Int. J. Electrochem. Sci., 17 **(2022)** Article Number: 220214, doi: 10.20964/2022.02.16

International Journal of
ELECTROCHEMICAL
SCIENCE
www.electrochemsci.org

The state of charge estimation of lithium-ions battery using combined multi-population genetic algorithm - BP and Kalman filter methods

Qingyun Ma, Chuanyun Zou, Shunli Wang, Jingsong Qiu

School of Information Engineering, Southwest University of Science and Technology (SWUST), Mianyang, Sichuan, China, 621010
*E-mail: zou_cy@qq.com

Received: 1 November 2021 / *Accepted:* 26 November 2021 / *Published:* 5 January 2022

Accurate estimation of state of charge (SOC) of lithium-ion batteries is the core technology of battery management system and the key to prolong battery life. However, it is difficult to estimate SOC accurately online and the estimation accuracy is not high. Taking ternary lithium battery as the research object, a Back-Propagation (BP) neural network optimized by multi-population genetic algorithm (MPGA) is proposed to compensate the nonlinear errors caused by EKF in the process of linearization and to avoid genetic algorithm (GA) immature phenomenon. The BP neural network optimized by MPGA is used to predict the EKF error at k time, so as to compensate the nonlinear error at extended Kalman K time. Adaptive FFRLS is used to identify model parameters, so that the algorithm can be identified online. The accuracy range of the proposed algorithm is less than 0.0121 verified by dynamic stress testing (DST) results, and the maximum error and average error are small. The proposed algorithm can track the theoretical value of SOC more effectively, and the SOC estimated by the proposed algorithm is stable.

Keywords: State of charge; Multi-population genetic algorithm; Adaptive FFRLS; Lithium-ion batteries

1. INTRODUCTION

In order to protect the environment and save non-renewable energy, every country is vigorously developing new energy. Lithium-ion has been widely used in new energy, especially in electric vehicles, large-scale energy storage, special robots play an extremely important role. In order to protect the battery and prolong the service life of the battery, the concept of battery management system (BMS)[1] is put forward, which includes SOC, state of health (SOH), state of power (SOP), state of energy (SOE), etc. The factors that affect SOC include temperature, aging degree of battery, noise of battery operating

environment, etc.

The SOC displays the current remaining charge of the battery. Let the user know roughly how long the battery will last and when it will be recharged[2]. In recent years, the common estimation methods of SOC are equivalent circuit model[3] (ECM) and data-driven methods[4].

The equivalent circuit method is widely used because of its good accuracy and robustness. The basic principle of equivalent modeling is to use resistors, capacitors and other components to simulate the chemical reactions in batteries. The equivalent circuit includes Thevenin[5], Rint[6], resistance capacitance(RC)[7], PNGV[8], fractional[9] and other models. In the equivalent circuit model, the increase in the number of RC will improve the accuracy of the model, but also make the calculation more and more complicated[10].For the established equivalent model, there are usually offline parameter identification and online parameter identification to calculate RC. However, offline parameter identification is generally not adopted because it cannot identify the model parameters of the battery in operation. Online parameter identification generally includes basic recursive least squares (RLS)[11], RLS with forgetting factors (FFRLS)[12], and RLS with adaptive forgetting factors (AFFRLS)[13]. Specific SOC estimation methods include extended Kalman (EKF)[14], untraced Kalman (UKF)[15, 16], Cubature Kalman filter (CKF)[17], H Infinity Filter(HIF)[18, 19].

Based on the data-driven approach, there is no need to model the equivalent circuit of the battery. It also does not need to know the exact parameters inside the battery, and has good applicability to different batteries. Data-driven methods can find hidden nonlinear relationships between input and output vectors, such as SOC and temperature, current, voltage, etc. Commonly used data driven models include BP neural network, extreme learning machine(ELM)[20], support vector machine(SVM)[21], recurrent neural network (RNN) related to the state of the previous moment, long and short-term memory model (LSTM)[22], gated cyclic unit (GRU)[23], bidirectional cyclic network and so on.

Reference [24] proposed FFRLS for parameter identification of the equivalent model, but as the forgetting factor is a fixed value, it cannot well track the real-time characteristics of the battery system. Reference [25] uses the improve EKF algorithm to estimate SOC, but does not consider the case that external noise is changing in real time. Reference [26] used BP to estimate SOC, but the results of each run were very different due to the uncertain initial weight of BP, and the estimation accuracy was not high. In reference [27], GA-BP was used to estimate SOC, but GA immaturity was not taken into account, resulting in low accuracy of BP prediction results.

The algorithm proposed in this paper overcomes the above shortcomings. Compared with other models, it has the following advantages :① AFFRLS is used to identify the parameters of the equivalent model. By changing the forgetting factor in real time, the internal characteristics of the battery can be better tracked. ② AEKF algorithm is adopted to overcome the real-time change of noise. ③ MPGA was used to optimize the initial weight of BP. The result of BP prediction is confirmed, and the immature phenomenon of GA is avoided. ④ Combine BP and AEKF algorithm. It avoids the over-dependence of the equivalent model method on the model accuracy, overcomes the need of BP neural data training and increases the estimation accuracy of the algorithm.

