithms training and always is time-consuming [15–18]. Another dominating method is state estimate algorithms based on battery model, like equivalent circuit models (ECMs) and electrochemical models [19–26]. This method is often utilized in BMS due to the easy implementation, high accuracy, and high efficiency. As for the battery models, ECMs are more suitable for online deployment than electrochemical models since ECMs utilize electrical parameters like current, voltage, and temperature, which are easier to obtain online than the parameters used in electrochemical models (e.g., electrolyte concentration and diffusion coefficients). Among the ECMs model-based methods, the two widely used SOC estimation techniques are extended Kalman filter (EKF) and unscented Kalman filter (UKF) [27–32]. However, the noise information (covariance matrixes) of the state space should be known to apply EKF and UKF algorithms, which is often obtained manually by “trial-and-error” method and this may be difficult and time-consuming. Some researches proposed adaptive technique to calculate the covariance matrixes based on the estimation error [28, 31]. However, this method may cause the covariance matrix becoming singular/negative and shut down the estimation process. Another important aspect of SOC estimation is the accuracy of battery model. The battery model parameters are always obtained by recursive least square (RLS) algorithm [7]. However, the results of RLS are always fluctuation and highly affected by system noise.

This paper introduces nature-inspired metaheuristic algorithm to find the optimal value of battery model parameters and the covariance matrixes. The key features of metaheuristic algorithms are randomness and the utilization of stochastic operators when searching in the solution spaces [33]. Some of the most popular metaheuristic algorithms contain particle swarm optimization (PSO) [34], simulated annealing algorithm (SA) [35], ant colony optimization (ACO) [36], genetic algorithm (GA) [37], firefly algorithm [38], and BAT Search Algorithm [39]. Ant lion optimizer (ALO), proposed by Mirjalili, is imitating the unique predation behavior of ant lions to find the optimal value [40, 41]. The performance of ALO in solving optimization problem has been vindicated and only a few parameters are needed. E.S. Ali uses the ALO to find the optimal positions and determine the sizing of Distributed Generation (DG) in various distribution systems [42]. However, ALO has a drawback of early prematurity and local optimum, especially in solving complicated problems [43]. To solving this problem, different methods have been combined with ALO to enhance the performance. Tian applied PSO after the solution space has been searched by ALO, which can combine the search ability of ALO and PSO [43]. Zhongqiang Wu applied chaotic sequence for generating the initial position of population (ants and antlions) to increase the uniformity and ergodicity [44].

Several improvements are introduced into ALO algorithm to enhance the optimization performance. Chaotic mapping theory is introduced in the initialization and random walk processes. Then after the elite (best) individual is

found by ALO in one iteration, individual mutant operator based on chaotic mapping is conducted on every elements of the elite individual, which can further exploit the region around elite. Based on the proposed improved ant lion optimization (IALO), the parameters of battery model are optimized by IALO. Also, the noise covariance matrixes in UKF are optimized by IALO. Another small improvement of this paper is that singular value decomposition (SVD) is utilized in UKF to solve the problem of the covariance matrix which may become singular and lose positive definiteness.

The remainder of the paper is arranged as follows: Section 2 introduces the battery model, the parameters estimation method, and the SOC estimation algorithm. Section 3 gives a brief review of the antlion algorithm. In Section 4, the improved ALO is proposed. Section 5 describes the experimental results and discussion. Section 6 provides the conclusion.

2. Battery Model and SOC Estimate Algorithm

2.1. Battery Model. Equivalent circuit model, electrochemical model, and black box model are three widely used models for modeling sophisticated battery behaviors. The three models have their own characteristic in complexity, accuracy, and computational cost [45]. Considering the hardware limits and real-time control of EVs, including enough accuracy and reasonable complexity, equivalent circuit model is widely used in EVs application. Many kinds of equivalent circuit models have been studied in the past. Twelve equivalent circuit models have been studied in [46] which include zero-state hysteresis, one-state hysteresis model, self-correcting (ESC) model, first-order RC model, second-order RC model, third-order RC model, and so on. Their experiments and comparison results demonstrated that, considering both model accuracy and model robustness, the first-order RC model, which is also called Thevenin model, is preferred for lithium nickel-manganese-cobalt oxide (LiNMC) cells. Therefore, the Thevenin battery model is adopted in this paper and demonstrated in Figure 1. The battery model consists of a voltage component U_{oc} which represents the open circuit voltage (OCV), a resistor component R_i , which denote the ohmic resistance, and a RC network component, which describes the concentration polarization and diffusion processes during battery operation, the resistance element R_p denotes the polarization resistance, and the capacitance C_p denotes the polarization capacitance. The electrical characteristic of the Thevenin model is calculated as

$$\begin{aligned} \dot{U}_p &= -\frac{1}{C_p R_p} U_p + \frac{1}{C_p} I_L \\ U_t &= U_{oc} - U_p - I_L R_i \end{aligned} \quad (1)$$

where I_L is the load current (assume greater than 0 for discharging) and U_t represents the terminal voltage. U_p is the polarization voltage over the polarization resistance, and the U_{oc} represents the OCV and can be modeled with function

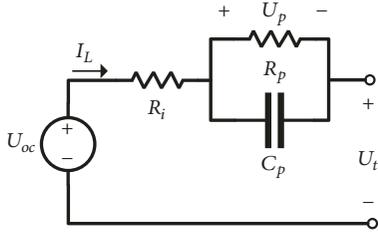


FIGURE 1: Sketch map of Thevenin model.

$f(z)$ which is a simple electrochemical model [7], and $f(z)$ can be calculated as

$$f(z) = K_0 + K_1 z + K_2 z^2 + K_3 z^3 + \frac{K_4}{z} + K_5 \ln(z) + K_6 \ln(1-z) \quad (2)$$

where K_i ($i = 0, 1, \dots, 6$) are multinomial coefficients and they need to be fitted based on experimental data and z denotes the state of charge (SOC) of battery.

2.2. Parameters Identification Method. For iteration calculation purpose, the discrete form of state space equations can be written as

$$\begin{aligned} U_{p,k+1} &= e^{-\Delta t/\tau} U_{p,k} + (1 - e^{-\Delta t/\tau}) R_p I_{L,k} \\ U_{t,k} &= U_{oc,k} - U_{p,k} - I_{L,k} R_i \end{aligned} \quad (3)$$

where Δt is fixed sample interval and assumed to be 1s in this study. τ is the time constant which equals $\tau = R_p C_p$. k is an integer variable which denotes the discrete index. I_k is the k th sampling time of load current. $U_{t,k}$ and $U_{oc,k}$ are the terminal voltage and open circuit voltage at k th sampling moment, respectively. $U_{p,k}$ is the polarization voltage at k th sampling moment.

The load current I_k and the terminal voltage $U_{t,k}$ are directly measured. The open circuit voltage $U_{oc,k}$ is calculated based on SOC which is integral of load current from a known initial SOC value. The model parameters, including R_i , R_p , and C_p , are identified by the improved ant lion optimization (IALO) algorithm which will be discussed in Sections 3 and 4. The optimization objective function is to minimize the quadratic sum of the voltage estimation error.

2.3. SOC Estimation Based on Singular Value Decomposition (SVD) UKF. After the parameters of battery model are identified by IALO, SOC is estimated by UKF (unscented Kalman filter). The state space equations of battery model are nonlinear. According the widely used EKF SOC estimation algorithm, the nonlinear state equations must be linearization, and the often-used linearization method is based on Taylor expansion which may lead into truncation error. The UKF algorithm does not need to linearize the model and apply nonlinear unscented transformation (UT) to generate sigma points from the original state. The sigma points are calculated based on known mean and covariance and then

substituted into the state function to obtain the new sigma points, then updating mean and covariance matrixes by the statistical characteristics of new sigma points.

The SOC denotes the proportion of the residual charge in battery to the rated capacity. The discrete form of SOC function is calculated by

$$z_k = z_{k-1} - \eta \frac{I_{L,k} \Delta t}{Q_n} \quad (4)$$

where z_k is the SOC at k th sampling moment, Q_n is the rated capacity, and η is the coulomb efficiency and related to current rate, battery chemistry, and temperature. For simplicity, the coulomb efficiency is set as $\eta = 1$.

To utilize the UKF algorithm for SOC estimation, the nonlinear battery dynamic state equation and measurement equation are described as

$$\begin{aligned} X_k &= \begin{bmatrix} U_{p,k} \\ z_k \end{bmatrix} = f(X_k, u_k) + \omega_k \\ Y_k &= U_{t,k} = h(X_k, u_k) + \nu_k \\ f(X_k, u_k) &= \begin{bmatrix} e^{-\Delta t/\tau} U_{p,k-1} + (1 - e^{-\Delta t/\tau}) R_p I_{L,k} \\ z_{k-1} - \eta I_{L,k} \Delta t / Q_n \end{bmatrix} \\ h(X_k, u_k) &= U_{oc,k} - U_{p,k} - R_i I_{L,k} \\ \omega_k &\sim N(0, Q_k), \\ \nu_k &\sim N(0, R_k) \end{aligned} \quad (5)$$

where $X_k = [U_{p,k} \ z_k]^T$ denotes the state vector and the system input vector u_k is the measured load current $I_{L,k}$. The output vector Y_k is the terminal voltage $U_{t,k}$, ω_k and ν_k represent process noise signal and measurement noise signal, and their covariance matrixes are Q_k and R_k , respectively.

The very important part of UKF nonlinear unscented transformation (UT) is calculated as follows:

$$\begin{aligned} \hat{X}_{k-1}^0 &= \hat{X}_{k-1}^+ \\ \hat{X}_{k-1}^i &= \hat{X}_{k-1}^+ + \sqrt{(n+\lambda)} \left(\sqrt{P_{k-1|k-1}} \right)_i, \quad i = 1, 2, \dots, n \\ \hat{X}_{k-1}^i &= \hat{X}_{k-1}^+ - \sqrt{(n+\lambda)} \left(\sqrt{P_{k-1|k-1}} \right)_i, \\ & \quad i = n+1, n+2, \dots, 2n \end{aligned} \quad (6)$$

$$\begin{aligned} \omega_i^m &= \omega_i^c = \frac{1}{2(n+\lambda)}, \quad i = 1, 2, \dots, 2n \\ \omega_0^m &= \frac{\lambda}{n+\lambda}, \\ \omega_0^c &= \omega_0^m + 1 + \beta - \alpha^2 \end{aligned} \quad (7)$$

where \hat{X}_{k-1}^i ($i = 0, 1, 2, \dots, 2n$) is the sigma points, ω_i^m and ω_i^c are the weight coefficients of mean and covariance, respectively, n is dimensional of the state vector X_k , $\lambda =$

$\alpha^2(n + \kappa) - n$ is a scale adjustment parameter which can adjust the distribution of sigma points and enhance accuracy, α and β are set to 1 and 2 in this paper, and κ is normally set to 0 or 3- n . $\sqrt{P_{k|k}}$ is the Cholesky decomposition of covariance matrix $P_{k|k}$.

There is a problem existing in Cholesky decomposition, namely, if the covariance matrix $P_{k|k}$ is singular, the Cholesky decomposition cannot be carried out, and the iterative calculation would be stopped. Moreover, the Cholesky decomposition is very sensitive to the calculation error, and the finite word length or rounding error in the computational process will often make the covariance matrix $P_{k|k}$ lose its symmetry or positive definiteness, thus resulting in the problem of numerical instability of the filtering and reducing the accuracy of the filtering algorithm. In order to solve this problem, we introduce singular value decomposition (SVD) of matrix into the generating of sigma points, which can enhance the numerical stability of iterative updating state covariance matrix and decompose covariance matrix even morbidity occurs in matrix.