2. DESCRIPTION OF SOC ESTIMATION METHODS

The estimation method of SOC involved in this paper is shown in Figure 1.

Method A is based on AFFRLS equivalent modeling and EKF estimation method[28]. Method B adopts the method of adaptive noise on the basis of A, namely AEKF[29]. Method C uses BP neural network to predict the AEKF error at time k, so as to compensate the error[30]. Method D uses GA to optimize BP on the basis of Method C, and not only determines the weight of BP, but also immobilized the results of BP estimation, which can greatly increase the prediction ability of BP[27]. On the basis of Method D, method E introduces multi-population probability to increase species diversity and avoid immature convergence.

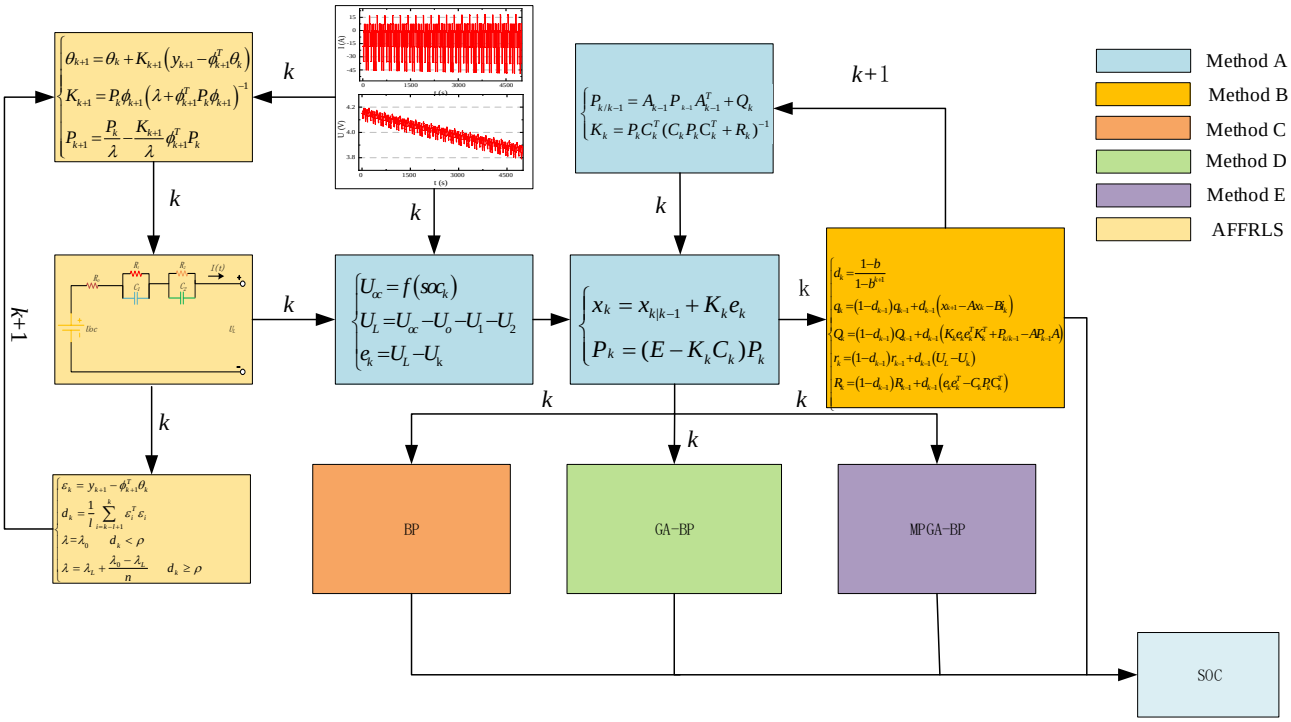


Figure 1. The flow of all the algorithms

2.1 EKF and AEKF estimate SOC (Method A and Method B)

2.1.1 Establishment of equivalent model

When selecting the equivalent model, the model containing RC circuit can be selected, but it should not be too many RC circuits, because too many will lead to more and more complicated calculation. In this paper, a set of RC circuits are added the basis of the common Thevenin equivalent model. The order of RC is not high, and the calculation degree is not complicated while the estimation accuracy is improved. The improved Thevenin equivalent model is shown in Figure 3.

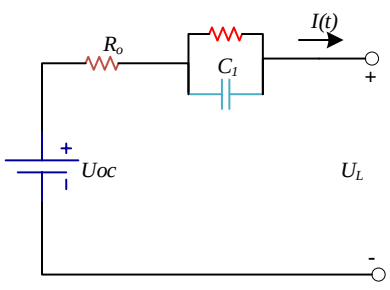


Figure 2. Thevenin's equivalent model

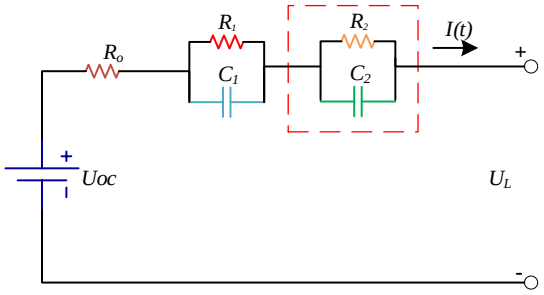


Figure 3. Improved Thevenin equivalent model

In Figure 2, U_{OC} indicates the open circuit voltage. U_L indicates the terminal voltage. R_0 is ohmic internal resistance. C_1 and C_2 represent the polarization capacitance, while R_1 and R_2 represent the polarization resistance. According to Kirchofsky's law, the current and voltage are shown in Equation 1.

$$\begin{cases} U_L = U_{oc} - U_o - U_1 - U_2 \\ \dfrac{dU_1}{dt} = -\dfrac{U_1}{R_1 C_1} + \dfrac{I}{C_1} \\ \dfrac{dU_2}{dt} = -\dfrac{U_2}{R_2 C_2} + \dfrac{I}{C_2} \end{cases} \tag{1}$$

In Figure 3, the open-circuit voltage can be represented by SOC. Combining with equation (2), the discrete state space equation can be obtained.

$$SOC = SOC_0 - \frac{1}{C}\int \eta I dt \tag{2}$$

$$\begin{bmatrix} SOC(k+1) \\ U_1(k+1) \\ U_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{-\frac{\Delta t}{\tau_1}} & 0 \\ 0 & 0 & e^{-\frac{\Delta t}{\tau_2}} \end{bmatrix} \begin{bmatrix} SOC(k) \\ U_1(k) \\ U_2(k) \end{bmatrix} + \begin{bmatrix} -\dfrac{\eta \Delta t}{Q_N} \\ R_1\left(1 - e^{-\frac{T}{\tau_1}}\right) \\ R_2\left(1 - e^{-\frac{T}{\tau_2}}\right) \end{bmatrix} I_k + w_k \tag{3}$$

$$U_L(k) = U_{oc}(k) - R_0 I(k) + \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix}^T \begin{bmatrix} SOC(k) \\ U_1(k) \\ U_2(k) \end{bmatrix} + v_k \tag{4}$$

η is coulomb efficiency, Δt is sampling time interval, w is state error, v is measurement error.

2.1.2 Online parameter identification based on adaptive FFRLS

Forgetting factor is introduced on the basis of RLS, which can avoid data saturation, reduce the influence of some data, and strengthen the influence of recent data. If the forgetting factor is too large,

the ability to weaken the old data is very small. If the forgetting factor is too small, the tracking ability will be enhanced but the ability to suppress noise will be greatly reduced. Therefore, the FFRLS of the adaptive forgetting factor is proposed, and the recursive formula is shown in Equation 5.

$$\begin{cases} \theta_{k+1} = \theta_k + K_{k+1}\left(y_{k+1} - \phi_{k+1}^T \theta_k\right) \\ K_{k+1} = P_k \phi_{k+1}\left(\lambda + \phi_{k+1}^T P_k \phi_{k+1}\right)^{-1} \\ P_{k+1} = \dfrac{P_k}{\lambda} - \dfrac{K_{k+1}}{\lambda}\phi_{k+1}^T P \end{cases} \tag{5}$$

K represents the recursive gain. P represents the covariance. and λ represents the forgetting factor. The adaptive method is shown in Equation (6).

$$\begin{cases} \varepsilon_k = y_{k+1} - \phi_{k+1}^T \theta_k \\ d_k = \dfrac{1}{l}\sum\limits_{i=k-l+1}^{k} \varepsilon_i^T \varepsilon_i \\ \lambda = \lambda_0 \qquad d_k < \rho \\ \lambda = \lambda_L + \dfrac{\lambda_0 - \lambda_L}{n} \qquad d_k \geq \rho \qquad k = 0,1,\cdots n \end{cases} \tag{6}$$

ε_k is the error between the predicted value and the real value. l is the window opening size. λ_L is the minimum forgetting factor, and ρ is the threshold value. When d_k is greater than the threshold value, the forgetting factor immediately decreases to λ_L. With the increase of data, the forgetting factor gradually increases, so as to increase the anti-noise ability.

2.1.3 EKF model of SOC

The basic principle of EKF algorithm for SOC estimation is shown below.

Step 1: Calculate the predicted value and the corresponding covariance matrix at time K.

$$\begin{cases} x_{k/k-1} = A_{k-1} x_{k-1} + B_{k-1} i_{k-1} + q_k \\ P_{k/k-1} = A_{k-1} P_{k-1} A_{k-1}^T + Q_k \end{cases} \tag{7}$$

Where, A *refers to* the state transition matrix. B refers to the control matrix. P refers to the error covariance, and Q_k refers to the external noise.