According to the singular value decomposition theory, a matrix $A_{m \times n}$ can be decomposed by orthogonal matrix $U_{m \times m}$ and $V_{n \times n}$ as

$$A = U\Lambda V^T$$

$$\Lambda = \begin{bmatrix} \Sigma & O \\ O & O \end{bmatrix}_{m \times n} \quad \Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \quad (8)$$

where σ_i is the eigenvalue of matrix $A^T A$ and called singular value, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$. When matrix A is symmetric positive definite matrix, the SVD of matrix A is rewritten as

$$A = UDU^T = U\sqrt{D}\sqrt{D}U^T \quad (9)$$

The SVD of covariance matrix $P_{k|k}$ is calculated as

$$P_{k-1|k-1} = U_{k-1}D_{k-1}^2U_{k-1}^T \quad (10)$$

where $P_{k-1|k-1}$ is covariance matrix of last iteration k -1th and then the sigma points at k th iteration are calculated by

$$\begin{aligned} \hat{X}_{k-1}^0 &= \hat{X}_{k-1}^+ \\ \hat{X}_{k-1}^i &= \hat{X}_{k-1}^+ + \sqrt{(n + \lambda)} (U_{k-1})_i (D_{k-1})_i, \\ & \quad i = 1, 2, \dots, n \quad (11) \\ \hat{X}_{k-1}^i &= \hat{X}_{k-1}^+ - \sqrt{(n + \lambda)} (U_{k-1})_i (D_{k-1})_i, \\ & \quad i = n + 1, n + 2, \dots, 2n \end{aligned}$$

The detailed SVDUKF algorithm process is shown in Table 1. In the SVDUKF algorithm process, the process noise covariance Q_k , and measurement noise covariance matrix

R_k are often considered as constant value. The Q_k and R_k are always decided by trial-and-error procedure which is arbitrary and time-consuming. In this study, the improved ant lion optimization (IALO) algorithm is utilized to search the optimal value of Q_k and R_k which will be discussed in Sections 3 and 4. The optimization objective function is to minimize the quadratic sum of the voltage estimation error and SOC estimation error with different weights.

3. Introduction of Ant Lion Optimization Algorithm

Ant lion optimizer (ALO), a recently proposed heuristics nature-inspired algorithm, imitates the foraging behavior of ant lion's larvae [40]. The ant lions belong to the genus of Myrmeleontidae, carnivorous, and feed on other insects. The larvae of ant lions live under dry ground and create funnel-shaped traps in sandy soil to trap prey. After the trap is built, the ant lion larvae wait the prey (mainly ants) by hiding underneath the bottom of the trap. In natural environment, ants move randomly and casually; once ants move into pits, the ant lion will try to catch ants. However, ants usually may try to creep out the trap. In this situation, ant lions will cast sands towards the edge of the trap to make the trap side steeper and slide the ant down to the bottom of the trap and grab it. After an ant is seized, ant lion will pull the prey under the sand and eat, then throw out the leftovers, and rebuilt the pit for the next prey. The ant lion denotes the optimized solution of the optimization problem and the ant stand for the feasible solution. By catching the ant which has higher fitness, the ant lion can update its position and fitness by the information of ant.

In the search space, there are many ant lions and ants, whose positions are located randomly. To imitate the interaction between ant lions and ants, ants will move over the search space with random walk, which may go through or around the traps. The superior individual/ant lion will build a larger and sharper pit and catch more ants. The random walk steps of ants are influenced by ant lions and elite (best) ant lion. Ants will update their position based on ant lions and elite ant lion to ensure the diversity and optimization capability of the algorithm.

During the optimization, the algorithm abides the following conditions: (1) ants are wandering over the solution space with random walks, and the random walks are influenced by the position of traps of ant lions; (2) pits built by ant lions are proportional to the fitness of themselves, and the better fitness (the bigger pit) has the high probability to catch ants; (3) the step of random walk is accordingly reduced when ant is sliding to the ant lion which is hiding under the bottom of the pit; (4) if the fitness of an ant is better than that of an ant lion, this signifies that the ant is seized and eaten by the ant lion, and the ant lion changes itself with the position of the ant and growing fitter; (5) after each hunting, an ant lion changes its position near the position where ant is caught and set up a new pit to improve its possibility of catching prey.

The random walk, which is chosen for modeling ants' stochastically moving behavior, is obtained as

$$\begin{aligned}
 X(t) = & [0, \text{cumsum}(2 \cdot r(t_1) - 1), \\
 & \text{cumsum}(2 \cdot r(t_2) - 1), \dots, \\
 & \text{cumsum}(2 \cdot r(t_n) - 1)]
 \end{aligned} \quad (12)$$

where *cumsum* is the function of cumulative sum, *n* is the biggest number of iterations, *t* denotes the current iteration, and *r(t)* represents a stochastic function calculated by

$$r(t) = \begin{cases} 1 & \text{if } rand > 0.5 \\ 0 & \text{if } rand \leq 0.5 \end{cases} \quad (13)$$

where *rand* is a random number between (0,1), which subject to uniform distribution.

The number of both ants and ant lions is *N*, and the dimension of position is *D*. The positions of ants are recorded in matrix M_{Ant} ; $A_{i,j}$ represents the value of *j*th dimension of position value of the *i*th ant; A fitness function f_j is utilized to evaluate each ant and matrix M_{OA} is utilized to record the fitness value of all ants; $M_{Antlion}$ stores the position of every antlion and $AL_{i,j}$ represents the value of *j*th dimension of position of the *i*th antlion; fL_j is utilized to evaluate each antlion and matrix M_{OAL} is utilized to store the fitness value of all ant lions.

$$\begin{aligned}
 M_{Ant} &= \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & \cdots & A_{1,D} \\ A_{2,1} & A_{2,2} & \cdots & \cdots & A_{2,D} \\ \vdots & \vdots & \ddots & & \vdots \\ A_{N,1} & A_{N,2} & \cdots & \cdots & A_{N,D} \end{bmatrix} \\
 M_{OA} &= \begin{bmatrix} f([A_{1,1}, A_{1,2}, \dots, A_{1,D}]) \\ f([A_{2,1}, A_{2,2}, \dots, A_{2,D}]) \\ \vdots \\ f([A_{N,1}, A_{N,2}, \dots, A_{N,D}]) \end{bmatrix} \\
 M_{Antlion} &= \begin{bmatrix} AL_{1,1} & AL_{1,2} & \cdots & \cdots & AL_{1,D} \\ AL_{2,1} & AL_{2,2} & \cdots & \cdots & AL_{2,D} \\ \vdots & \vdots & \ddots & & \vdots \\ AL_{N,1} & AL_{N,2} & \cdots & \cdots & AL_{N,D} \end{bmatrix} \\
 M_{OAL} &= \begin{bmatrix} fL([A_{1,1}, A_{1,2}, \dots, A_{1,D}]) \\ fL([A_{2,1}, A_{2,2}, \dots, A_{2,D}]) \\ \vdots \\ fL([A_{N,1}, A_{N,2}, \dots, A_{N,D}]) \end{bmatrix}
 \end{aligned} \quad (14)$$

Random walk of ants is based on (12). Ant changes its position by random walk at each iteration during optimization. Considering the search space has its own boundary, in

order to keep the position of ant after random walk remain in the search space, the following equation is utilized to normalize and calculate the new position:

$$R_i^t = \frac{(X_i^t - a_i)(d_i^t - c_i^t)}{b_i - a_i} + c_i^t \quad (15)$$

where X_i^t is random walk step of *i*th ant; R_i^t is the normalized random walk displacement; a_i and b_i are the lowest and highest of random walk displacement of *i*th ant; c_i^t and d_i^t indicate the lowest and highest of random walk displacement of *i*th ant at current iteration.

As mentioned before, ant lions' position can influence the random walks of ants. This influence can be mathematically modeled by following equations:

$$\begin{aligned}
 c_i^t &= Antlion_j^t + c^t \\
 d_i^t &= Antlion_j^t + d^t
 \end{aligned} \quad (16)$$

where $Antlion_j^t$ denotes the position of selected *j*th ant lion at *t*th iteration and c^t and d^t indicate the lowest and highest of random walk displacement of all ants in the *t*th iteration. The selected antlion is determined by roulette wheel method. Ant lions throw sands out the trap after they perceive that an ant fall in the pit and slide ant towards the pit center. To model this mechanism, the region of ants' random walks is shrunk adaptively as the increasing of iteration and calculated by following equations:

$$\begin{aligned}
 d^t &= \frac{d^t}{I} \\
 c^t &= \frac{c^t}{I}
 \end{aligned} \quad (17)$$

where *I* is a ration, and shown as $I = 10^{w}t/T$, *t* denotes the current iteration, *T* denotes the biggest number of iteration, and *w* is an index varying with the current iteration ($w=2$ if $t/T > 0.1 > 0.1$, $w=3$ if $t/T > 0.5$, $w=4$ if $t/T > 0.75$, $w=5$ if $t/T > 0.9$, and $w=6$ if $t/T > 0.95$).

During optimization, the elite is set as the best ant lion and updated every iteration if better ant lion shows up. And the random walk of ants is also influenced by the elite ant lion. Therefore, it is assumed that the random walks of every ant are over the selected ant lion and the elite antlion and calculated by

$$Ant_i^t = \frac{R_S^t + R_E^t}{2} \quad (18)$$

where *t* denotes iteration, R_S^t denotes the random walk over the selected ant lion, R_E^t denotes the random walk over the elite ant lion, and Ant_i^t denotes updated position of *i*th ant. After random walk, the fitness is calculated. Comparing the fitness of ant lion and ant, assign better one to ant lion, which can also be regarded as the ant which is hunted. The whole procedure of ALO algorithm is demonstrated in Figure 2.

TABLE 1: SOC estimation procedure based on SVD-UKE.