Step 2: Calculate kalman gain.

$$K_k = P_k C_k^T (C_k P_k C_k^T + R_k)^{-1} \tag{8}$$

K refers to Kalman gain. C refers to system measurement matrix and R_k refers to measurement noise.

Step 3: Covariance with the new state predicted value.

$$\begin{cases} e_k = U_L - U_k - r_k \\ x_k = x(k \mid k-1) + K_k e_k \\ P_k = (E - K_k C_k) P_k \end{cases} \tag{9}$$

U_L refers to the open-circuit voltage value measured in real time, U_k refers to the predicted open-circuit voltage value obtained according to the predicted value and KVL theorem.

2.1.4 AEKF estimating SOC

In the actual environment of lithium battery, the noise changes from time to time, while the extended Kalman ignores the high-order term to linearize the whole system, but does not take into account the influence of the noise change on SOC estimation. Therefore, an adaptive estimation method of noise statistical characteristics is proposed, as shown in the equation (10).

$$\begin{cases} d_k = \dfrac{1-b}{1-b^{k+1}} \\ q_k = (1-d_{k-1})q_{k-1} + d_{k-1}\left(x_{k+1} - Ax_k - Bi_k\right) \\ Q_k = (1-d_{k-1})Q_{k-1} + d_{k-1}\left(K_k e_k e_k^T K_k^T + P_{k/k-1} - AP_{k-1}A\right) \\ r_k = (1-d_{k-1})r_{k-1} + d_{k-1}\left(U_L - U_k\right) \\ R_k = (1-d_{k-1})R_{k-1} + d_{k-1}\left(e_k e_k^T - C_k P_k C_k^T\right) \end{cases} \quad (10)$$

b is the forgetting factor, which generally ranges from 0.96 to 0.99.

2.2 BP, GA-BP and MPGA-BP (Methods C, D and E)

2.2.1 Improved BP neural network

BP is a network that generalizes W-H training rules to train weights of nonlinear functions. BP consists of input, hidden and output layers, among which the hidden layer may be one layer or several layers. BP network is mainly used for: function approximation, pattern recognition, classification, data compression. A simple BP is shown in Figure 4.

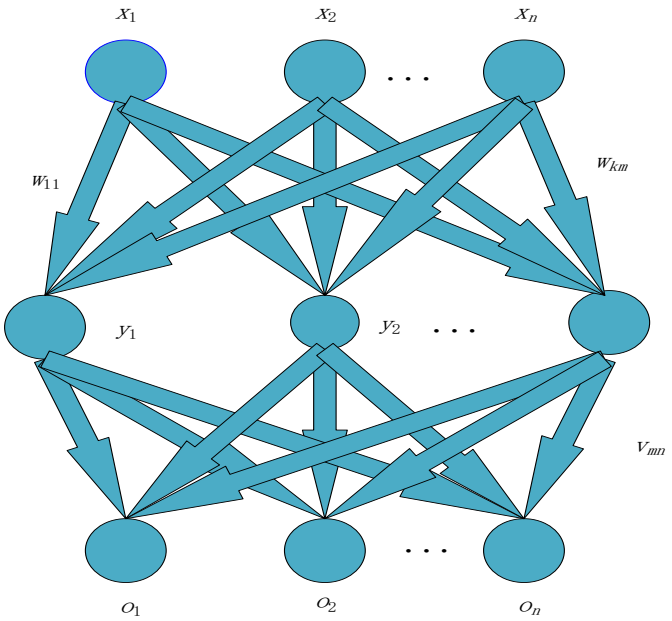


Figure 4. Relatively simple BP network

In Figure 4, *X* represents the input layer, *Y* represents the hidden layer, *O* represents the output

layer, *W and V represent the weights between the different layers.*

1. Transfer of information from input layer to output layer.

① The output of the ith neuron in the hidden layer is shown in Equation (11).

$$h_i = f1\left(\sum_{j=1}^{r} w1_{ij}\, p_j + b1_i\right) \qquad (11)$$

h refers to the output value of the hidden layer. *f*1 refers to the activation function of the hidden layer. w refers to the weights between the input and output layers. p is the input vector. b refers to the deviation of hidden layer neurons. r is the number of input neurons.

②The output of the Kth neuron in the output layer is shown in Equation (12).

$$o_k = f2\left(\sum_{i=1}^{m} w2_{kj}\, h_j + b2_k\right) \qquad (12)$$

o refers to the output value of the output layer. *f*2 refers to the activation function of the output layer. w2 refers to the weight between the hidden and output layers. b2 refers to the neuron bias. m refers to the number of neurons in the hidden layer.

③Let's define the error function as zero.

$$E(W,B) = \frac{1}{2}\sum_{k=1}^{n}\left(t_k - o_k\right)^2 \qquad (13)$$

E is the error function. t is the expected value. n refers to the number of neurons in the output layer.

2. Use the gradient descent method to calculate the weight change and error back propagation

① Weight change of output layer

$$\Delta w2_{ki} = -\eta\frac{\partial E}{\partial w2_{ki}} = -\eta\frac{\partial E}{\partial ok}\cdot\frac{\partial ok}{\partial w2_{ki}} = \eta\cdot\delta_{ki}\cdot h_i \qquad (14)$$

$$\begin{cases}\delta_{ki} = \left(t_k - o_k\right)\cdot f2' = e_k\cdot f2' \\ e_k = t_k - o_k\end{cases} \qquad (15)$$

Similarly, equation (16) can be obtained.

$$\Delta b2_{ki} = -\eta\frac{\partial E}{\partial b2_{ki}} = \eta\left(t_k - o_k\right)\cdot f2' = \eta\cdot\delta_{ki} \qquad (16)$$

②The weight change of hidden layer is shown in Equation (17).

$$\Delta w1_{ij} = -\eta\frac{\partial E}{\partial w1_{ij}} = -\eta\frac{\partial E}{\partial o_k}\cdot\frac{\partial o_k}{\partial h_i}\cdot\frac{\partial h_i}{\partial w1_{ij}} = \eta\cdot\delta_{ij}\cdot p_j \qquad (17)$$

$$\begin{cases}\delta_{ij} = e_i\cdot f1' \\ e_i = \sum_{k=1}^{n}\delta_{ki} w2_{ki} \\ \delta_{ki} = e_k\cdot f2' \\ e_k = t_k - o_k\end{cases} \qquad (18)$$

Similarly, equation (19) can be obtained.

$$\Delta b1_i = \eta \delta_{ij} \tag{19}$$

The general BP algorithm uses the gradient descent method for iterative training, but the gradient descent method only uses the first order reciprocal of the objective function for training. The number of iterations is more, the iteration time is longer, and the final error is not ideal. Therefore, this paper proposes the method of numerical optimization to train. This method utilizes not only the first order inverse of the objective function but also the second order inverse information of the objective function. The specific description is shown in Formula (20).

$$\begin{cases} f\left(X^{(k+1)}\right) = \min f\left(X^{(k)} + \eta^{(k)} S\left(X^{(k)}\right)\right) \\ X^{(k+1)} = X^{(k)^{\eta}} + \eta^{(k)} S\left(X^{(k)}\right) \end{cases} \tag{20}$$

$X^{(k)}$ is the vector composed of network ownership value and bias value; $S\left(X^{(k)}\right)$ is the search direction in the vector space of the components of X； $\eta^{(k)}$ is in the direction of $S\left(X^{(k)}\right)$, making $f\left(X^{(k+1)}\right)$ a minimal step。The optimization of network weight can be divided into two steps：First, the optimal search direction $S\left(X^{(k)}\right)$ of the current iteration is determined, and then the optimal iteration step is sought in this direction. The search direction is shown in Equation (21)。

$$S\left(X^{(k)}\right) = -\left(H^{(k)} + \lambda^{(k)} I\right)^{-1} \nabla f\left(X^{(k)}\right) \tag{21}$$

$H^{(k)}$ is the Hessen matrix, which is a second-order reciprocal matrix. At the beginning, λ takes a large value, which is equivalent to gradient descent with a small step size; As the optimal approaches, λ decreases to zero. In general, when $f\left(X^{(k+1)}\right)$ is less than $f\left(X^{(k)}\right)$ it decreases by λ; Otherwise it goes up by λ.

2.2.2 GA-BP and MPGA-BP algorithms

When BP is predicting the nonlinear error of EKF, due to the uncertainty of weight, the prediction result is stochastic. Therefore, GA is proposed to optimize the weight of BP to find the optimal initial weight of BP neural network.

Genetic algorithm is an evolutionary algorithm, and its basic principle is to imitate the evolutionary law in nature that only organisms constantly evolve to adapt to their own living environment can they not be eliminated. GA encodes the unknown to be optimized into chromosomes. Iterating in a genetic way. In the process of iteration, information in chromosomes is exchanged by selection, crossover and variation. Eventually, a chromosome that matches the target is made. The basic genetic modes of GA are selection, crossover and mutation.

Selection operation: select the chromosomes with high fitness from the old population and put them into the matching set to prepare for the exchange, mutation and generation of new chromosomes in the future. The probability of a chromosome being selected is shown in Equation (22).

$$P_c = \frac{f\left(x_i\right)}{\sum f\left(x_i\right)} \tag{22}$$

P_c is the probability of being selected. x_i is the ith chromosome of population. $f(x_i)$ is the fitness value of chromosome i. $\sum f(x_i)$ is the sum of the fitness of all chromosomes in the population.