Step I: Initialization: At $k=0$, assign \hat{X}_0^+ , P_0^+ , Q_0 , R_0

Step II: Time update: For $k = 1, 2, \dots$,
 Calculate sigma points: \hat{X}_{k-1}^i ($i = 0, 1, 2, \dots, 2n$), $P_{k-1|k-1} = U_{k-1} D_{k-1}^2 U_{k-1}^T$
 $\hat{X}_{k-1}^0 = \hat{X}_{k-1}^+$; $\hat{X}_{k-1}^i = \hat{X}_{k-1}^+ + \sqrt{(n+\lambda)}(U_{k-1})_i(D_{k-1})_i$, $i = 1, 2, \dots, n$;
 $\hat{X}_{k-1}^i = \hat{X}_{k-1}^+ - \sqrt{(n+\lambda)}(U_{k-1})_i(D_{k-1})_i$, $i = n+1, n+2, \dots, 2n$,
 $\omega_0^m = \frac{\lambda}{n+\lambda}$, $\omega_0^c = \omega_0^m + 1 + \beta - \alpha^2$, $\omega_i^m = \omega_i^c = \frac{1}{2(n+\lambda)}$, $i = 1, 2, \dots, 2n$
 Calculate new sigma points: \hat{X}_k^i ($i = 0, 1, 2, \dots, 2n$) $\hat{X}_k^i = f(\hat{X}_{k-1}^i, u_k)$
 Update the state mean: \hat{X}_k^- $\hat{X}_k^- = \sum_{i=0}^{2n} \omega_i^m \hat{X}_k^i$
 Update the state error covariance matrix: P_k^- $P_k^- = \sum_{i=0}^{2n} \omega_i^c (\hat{X}_k^i - \hat{X}_k^-)(\hat{X}_k^i - \hat{X}_k^-)^T + Q_k$

Step III: Measurement update:
 Obtain the output sigma points: \hat{Y}_k^i ($i = 0, 1, 2, \dots, 2n$) $\hat{Y}_k^i = h(\hat{X}_k^i, u_k)$
 Calculate the measurement mean: \hat{Y}_k^- $\hat{Y}_k^- = \sum_{i=0}^{2n} \omega_i^m \hat{Y}_k^i$
 Calculate the measurement error covariance matrix: P_k^h
 $P_k^h = \sum_{i=0}^{2n} \omega_i^c (\hat{Y}_k^i - \hat{Y}_k^-)(\hat{Y}_k^i - \hat{Y}_k^-)^T + R_k$
 Calculate the state and measurement error covariance matrix: P_k^{fh}
 $P_k^{fh} = \sum_{i=0}^{2n} \omega_i^c (\hat{Y}_k^i - \hat{Y}_k^-)(\hat{X}_k^i - \hat{X}_k^-)^T$
 Calculate Kalman gain: $K_k = P_k^{fh} (P_k^h)^{-1}$
 Measurement update of state estimate: \hat{X}_k^+ $\hat{X}_k^+ = \hat{X}_k^- + K_k (Y_k - \hat{Y}_k^-)$
 Measurement update of error covariance: P_k^+ $P_k^+ = P_k^- + K_k P_k^h K_k^T$

Step IV: SOC estimation result:
 $z(k) = X_k(2)$ and turn to Step 2 for next iteration estimation

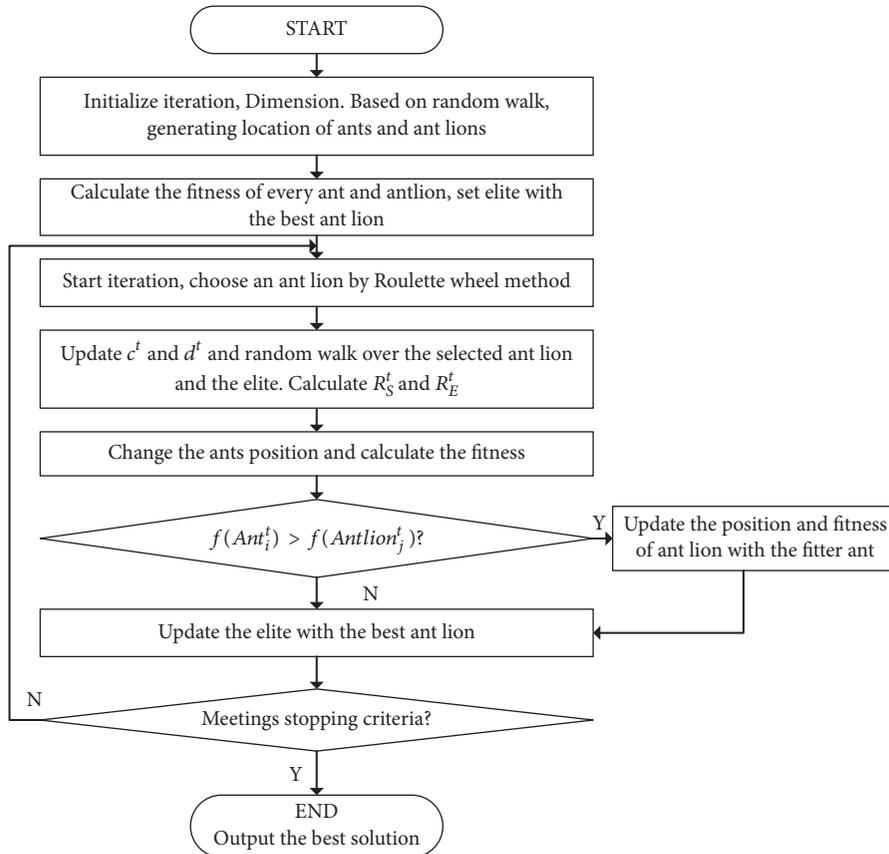


FIGURE 2: The procedure of ALO algorithm.

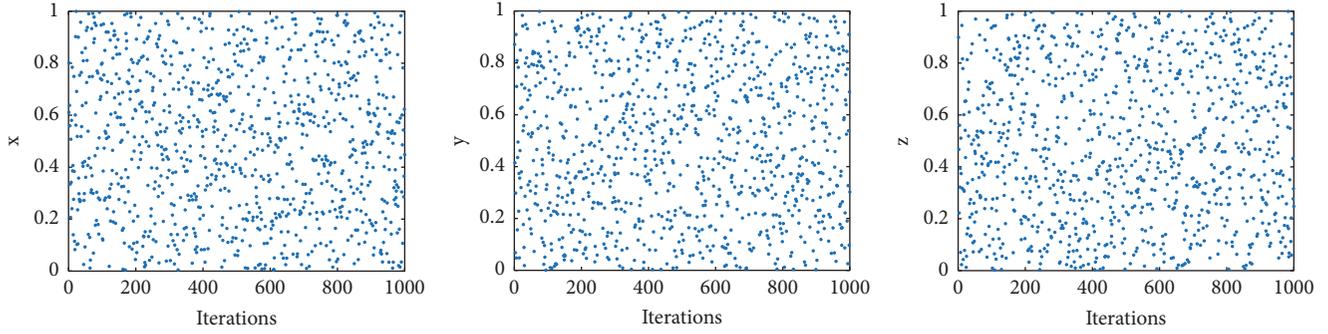


FIGURE 3: Chaotic property of the intertwining logistic map.

4. Improvements on Ant Lion Optimizer

In the ALO algorithms optimization procedure, the updating of ants' positions is based on the random walk over the chosen ant lion and the elite ant lion. Save the greatest agent (ant) as elite in each iteration. These can make ALO have low calculating cost, high efficiency, and high convergence rate. However, the ALO has the problem of premature convergence and local optimum when it is used for solving complex optimization problems. Several improvements are inserted into ALO to deal with these problems and further enhance the global optimization ability and accuracy.

4.1. Chaotic Map for Position Initialization and Random Walk. Random function has been used in many optimization algorithms. Chaos function has the same character of long-scale indeterminism and irregular property with random function. Chaos can help orders arise from disorder, which is like many biologicistic systems inspired optimization algorithms like GA, PSO, and ALO. Considering these common features between chaos function and optimization algorithms, it is natural to think about combining these theories to enhance the capability of the optimizer.

In this paper, to improve the population homogeneity and ergodicity and skip local optima, chaos function is utilized to generate random walk sequences. The random walk, which models the ants' stochastic movement, is calculated by iterating the steps of chaotic map from a random initial value.

Logistic map, a well-known chaotic map, has been used in many papers. Logistic map is based on polynomial model, which is a classical evidence of demonstrating how a very simple nonlinear dynamic system can generate complicated behavior [47, 48]. Logistic map is defined as follows:

$$x_{n+1} = \lambda \times x_n (1 - x_n) \quad (19)$$

where x_n is the n th chaotic value and n denotes the iteration number. λ is the control parameters of chaotic map. When $3.59 < \lambda < 4$, the system is in chaotic status. However, the logistic map demonstrates some shortages such as blank windows, stable windows, narrow key space, and uneven distribution of sequences. To obtain better chaotic behavior and larger key space and overcome this weakness, a 3D

intertwining logistic map is adopted in this paper [49, 50] and can be defined by

$$\begin{aligned} x_{n+1} &= [\lambda \times k_1 \times y_n \times (1 - x_n) + z_n] \pmod{1} \\ y_{n+1} &= \left[\lambda \times k_2 \times y_n + z_n \times \frac{1}{(1 + x_{n+1}^2)} \right] \pmod{1} \\ z_{n+1} &= [\lambda \times (x_{n+1} + y_{n+1} + k_3) \times \sin(z_n)] \pmod{1} \end{aligned} \quad (20)$$

where λ and k_i are control parameters and $0 < \lambda \leq 3.999$, $|k_1| > 33.5$, $|k_2| > 37.5$, and $|k_3| > 35.7$. The intertwining logistic map has better chaotic behavior and more even distribution, without the shortages of blank windows, stable windows. To more clearly demonstrate the uniformity and ergodicity of intertwining logistic map, the chaos sequence is generated by parameter set $x_0 = 0.75$, $y_0 = 0.26$, $z_0 = 0.53$, $\lambda = 3.75$, $k_1 = 36.1$, $k_2 = -42.6$, and $k_3 = 46.7$, as shown in Figure 3, and the first 100 iterations are removed to avoid the transient effect. From this figure, the chaotic function is efficient and random and chaotic search is easier to get rid of local minimum comparing with other stochastic optimization algorithms [51].

The chaotic value x_n belongs to the interval $(0, 1)$; it needs scale to the solution space by the following equation:

$$X_n = x_n \times (ub - lb) + lb \quad (21)$$

where X_n is the updated position after random walk; lb and ub are the lowest and biggest value of search space, respectively.

4.2. Individual Mutation Based Chaotic Mapping. After the elite individual is obtained, to further explore of the solution space and exploit of the small region around elite, chaotic mapping is used to mutate every dimension of the elite ant lion with a new value. This approach can enhance the stochastic property and increase the variety of the individual. The process is summarized below.

Initialization. The obtained elite individual is assumed as $e_0 = (e_0(1), \dots, e_0(j), \dots, e_0(dim))$; dim denotes the dimension and is set as 3 in this paper. Set the maximum number of chaotic mutations as N_c .

Start Iteration(For $k = 1$). Convert every element of elite individual to a normalized value $x_0(j)$ in the range (0, 1).

$$x_0(j) = \frac{e_0(j) - lb(j)}{ub(j) - lb(j)}, \quad j = 1, 2, 3 \quad (22)$$

where $ub(j)$ and $lb(j)$ are maximum and minimum of the j th element, respectively.

Outer Loop ($k = 2: Nc$). Generate new chaotic value through 3D intertwining logistic map theory as follows:

$$\begin{aligned} x_k(1) &= [\lambda \times k_1 \times x_{k-1}(2) \times (1 - x_{k-1}(1)) + x_{k-1}(3)] \pmod{1} \\ x_k(2) &= \left[\lambda \times k_2 \times x_{k-1}(2) + x_{k-1}(3) \times \frac{1}{(1 + x_k^2(1))} \right] \pmod{1} \\ x_k(3) &= [\lambda \times (x_k(1) + x_k(2) + k_3) \times \sin(x_{k-1}(3))] \pmod{1} \end{aligned} \quad (23)$$

Inner Loop ($j = 1: dim$). Invert the chaotic value into the actual position as follows:

$$e_k(j) = lb(j) + x_k(j) \times (ub(j) - lb(j)) \quad (24)$$

Replacing $e_0(j)$ with $e_k(j)$, the new individual is $e_{new} = (e_0(1), \dots, e_k(j), \dots, e_0(dim))$.

Calculate the fitness and replace e_0 with e_{new} when e_{new} is better than e_0 .