Exchange operation: two chromosomes are randomly selected, one or more points are randomly designated for exchange, and new chromosomes are generated.

Mutation: the simulation of biological changes in the natural environment, resulting in gene mutations. Mutations produce chromosome diversity, avoiding early maturation in evolution and trapping in local extremes.

The general structure of genetic algorithm is shown in Figure 5.

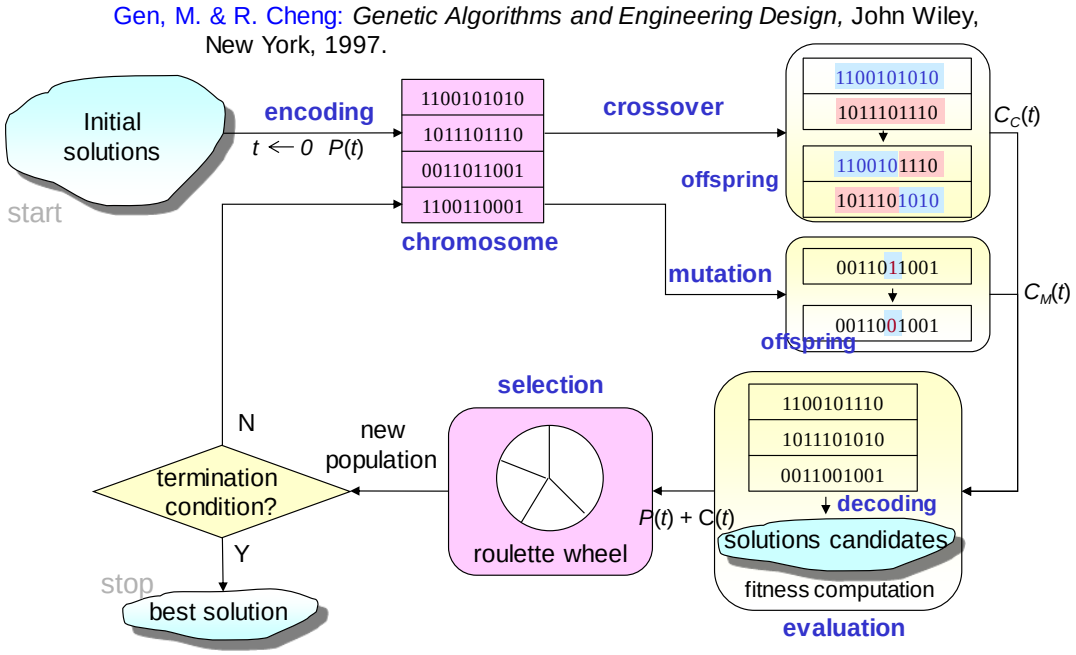


Figure 5. The general structure of genetic algorithms

The basic steps of GA include coding, generation of original population, fitness calculation, selection, crossover and variation. The GA flow is shown in figure 6.

GA is a kind of algorithm which includes both local search and global search. GA has strong anti-interference and global search ability because it is independent of gradient when optimizing parameters. However, immature phenomenon is a common phenomenon of GA, which is mainly manifested in the cessation of evolution when all individuals in the population tend to the same state, and the final result of evolution cannot reach the expected goal. To solve this problem, multi-population genetic algorithm was proposed to optimize the immature phenomenon of standard genetic algorithm (SGA).

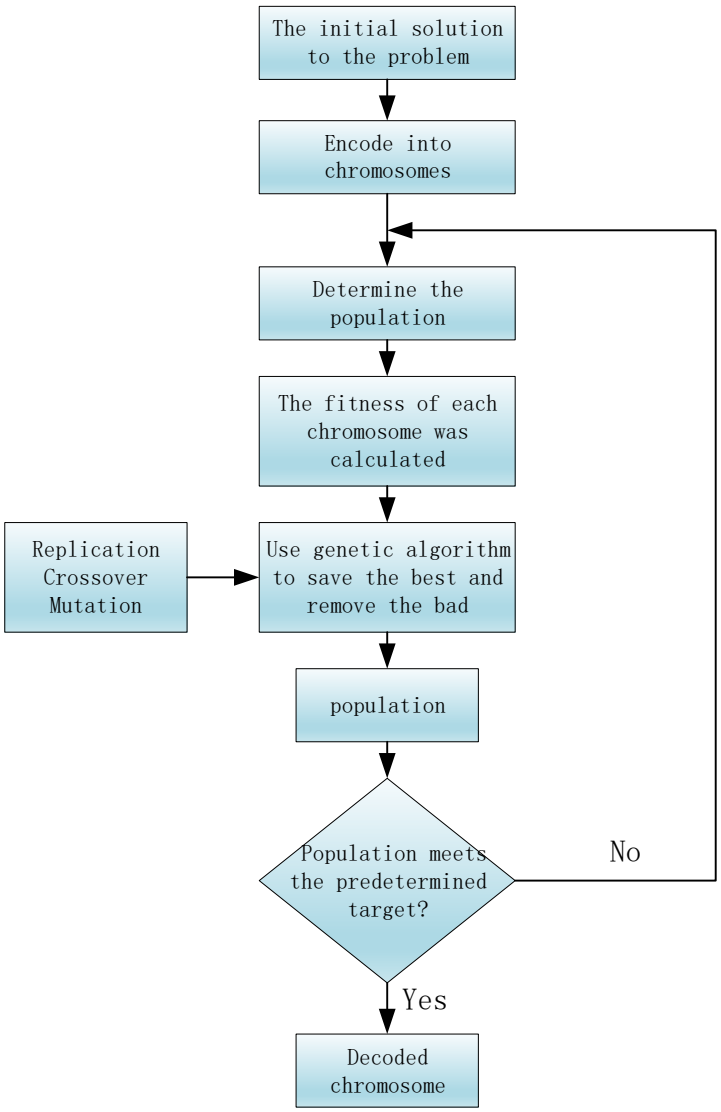


Figure 6. Genetic algorithm flow chart

MPGA puts forward the following advantages based on SGA:

(1) The phenomenon that SGA has only one population is changed and the concept of simultaneous evolution of multiple populations is introduced. The initial value between different populations is different, so multiple populations can search the same problem at the same time.

(2) In order to exchange information between different populations, a migration operator is introduced. The migration operator makes each population coevolve and solves the problem of individual evolution of each population with different initial values.

(3) At the end of each iteration, the optimal individuals of each population are compared. The optimal individuals are comparing with each other to get the optimal individuals of all populations.

In Figure 7, the evolution mechanism of population 1-N is conventional SGA, which adopts roulette wheel selection, single point crossover and variation.

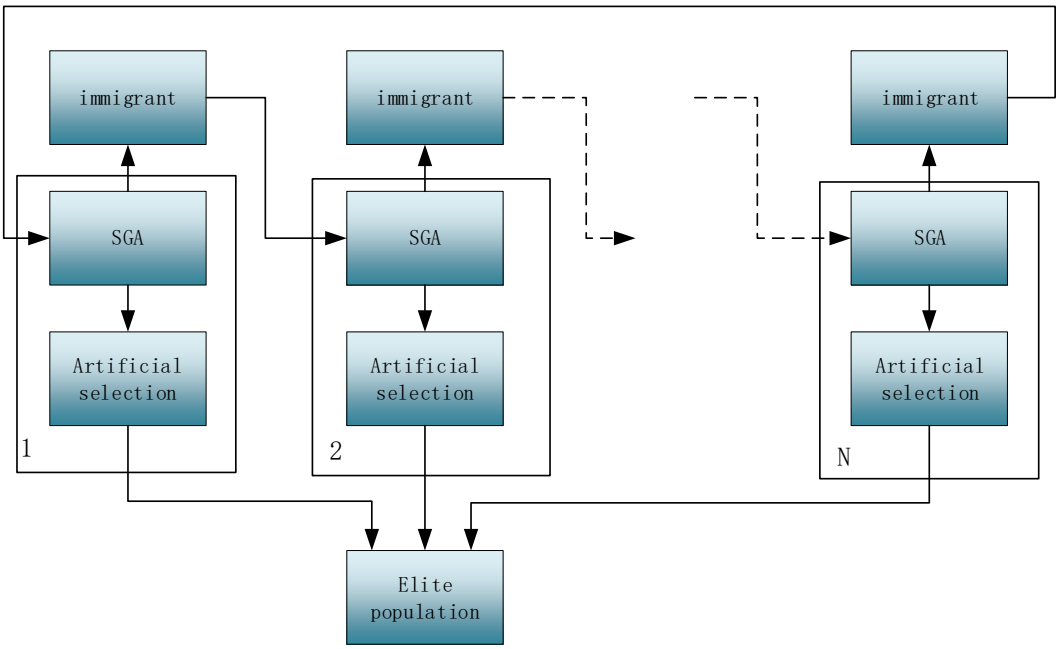


Figure 7. Multi-population genetic algorithm

MPGA sets different parameters such as crossover and mutation probability in different populations, and all populations coevolve. Both global and local search are considered. Each population is independent of each other and is connected by migration operator. Migration operators periodically introduce the optimal individuals in the evolution of various groups into other populations to achieve the exchange of the best individuals in the population. Migration operator is indispensable in MPGA algorithm. Without the migration operator as the intermediary of information exchange, the SGA of different parameters can be calculated for several times, thus losing the essence of MPGA.

GA and MPGA optimized BP neural networks include: the determination of BP structure, GA or MPGA optimization weights and thresholds, BP training and prediction. Among them, BP neural network generally adopts a hidden layer, and the process of BP optimization by GA or MPGA is shown in FIG. 8.