End Inner Loop. Form the new individual $e_k = (e_k(1), \dots, e_k(j), \dots, e_k(dim))$. Then comparing the fitness of e_0 and e_k , assign the bigger one to e_0 .

End Outer Loop. By means of individual mutant with chaotic mapping theory, there is better probability for the elite individual to skip over the local optimum and find a greater solution.

According to the above analysis of improvements on the ant lion optimization algorithm, the improved ant lion optimizer (IALO) procedure is demonstrated in Figure 4. In the following section, IALO algorithm will be used to optimize the value of battery model parameters. The state noise covariance matrix Q and measurement noise covariance matrix R values are also optimized by IALO algorithm.

5. Experiment and Results Discussion

In this section, to validate the performance of the developed IALO-SVDUKF SOC estimation method, experiments on LiNiMnCoO₂/Graphite lithium-ion cells were conducted under DST, FUDS, US06, and BJDST, respectively. Section 5.1 presents a brief introduction of the experiments. The results are discussed in Section 5.2.

5.1. Experiment Detail. The test bench includes an Arbin BT2000 battery tester, a thermal chamber with temperature controllable, a PC with Arbin Software to control the test system, and monitor and storage test data. 18,650 LiNiMnCoO₂/Graphite (NMC) lithium-ion cells are tested in the experiment and the basic information is shown in Table 2. The experiment ambient temperature is set to 25°C by using the thermal chamber. Sampling interval is set to 1s during the test. The test results are available at <https://web.calce.umd.edu/batteries/data.htm>.

The pulse-current tests are conducted to obtain the OCV-SOC function. The cell is first fully charged to 100% SOC by C/2 constant current charge phase and 4.2V constant voltage charge phase with a C/20 cut-off current. Second, the cell is discharged an amount of charge (10% SOC) with C/2, followed by 2 hours relaxation period to achieve equilibrium state inside the cell. The pulse-current, relaxation period is repeated every 10% SOC until reaching the lower cut-off voltage. Then, the cell is charged by following the similar steps until reaching the upper cut-off voltage: a constant voltage charge with a C/20 cut-off current is followed to make sure the cell is fully charged. Finally, the data from charge phase and discharge phase are averaged and nonlinear least square (LS) is applied to identify parameters of OCV-SOC function which is calculated by (2) and the results are given in Table 3 and Figure 5. As can be seen from the figure, the OCV function can follow the experiment data accurately.

Several widely used cycle conditions are conducted in the experiment, including DST (dynamic stress test) [52], FUDS (Federal Urban Driving Schedule) [53], BJDST (Beijing Dynamic Stress Test) [54], and US06 (US-6 Highway Driving Schedule) [55]. DST is the simplest condition and is utilized for identifying the battery parameters and the noise information. To validate the battery model and SOC estimation algorithm, other more complicated cycle conditions are utilized for SOC estimation.

5.2. Results and Discussion

(1) Certification of the Performance of IALO. Six widely used functions are selected to certify the optimization ability of IALO algorithm and compared with Genetic Algorithm (GA), particle swarm algorithm (PSO), and ALO algorithms. The test functions are shown in Table 4. Griewank function has many local extremum and multiple peaks. Ackley function has many local extremums around global extremum points in a narrow area. The global optimum of Rosenbrock function is located in a smooth, long, and narrow parabolic valley. Rastrigin function has many widespread local extremums. Branin function has 3 global minimum points and surrounded by many local extremums. Six-Hump Camelback function has six local extremums, and two of them are global optima. The parameters of GA algorithm are as follows: the population size is 100, the crossover probability is 0.7, and the mutation probability is 0.3. The parameters of PSO algorithm are as follows: the learning coefficients c_1 and c_2 are 1.5, and the population size is 200. The parameters of

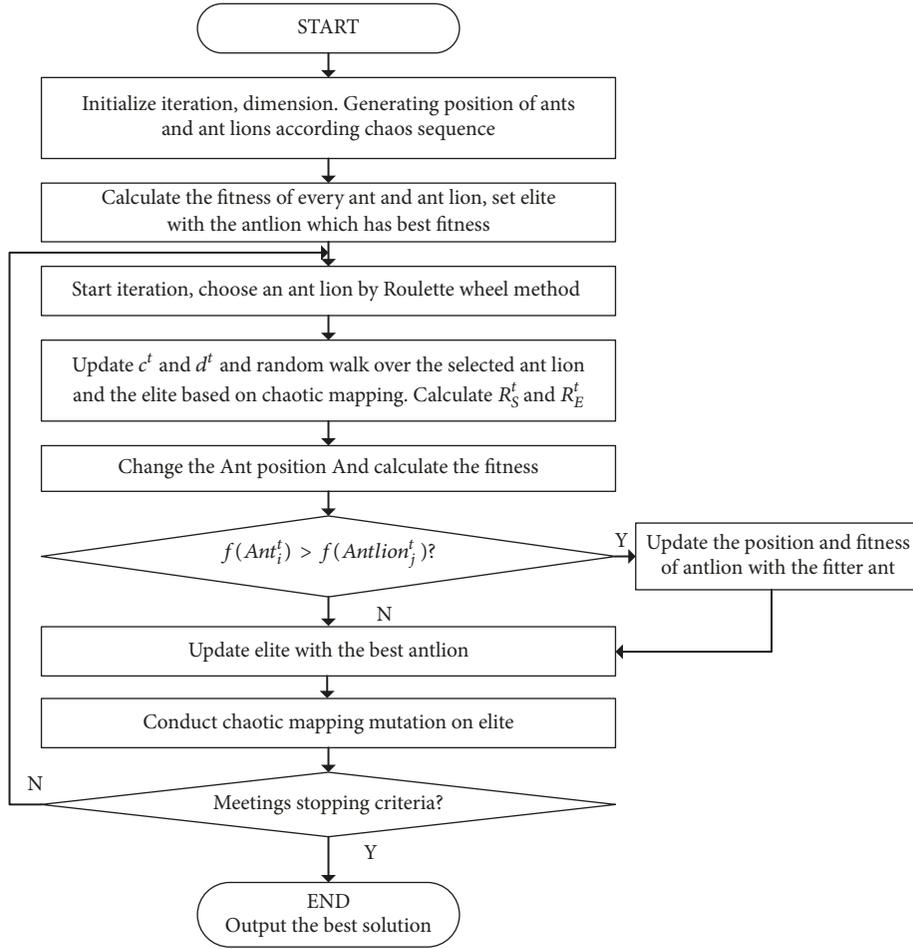


FIGURE 4: The process of IALO algorithm.

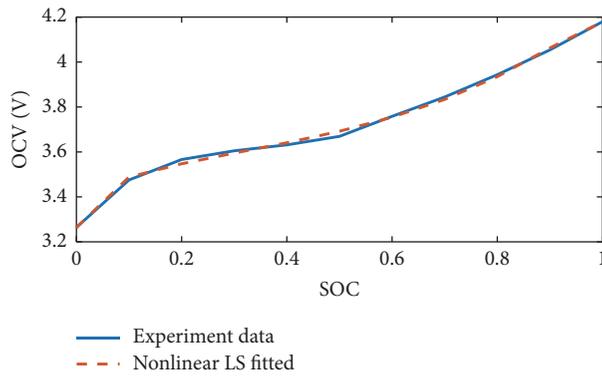


FIGURE 5: OCV-SOC curve of NMC cell.

TABLE 2: Basic information of test cells.

Nominal voltage (V)	Nominal capacity (Ah)	Upper cut-off voltage (V)	Lower cut-off voltage (V)
3.6	2.0	4.2	2.5

TABLE 3: The fitted parameters of the analytic OCV function for NMC cells.

K_0	K_1	K_2	K_3	K_4	K_5	K_6
3.4938	0.5755	-0.7051	0.8897	-2.55E-11	0.0232	0.0082

TABLE 4: The detail of test functions.

Name	Variable range	Function	Optimal value
Griewank	[-5, 5]	$f(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0
Ackley	[-5, 5]	$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	0
Rosenbrock	[-5, 5]	$f(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1}) + (x_i - 1)^2]$	0
Rastrigin	[-5, 5]	$f(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	0
Branin	[-5, 15]	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6\right)^2 + 10 \left(1 - \frac{1}{8\pi}\right) \cos(x_1) + 10$	0.397887
Six-Hump Camelback	[-5, 5]	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{x_1^6}{3} + x_1x_2 - 4x_2^2 + 4x_2^4$	-1.0316285

* D is set to 2 in this paper.

TABLE 5: The performance of algorithms.

Algorithm Function	Optimum value				Convergence iteration				Convergence time /s			
	GA	PSO	ALO	IALO	GA	PSO	ALO	IALO	GA	PSO	ALO	IALO
Griewank	0.0074	0.0075	3.879e-6	3.585e-12	18	62	100	88	0.0663	0.3604	6.415	6.059
Ackley	0.0010	0.0116	1.263e-6	3.459e-6	100	177	400	350	0.1003	0.4996	3.972	3.664
Rosenbrock	0.0506	0.0623	1.191e-13	2.872e-13	86	98	200	162	0.1273	0.2402	2.582	2.578
Rastrigin	2.946e-5	1.876e-5	1.421e-12	5.272e-12	90	45	200	154	0.0691	0.1386	2.413	1.842
Branin	0.3981	0.3980	0.39791	0.39789	12	15	144	95	0.0135	0.0450	6.816	6.282
Six-Hump Camelback	-1.0238	-1.0306	-1.0316	-1.0316	4	61	175	106	0.0069	0.1571	2.9613	2.677

TABLE 6: Identified battery parameters.

Algorithm	$R_i(\Omega)$	$R_p(\Omega)$	$C_p(F)$	RMSE
IALO	0.0766	0.0399	2692.06	0.0079
ALO	0.0763	0.0400	2400.55	0.0080
RLS	0.0714	0.0392	1130.1	0.0100

ALO algorithm are as follows: the number of search agents (ants) is 40, which is also the number of ant lions. The parameters of IALO algorithm are as follows: the number of ants or ant lions is the same as that of ALO, and the parameters of intertwining logistic map are $\lambda = 3.75$, $k_1 = 36.1$, $k_2 = -42.6$, and $k_3 = 46.7$.

The performance of optimal value search of GA, PSO, ALO, and IALO algorithm is demonstrated in Table 5. In the table, optimum value is the best optimal value that the algorithm can find. Convergence iteration represents the iteration when the optimum value keeps stable and there is no longer change. Convergence time is the time cost for the algorithm to find the optimal value. As shown in Table 5, four algorithms show different performance on different functions. GA algorithm and PSO algorithm have the similar performance and can only find the global value of Rastrigin function. When handling other functions, GA and PSO algorithms are stuck in local extremums with early iterations and cannot jump out local extremums. Due to the structural property, the convergence time of PSO is a little bit of higher than that of GA algorithms. ALO algorithm finds the global optimal value of all the test functions but Branin

function with only a little deviation. IALO algorithm explores the solution spaces thoroughly and finds the global optimum values of all of the test function. Comparing with ALO algorithm, IALO algorithm can find the global value by less iteration and less computation time and show superior search ability and fast convergence property. In comparison, IALO algorithm has the best optimization performance among all the algorithms.

(2) *Parameter Identification.* The model parameters are optimized by the proposed IALO algorithm. The initial population is $X = [R_i, R_p, C_p]$. The fitness of the individual is defined as follows:

$$f_{para} = \sum_{h=0}^N (U_{t,h} - U_{est,h})^2 \quad (25)$$

where N is the time length of the experiment data, $U_{t,h}$ denotes the measured terminal voltage at h th time, and $U_{est,h}$ is the estimated terminal voltage at h th time calculated by the corresponding individual (battery model parameters). The maximum iteration of IALO is set to 500, and iteration stops

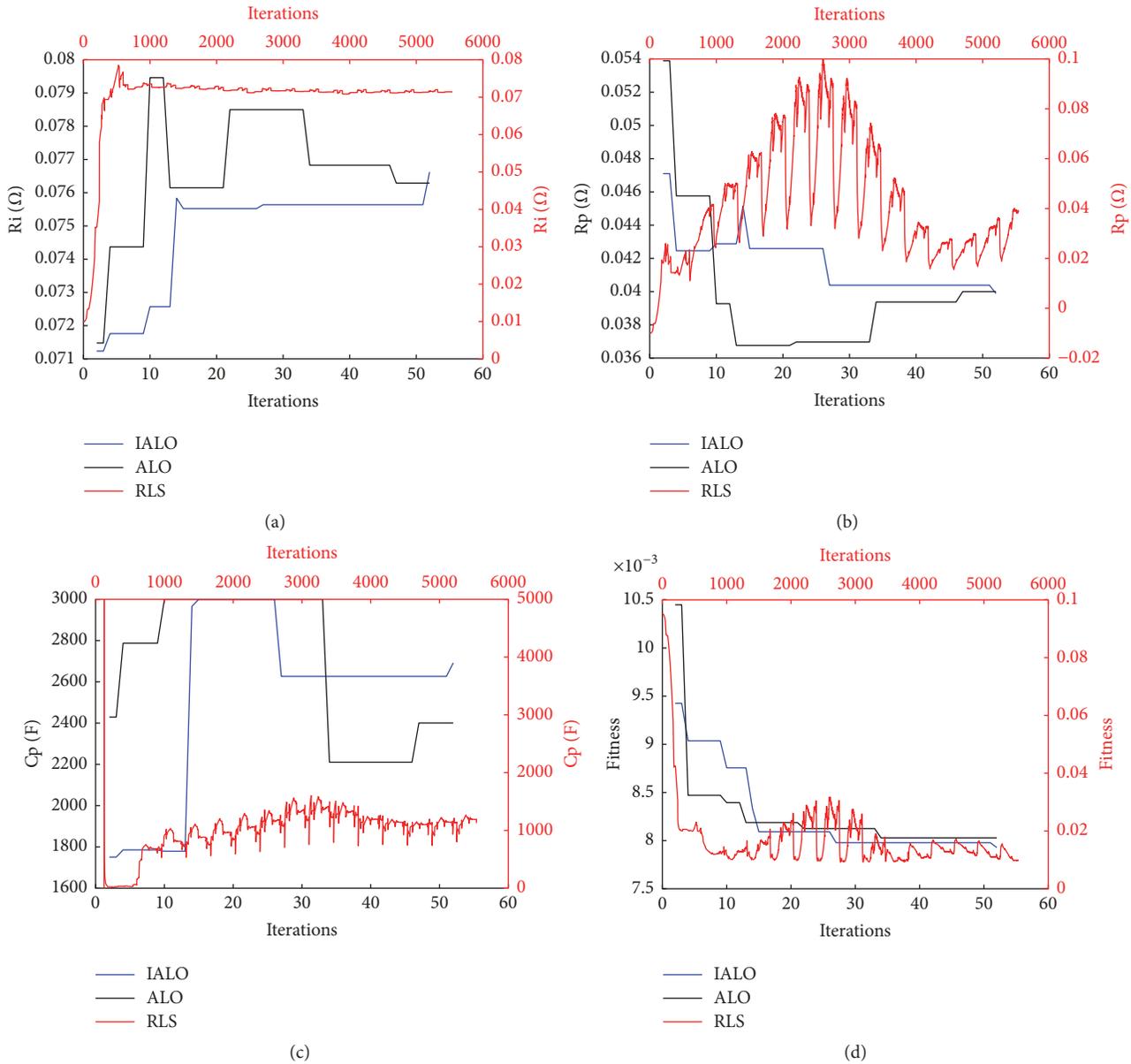


FIGURE 6: Parameter identification of battery model.

when the fitness of elite individual does not change in 10 continuous iterations.

The results of parameter identification are shown in Figure 6. As can be seen from the figure, the fitness convergence curves show that the ALO algorithm has early prematurity phenomenon and may fall in local optimal. On the contrary, the proposed improved ALO algorithm can further explore the solution space and find better solution. Comparing the conventions used battery parameters identification methods, recursive least square (RLS) algorithm, ALO and IALO can find the optimal solution in 50 iterations, while RLS needs more than 5000 iterations to reach the parameter values. Also, there are fluctuations in the results of RLS, which is adverse to the optimizing process. The results of ALO and IALO are more stable and beneficial for optimization. The final

identified parameters are shown in Table 6. The estimated terminal voltage is shown in Figure 7. The root mean square error (RMSE) of voltage estimation error by IALO algorithm is the smallest and equal to 0.0079 which is much smaller than the normal working range of voltage (2.5V-4.2V). These verify that the parameters obtained by IALO optimization algorithm can fit the measurement data quite well.

(3) *SOC Estimation.* After the battery model parameters are obtained, the proposed SVDUKF is utilized for SOC estimation. However, the process noise covariance matrix Q and the measurement noise covariance matrix R are always unknown and usually tuned by “trial-and-error” strategy manually by choosing random or experiential values, which may be very difficult, time-consuming, and unreliable. To

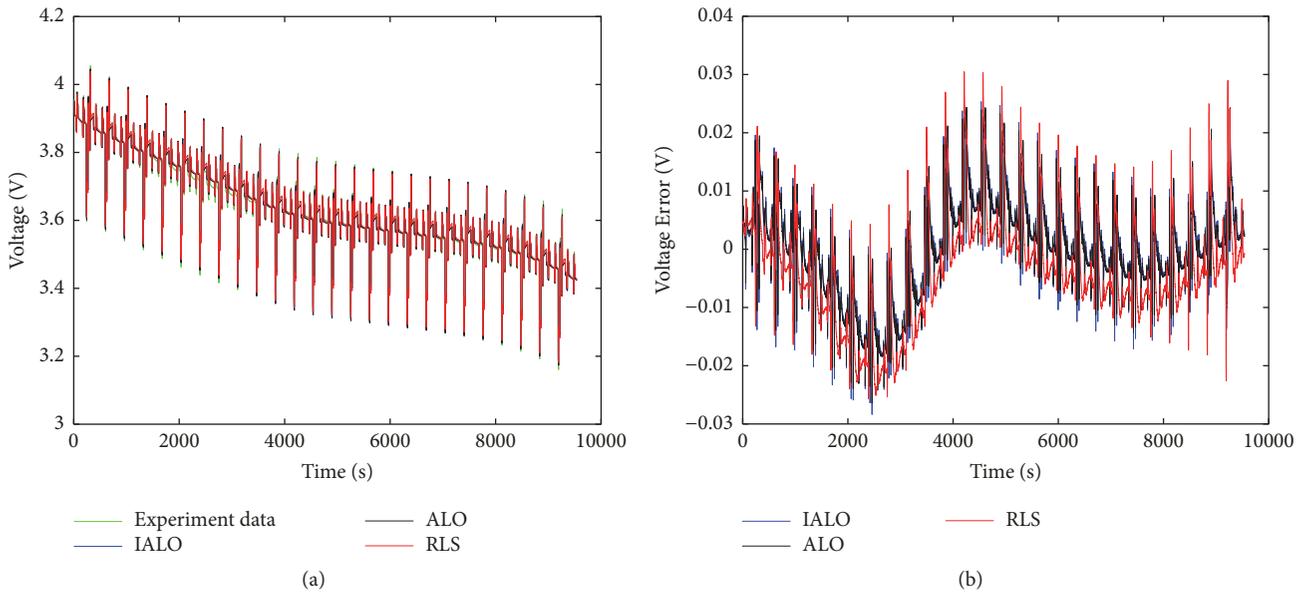


FIGURE 7: Terminal voltage estimation: (a) voltage results estimated with identified parameters; (b) voltage estimate error.

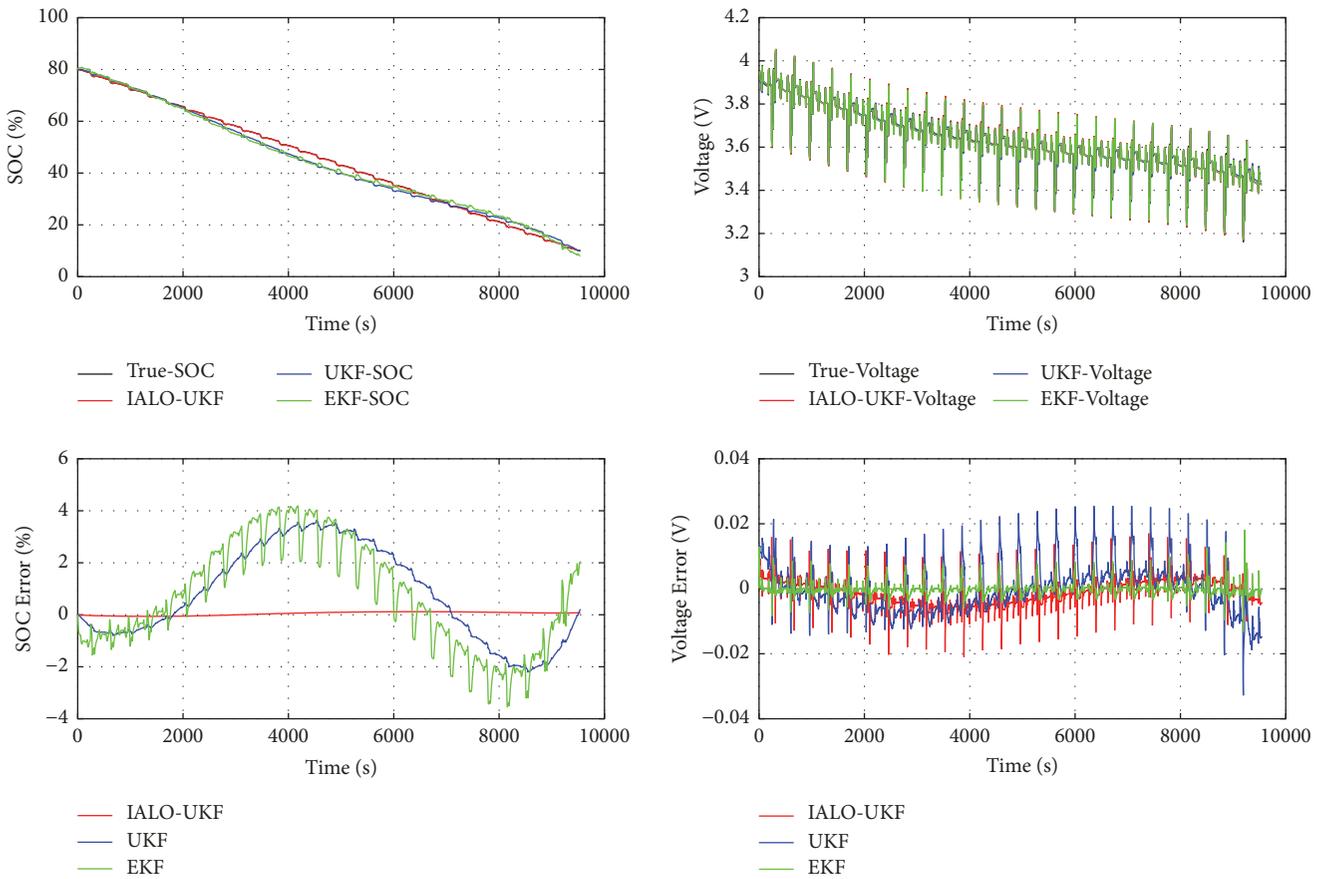


FIGURE 8: SOC and voltage estimation results via DST.

TABLE 7: The SOC estimation error different algorithms via DST.

Method	Q and R values $[q_1, q_2, r_1]$	RMSE of SOC error	RMSE of voltage error
IALO-SVDUKF	$[1e-10, 0.0011, 0.1338]$	7.88e-4	0.0040
UKF	$[1e-6, 1e-6, 0.1]$	0.0075	0.0195
EKF	$[1e-6, 1e-6, 0.1]$	0.0213	0.0021

TABLE 8: The RMSE of SOC estimation error via three dynamic conditions.

	RMSE	IALO-SVDUKF	UKF	EKF
SOC	FUDS	6.29e-4	0.0201	0.0224
	US06	0.0058	0.0211	0.0229
	BJDST	0.0058	0.0214	0.0231
Voltage	FUDS	0.0046	0.0073	0.0020
	US06	0.0038	0.0072	0.0025
	BJDST	0.0035	0.0070	0.0015

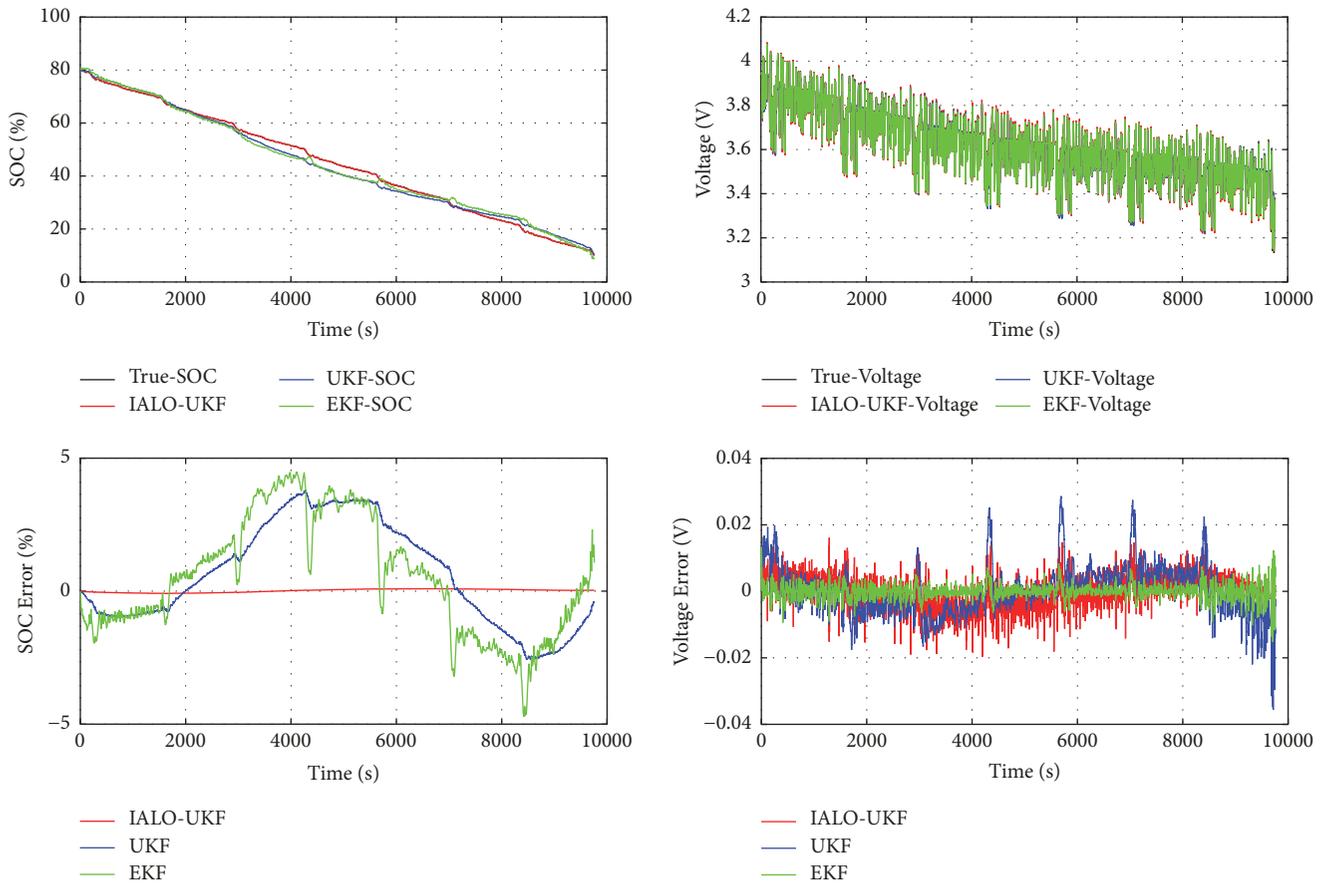


FIGURE 9: SOC and voltage estimation results via FUDS.

solving this problem, the proposed IALO algorithm is applied to search the optimal value for Q and R.

In UKF observer, Q and R matrixes are defined as diagonal and positive. In this paper, Q is 2×2 matrix and R is 1×1 matrix. Then the initial population is set as

$$\begin{aligned} Q &= \text{diag} [(q_1, q_2)], \\ R &= \text{diag} [(r_1)] \end{aligned} \quad (1)$$

$$X = [q_1, q_2, r_1] \quad (26)$$

The fitness of the individual is defined as follows:

$$f_{soc} = \sum_{h=0}^N [w_1 (U_{t,h} - U_{est,h})^2 + w_2 (z_{ref,h} - z_{est,h})^2] \quad (27)$$

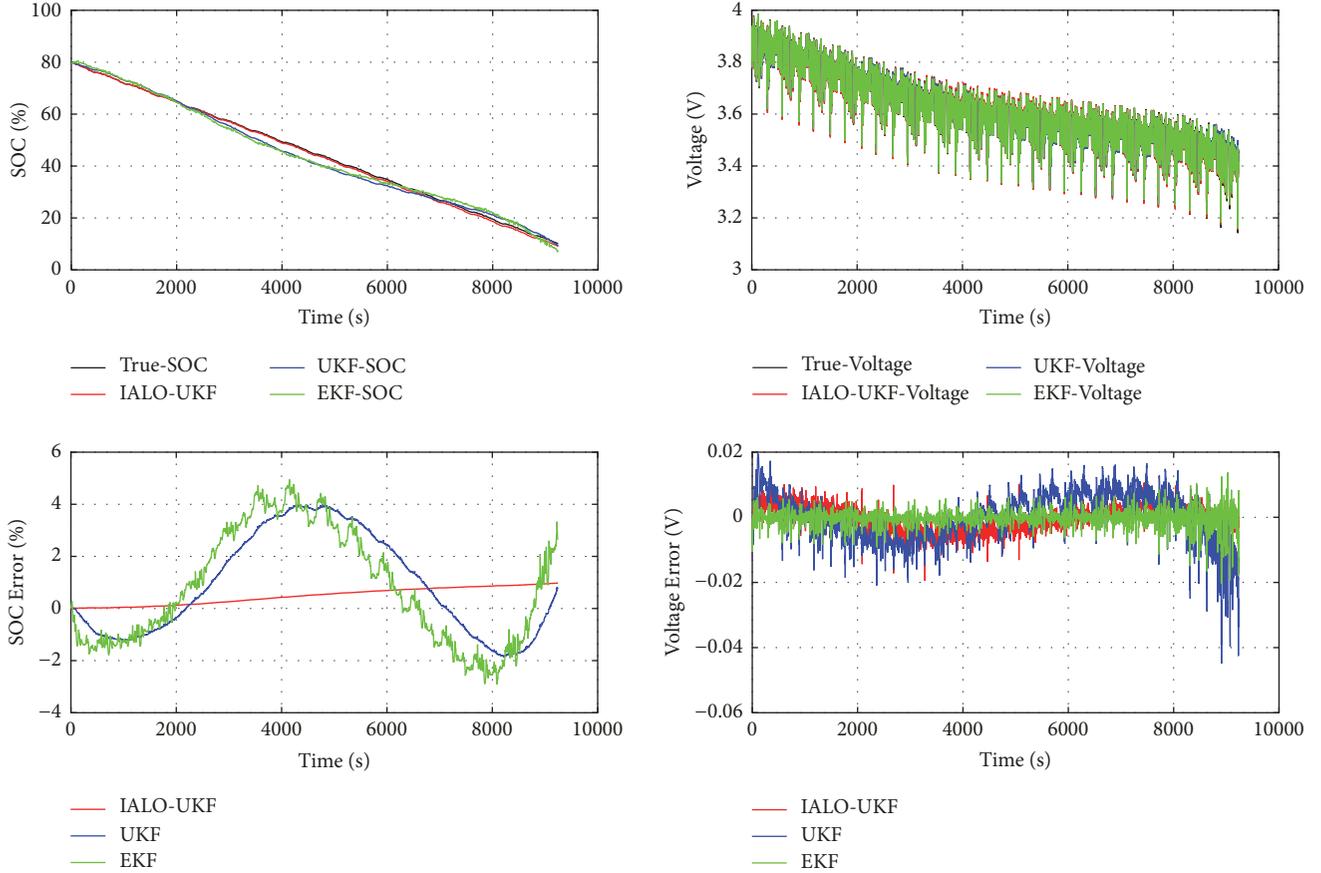


FIGURE 10: SOC and voltage estimation results via US06.

where $[w_1, w_2]$ is the weight of voltage and SOC estimation error, respectively, and $w_1 = 0.7$ and $w_2 = 0.3$ in this paper. N is the time length of the experiment data, $U_{t,h}$ denotes the measured terminal voltage at h th time, $U_{est,h}$ denotes the estimated terminal voltage at h th time, and $z_{ref,h}$ denotes the true SOC which is obtained by calculating the integral of current over time via the high accuracy current sensors of the battery test system and the precision control initial SOC. $z_{est,h}$ denotes the estimated SOC based on the corresponding individual (diagonal element of Q and R matrixes). The maximum iteration of IALO is set to 500, and iteration stops when the fitness of elite individual does not change in 10 continuous iterations. Then the first two elements of the optimal results are assigned to the covariance matrixes Q and the third element of the optimal results is assigned to the covariance matrixes R . Based on the optimal Q and R and the former obtained battery model, the SVDUKF is conducted for the SOC estimation via DST. For comparison, the UKF and EKF with arbitrarily tuning parameters are also applied for SOC estimation. The estimation results are demonstrated in Figure 8 and Table 7. From the results, the proposed IALO-SVDUKF algorithm can obtain precise SOC estimation and the RMSE is nearly 10 times smaller than that of UKF and 30 times smaller than that of EKF, and the maximum error rate is smaller than 1%, which can certify that the IALO algorithm can explore and exploit the solution space thoroughly and

find the better solution quickly and effectively and is labor-saving. Also, the terminal voltage estimation of three methods is comparable with each other and the maximum of the error is 2 orders of magnitude smaller than the normal range of battery voltage (2.5V-4.2V) and the maximum error rate is less than 0.5%.

To further validate the performance of IALO algorithm, the battery model parameters and the noise covariance matrixes are adopted to estimate the battery SOC under different dynamic conditions. The estimation results via FUDDS, US06, and BJDST are shown in Figures 9–11, similarly, comparing with UKF and EKF. The RMSEs of SOC and voltage estimation are shown in Table 8.

It is clearly shown that, under FUDDS cycle condition, with the parameters and noise covariance matrixes identified by IALO algorithm, SVDUKF can obtain more than 30 times smaller SOC estimation RMSE comparing with UKF and EKF which are using arbitrary noise covariance matrixes. Under other more complicated and fluctuated cycle condition, like US06 and BJDST, IALO-SVDUKF algorithm can still achieve nearly 4 times smaller SOC estimation RMSE. As for the terminal voltage estimation, the developed algorithm can acquire small RMSE which is less than that of UKF algorithm. EKF algorithm obtains the smallest voltage estimation RMSE but the largest SOC estimation RMSE. In the EV application, BMS is more concerned about SOC

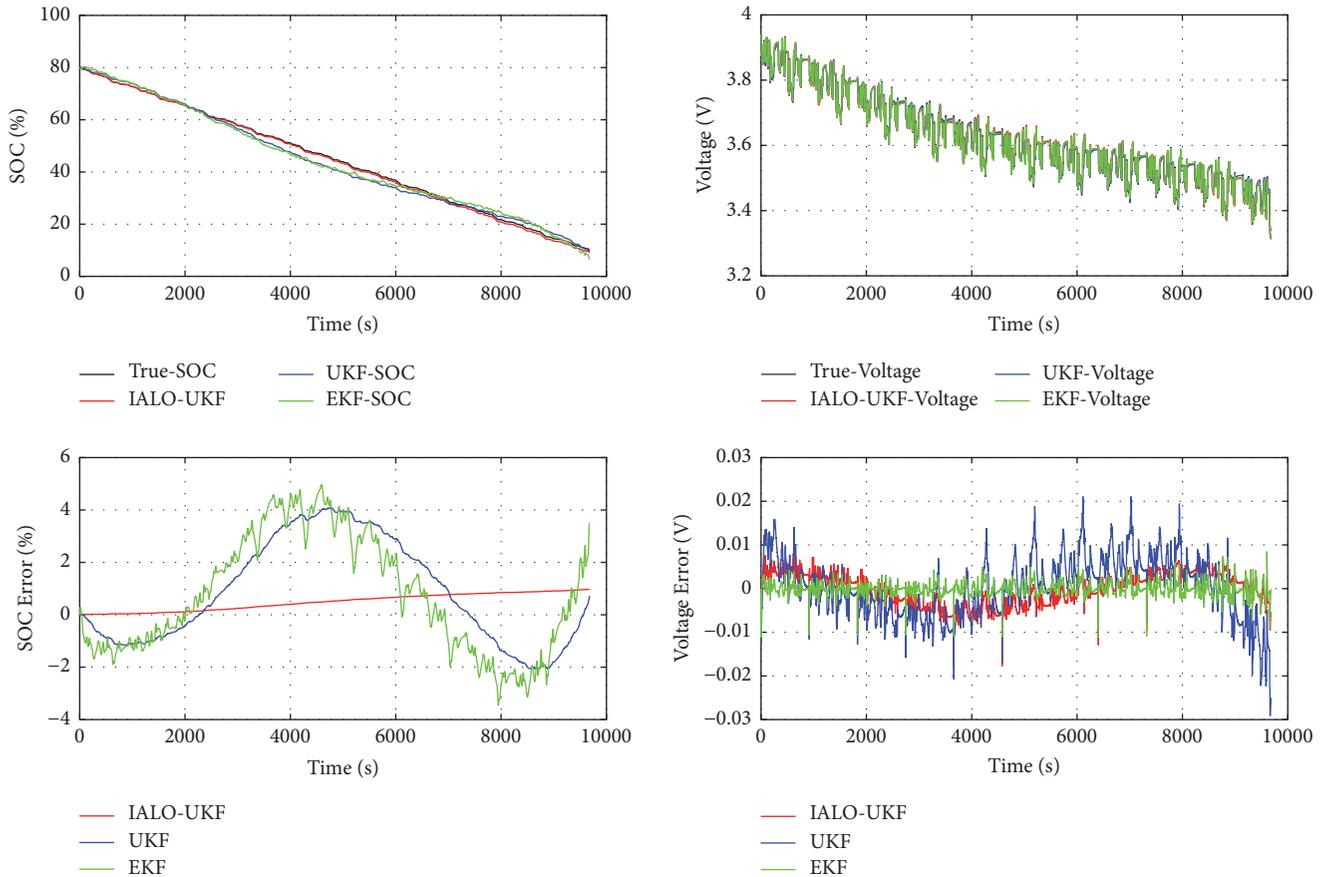


FIGURE 11: SOC and voltage estimation results via BJDST.

estimation. Therefore, the proposed algorithm can obtain the optimal value of the covariance matrix Q and R and then estimate SOC with high-degree precision, which is more preferable for EV and BMS application.

6. Conclusion

As a critical part of EV BMS algorithms, accurate SOC estimation and battery model can guarantee a precise control of battery inner state, lengthen the battery usage life, and reduce mileage anxiety. The widely used SOC estimation methods including EKF and UKF algorithms have the drawbacks of that the prior knowledge of noise covariance matrixes, which are often selected manually by “trial-and-error” method and are arbitrary and time-consuming.

This paper developed an improved ant lion optimization (IALO) algorithm to identify the battery model parameters and noise covariance matrixes. The IALO algorithm is based on a recently developed heuristics search method, namely, ant lion optimization (ALO) algorithm. To overcome the drawbacks of ALO, including early convergence and local optimum trapping, the chaotic mapping is utilized for generating the initial position of population and the random walk sequences to ensure the population homogeneity and ergodicity. Another improvement is the individual mutant which transfers each element of the elite ant individual according

to chaotic sequences generated by intertwining logistic map, to jump out local extremum and thoroughly explore the solution space.

To validate the performance of developed IALO algorithm for battery model parameters identification and SOC estimation, systematic experiments are conducted on NMC cells over different dynamic conditions. The results verify that the developed algorithm can obtain optimal battery model parameters in only 50 iterations. The accuracy of SOC estimation is less than 1% maximum error rate. The terminal voltage estimation is also desirable with less than 0.5% maximum error rate. Therefore, the proposed IALO algorithm has superior performance and can achieve fast and reliable results. Future work will be concentrated on online application of the IALO for SOC estimation.

Nomenclature

- LiNMC: Lithium nickel-manganese-cobalt oxide
- EVs: Electric vehicles
- SOC: State of charge
- ALO: Ant lion optimizer
- IALO: Improved ant lion optimizer
- UKF: Unscented Kalman filter
- EKF: Extended Kalman filtering
- SVD: Singular value decomposition

BMS: Battery Management System
 ECMs: Equivalent circuit models
 RLS: Recursive least square
 PSO: Particle swarm optimization
 SA: Simulated annealing
 GA: Genetic algorithm
 DG: Distributed Generation
 UT: Unscented transformation
 DST: Dynamic stress test
 FUDS: Federal Urban Driving Schedule
 BJDST: Beijing Dynamic Stress Test
 US06: US-6 Highway Driving Schedule
 U_{oc} : Open circuit voltage
 R_i : Ohmic resistance
 R_p : Polarization resistance
 C_p : Polarization capacitance
 I_L : Load current
 U_t : Terminal voltage
 U_p : Polarization voltage over the polarization resistance
 K_i : Multinomial coefficients
 Δt : Sample interval
 τ : Time constant
 z_k : The SOC at k th sampling moment
 Q_n : Rated capacity
 η : The coulomb efficiency
 ω_k : Process noise signal
 ν_k : Measurement noise signal
 Q_k : Process noise covariance matrix
 R_k : Measurement noise covariance matrix
 \hat{X}_{k-1}^i : Sigma points
 R_S^t : Random walk over the chosen ant lion
 R_E^t : Random walk over the elite ant lion
 Ant_i^t : Updated position of i th ant
 n : n th chaotic value
 x_n : n th chaotic value
 ω_i^m : Weight coefficients of mean
 ω_i^c : Weight coefficients of covariance
 $P_{k|k}$: Covariance matrix
 λ : Scale adjustment parameter
 α : Scale adjustment parameter
 β : Scale adjustment parameter
 \hat{Y}_k^i : Output sigma points
 K_k : Kalman gain
 $X(t)$: Random walk sequences
 n : Biggest number of iterations
 $r(t)$: A stochastic function
 N : The number of ants and ant lions
 D : The dimension of position
 M_{Ant} : The positions matrix of ants
 $A_{i,j}$: The value of j th dimension of position value of the i th ant
 f_j : Fitness function of ant
 M_{OA} : Fitness value matrix of ants
 $M_{Antlion}$: Position matrix of antlion
 $AL_{i,j}$: The value of j th dimension of position of the i th antlion
 fL_j : Fitness function of antlion
 M_{OAL} : Fitness value matrix of ant lions

R_i^t : Normalized random walk displacement
 a_i : Lowest random walk displacement of i th ant
 b_i : Highest random walk displacement of i th ant
 c_i^t : The lowest of random walk displacement of i th ant at current iteration
 d_i^t : The highest of random walk displacement of i th ant at current iteration
 c^t : The lowest of random walk displacement of all ant at current iteration
 d^t : The highest of random walk displacement of all ant at current iteration
 I : Ration of the shrinking of random region
 w : Index varying with the current iteration
 e_0 : Elite individual
 e_{new} : New individual
 e_k : New individual.

Data Availability

The data used to support the findings of this study have been uploaded to website (<https://web.calce.umd.edu/batteries/data.htm?type3>) which can be downloaded freely.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the National Key R&D Program of China, 2017YFC0803904, China Postdoctoral Science Foundation, 2018T111006 and 2017M613034, Postdoctoral Science Foundation of Shaanxi Province, 2017BSHEDZZ36, Shaanxi Province Industrial Innovation Chain Project, 2018ZDCXL-GY-05-03-01, Shaanxi Provincial Key Research and Development Plan Project, 2018ZDXM-GY-082, and Shaanxi Innovative Talents Promotion Plan Project, 2018KJXX-005.

References

- [1] X. Zhao, Q. Yu, J. Ma, Y. Wu, M. Yu, and Y. Ye, "Development of a representative EV urban driving cycle based on a k-means and SVM hybrid clustering algorithm," *Journal of Advanced Transportation*, vol. 2018, Article ID 1890753, 18 pages, 2018.
- [2] H. Chen, B. Xie, J. Ma, and Y. Chen, "NOx emission of biodiesel compared to diesel: Higher or lower?" *Applied Thermal Engineering*, vol. 137, pp. 584–593, 2018.
- [3] X. Zhao, J. Ma, S. Wang, Y. Ye, Y. Wu, and M. Yu, "Developing an electric vehicle urban driving cycle to study differences in energy consumption," *Environmental Science and Pollution Research*, pp. 1–15, 2018.
- [4] Y. Zou, T. Liu, D. Liu, and F. Sun, "Reinforcement learning-based real-time energy management for a hybrid tracked vehicle," *Applied Energy*, vol. 171, pp. 372–382, 2016.
- [5] F. Zheng, Y. Xing, J. Jiang, B. Sun, J. Kim, and M. Pecht, "Influence of different open circuit voltage tests on state of charge online estimation for lithium-ion batteries," *Applied Energy*, vol. 183, pp. 513–525, 2016.

- [6] X. Zhao, Q. Yu, M. Yu, and Z. Tang, "Research on an equal power allocation electronic differential system for electric vehicle with dual-wheeled-motor front drive based on a wavelet controller," *Advances in Mechanical Engineering*, vol. 10, no. 2, pp. 1–24, 2018.
- [7] R. Xiong, F. Sun, X. Gong, and C. Gao, "A data-driven based adaptive state of charge estimator of lithium-ion polymer battery used in electric vehicles," *Applied Energy*, vol. 113, pp. 1421–1433, 2014.
- [8] Y. Xing, E. W. M. Ma, K. L. Tsui, and M. Pecht, "Battery management systems in electric and hybrid vehicles," *Energies*, vol. 4, no. 11, pp. 1840–1857, 2011.
- [9] Y. Xing, W. He, M. Pecht, and K. L. Tsui, "State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures," *Applied Energy*, vol. 113, pp. 106–115, 2014.
- [10] S. Piller, M. Perrin, and A. Jossen, "Methods for state-of-charge determination and their applications," *Journal of Power Sources*, vol. 96, no. 1, pp. 113–120, 2001.
- [11] J. H. Aylor, A. Thieme, and B. W. Johnson, "A battery state-of-charge indicator for electric wheelchairs," *IEEE Transactions on Industrial Electronics*, vol. 39, no. 5, pp. 398–409, 1992.
- [12] T. Liu, D. Chen, and C. Fang, "Design and implementation of a battery charger with a state-of-charge estimator," *International Journal of Electronics*, vol. 87, no. 2, pp. 211–226, 2000.
- [13] R. Xiong, H. He, F. Sun, X. Liu, and Z. Liu, "Model-based state of charge and peak power capability joint estimation of lithium-ion battery in plug-in hybrid electric vehicles," *Journal of Power Sources*, vol. 229, pp. 159–169, 2013.
- [14] X. Zhao, S. Wang, J. Ma, and Q. Yu, "Identification of driver's braking intention based on a hybrid model of GHMM and GGAP-RBFNN," *Neural Computing & Applications*, pp. 1–24, 2018.
- [15] B. Cheng, Z. Bai, and B. Cao, "State of charge estimation based on evolutionary neural network," *Energy Conversion and Management*, vol. 49, no. 10, pp. 2788–2794, 2008.
- [16] X. Hu and F. Sun, "Fuzzy clustering based multi-model support vector regression state of charge estimator for lithium-ion battery of electric vehicle," in *Proceedings of the 2009 International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2009*, pp. 392–396, China, August 2009.
- [17] P. Singh and D. Reisner, "Fuzzy logic-based state-of-health determination of lead acid batteries," in *Proceedings of the 24th International Telecommunications Energy Conference (INTELEC 2002)*, pp. 583–590, IEEE, Canada, October 2002.
- [18] Y. Shen, "Adaptive online state-of-charge determination based on neuro-controller and neural network," *Energy Conversion and Management*, vol. 51, no. 5, pp. 1093–1098, 2010.
- [19] S. Lee and J. Kim, "Discrete wavelet transform-based denoising technique for advanced state-of-charge estimator of a lithium-ion battery in electric vehicles," *Energy*, vol. 83, pp. 462–473, 2015.
- [20] Y. Li, C. Wang, and J. Gong, "A wavelet transform-adaptive unscented Kalman filter approach for state of charge estimation of LiFePo₄ battery," *International Journal of Energy Research*, vol. 42, no. 2, pp. 587–600, 2018.
- [21] H. Pan, Z. Lü, W. Lin, J. Li, and L. Chen, "State of charge estimation of lithium-ion batteries using a grey extended Kalman filter and a novel open-circuit voltage model," *Energy*, vol. 138, pp. 764–775, 2017.
- [22] C. Weng, J. Sun, and H. Peng, "A unified open-circuit-voltage model of lithium-ion batteries for state-of-charge estimation and state-of-health monitoring," *Journal of Power Sources*, vol. 258, pp. 228–237, 2014.
- [23] B. Xia, C. Chen, Y. Tian, M. Wang, W. Sun, and Z. Xu, "State of charge estimation of lithium-ion batteries based on an improved parameter identification method," *Energy*, vol. 90, pp. 1426–1434, 2015.
- [24] J. Li, L. Wang, C. Lyu, and M. Pecht, "State of charge estimation based on a simplified electrochemical model for a single LiCoO₂ battery and battery pack," *Energy*, vol. 133, pp. 572–583, 2017.
- [25] J. Li, L. Wang, C. Lyu, H. Wang, and X. Liu, "New method for parameter estimation of an electrochemical-thermal coupling model for LiCoO₂ battery," *Journal of Power Sources*, vol. 307, pp. 220–230, 2016.
- [26] J. Li, Q. Lai, L. Wang, C. Lyu, and H. Wang, "A method for SOC estimation based on simplified mechanistic model for LiFePO₄ battery," *Energy*, vol. 114, pp. 1266–1276, 2016.
- [27] Z. Chen, S. Qiu, M. A. Masrur, and Y. L. Murphey, "Battery state of charge estimation based on a combined model of extended kalman filter and neural networks," in *Proceedings of the International Joint Conference on Neural Network, IJCNN 2011*, pp. 2156–2163, USA, August 2011.
- [28] J. Han, D. Kim, and M. Sunwoo, "State-of-charge estimation of lead-acid batteries using an adaptive extended Kalman filter," *Journal of Power Sources*, vol. 188, no. 2, pp. 606–612, 2009.
- [29] R. Xiong, H. He, F. Sun, and K. Zhao, "Evaluation on state of charge estimation of batteries with adaptive extended kalman filter by experiment approach," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 108–117, 2013.
- [30] Y. Li, C. Wang, and J. Gong, "A multi-model probability SOC fusion estimation approach using an improved adaptive unscented Kalman filter technique," *Energy*, vol. 141, pp. 1402–1415, 2017.
- [31] M. Partovibakhsh and G. Liu, "An adaptive unscented kalman filtering approach for online estimation of model parameters and state-of-charge of lithium-ion batteries for autonomous mobile robots," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 357–363, 2015.
- [32] W. He, N. Williard, C. Chen, and M. Pecht, "State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation," *International Journal of Electrical Power & Energy Systems*, vol. 62, pp. 783–791, 2014.
- [33] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [34] M. A. Hossain, H. R. Pota, S. Squartini, and A. F. Abdou, "Modified PSO algorithm for real-time energy management in grid-connected microgrids," *Renew Energy*, vol. 136, pp. 746–757, 2019.
- [35] W. Zhang, A. Maleki, M. A. Rosen, and J. Liu, "Optimization with a simulated annealing algorithm of a hybrid system for renewable energy including battery and hydrogen storage," *Energy*, vol. 163, pp. 191–207, 2018.
- [36] Z. Chen, S. Zhou, and J. Luo, "A robust ant colony optimization for continuous functions," *Expert Systems with Applications*, vol. 81, pp. 309–320, 2017.
- [37] N. Ghorbani, A. Kasaeian, A. Toopshekan, L. Bahrami, and A. Maghami, "Optimizing a hybrid wind-PV-battery system using GA-PSO and MOPSO for reducing cost and increasing reliability," *Energy*, vol. 154, pp. 581–591, 2018.

- [38] S. M. Abd-Elazim and E. S. Ali, "Load frequency controller design of a two-area system composing of PV grid and thermal generator via firefly algorithm," *Neural Computing and Applications*, vol. 30, no. 2, pp. 607–616, 2018.
- [39] A. S. Oshaba, E. S. Ali, and S. M. Abd Elazim, "PI controller design for MPPT of photovoltaic system supplying SRM via BAT search algorithm," *Neural Computing and Applications*, vol. 28, no. 4, pp. 651–667, 2017.
- [40] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.
- [41] S. Mirjalili, P. Jangir, and S. Saremi, "Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems," *Applied Intelligence*, vol. 46, no. 1, pp. 79–95, 2017.
- [42] E. S. Ali, S. M. Abd Elazim, and A. Y. Abdelaziz, "Ant Lion Optimization Algorithm for optimal location and sizing of renewable distributed generations," *Journal of Renewable Energy*, vol. 101, pp. 1311–1324, 2017.
- [43] T. Tian, C. Liu, Q. Guo, Y. Yuan, W. Li, and Q. Yan, "An improved ant lion optimization algorithm and its application in hydraulic turbine governing system parameter identification," *Energies*, vol. 11, no. 1, p. 95, 2018.
- [44] Z. Wu, D. Yu, and X. Kang, "Parameter identification of photovoltaic cell model based on improved ant lion optimizer," *Energy Conversion and Management*, vol. 151, pp. 107–115, 2017.
- [45] M. A. Hannan, M. S. H. Lipu, A. Hussain, and A. Mohamed, "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations," *Renewable & Sustainable Energy Reviews*, vol. 78, pp. 834–854, 2017.
- [46] X. S. Hu, S. B. Li, and H. Peng, "A comparative study of equivalent circuit models for Li-ion batteries," *Journal of Power Sources*, vol. 198, pp. 359–367, 2012.
- [47] E. Ott, *Chaos in Dynamical Systems*, Cambridge University Press, Cambridge, UK, 2nd edition, 2002.
- [48] A. Kaveh, R. Sheikholeslami, S. Talatahari, and M. Keshvari-Ilkhichi, "Chaotic swarming of particles: A new method for size optimization of truss structures," *Advances in Engineering Software*, vol. 67, pp. 136–147, 2014.
- [49] M. Ragulskis, Z. Navickas, R. Palivonaite, and M. Landauskas, "Algebraic approach for the exploration of the onset of chaos in discrete nonlinear dynamical systems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 11, pp. 4304–4315, 2012.
- [50] G. Ye and X. Huang, "An efficient symmetric image encryption algorithm based on an intertwining logistic map," *Neurocomputing*, vol. 251, pp. 45–53, 2017.
- [51] X. Wang and D. Luan, "A novel image encryption algorithm using chaos and reversible cellular automata," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 11, pp. 3075–3085, 2013.
- [52] T. Q. Duong, "USABC and PNGV test procedures," *Journal of Power Sources*, vol. 89, no. 2, pp. 244–248, 2000.
- [53] W. He, N. Williard, C. C. Chen, and M. Pecht, "Robust and adaptive estimation of state of charge for lithium-ion batteries," *International Journal of Power Electronics*, vol. 62, pp. 783–791, 2014.
- [54] Z. Ma, J. Jiang, W. Shi, W. Zhang, and C. C. Mi, "Investigation of path dependence in commercial lithium-ion cells for pure electric bus applications: aging mechanism identification," *Journal of Power Sources*, vol. 274, pp. 29–40, 2015.
- [55] M. M. Maricq, D. H. Podsiadlik, and R. E. Chase, "Gasoline vehicle particle size distributions: Comparison of steady state, FTP, and US06 measurements," *Environmental Science & Technology*, vol. 33, no. 12, pp. 2007–2015, 1999.