The BP network part consists of four inputs and one output. The number of neurons in input layer, hidden layer and output layer is 4, 10 and 1, so the network structure is 4-10-1. There are 61 parameters to be optimized in the whole BP, and the binary number encoded can be adjusted according to the variation range and expected accuracy of its own parameters. In this paper, the weight and threshold encoding choose 6-bit binary number. In order to make the predicted value closer to the expected value, the error matrix norm of the predicted and expected value is selected as the output of the function. and the fitness is calculated by the objective function. The selection operator adopts random ergodic sampling and the crossover operator selects single point crossover operator.

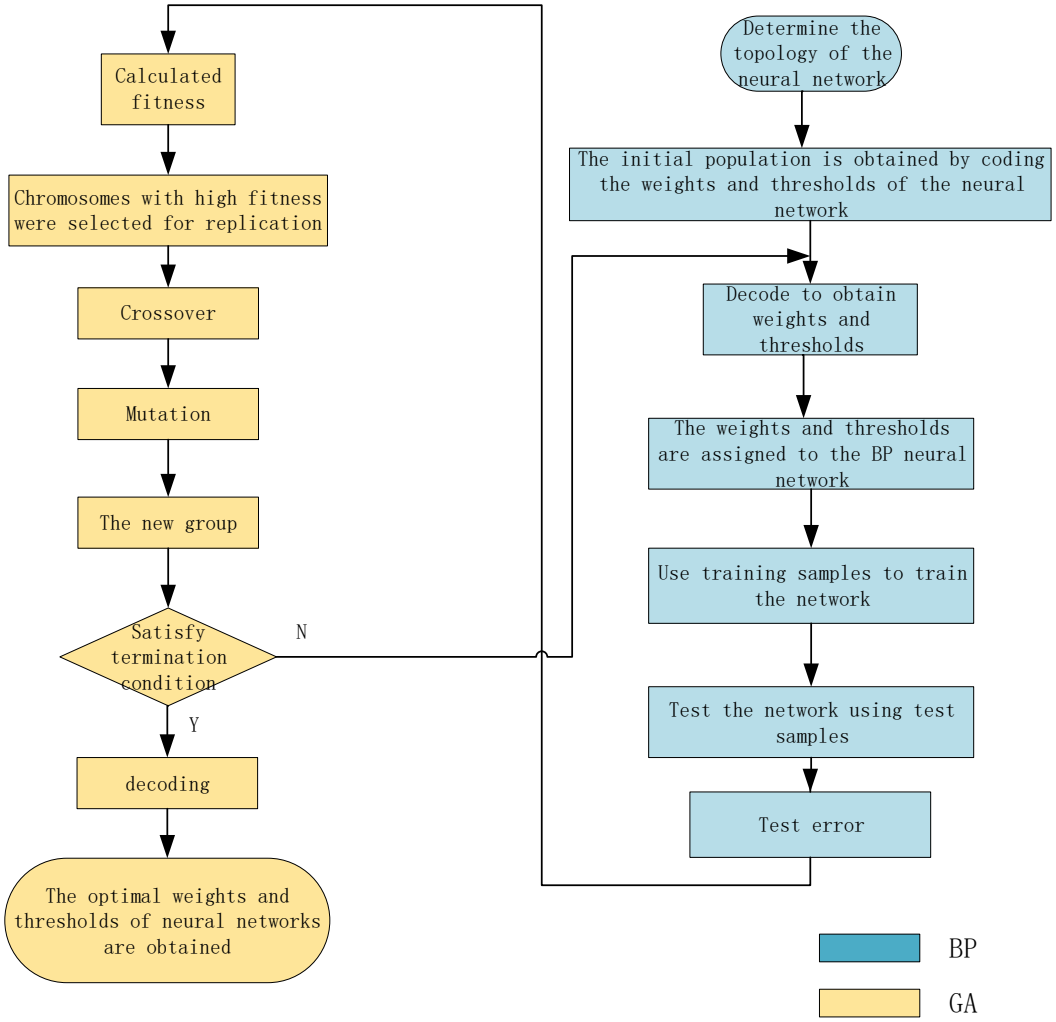


Figure 8. BP flow chart of GA optimization

2.2.3 Methods C, D and E

A three-layer BP with four inputs and one output was established on MATLAB. 10 neurons were selected for the hidden layer, and the expected value was set to 10-5. The maximum iteration was 2000 times. The input vectors of BP network optimized by genetic algorithm are estimated SOC value and polarization voltage at extended Kalman k time, Kalman gain of SOC, and polarization voltage gain. The output vector is the error of the SOC estimated by extended Kalman. The overall process of SOC estimation of methods C, D and E is the same, but the difference is that different BP neural networks are used to predict EKF nonlinear errors. The essence of the algorithm is to use different BP neural networks to compensate the nonlinear errors of SOC estimated by EKF, so that the SOC estimated is closer to the theoretical value. The overall flow chart is shown in Figure 9.

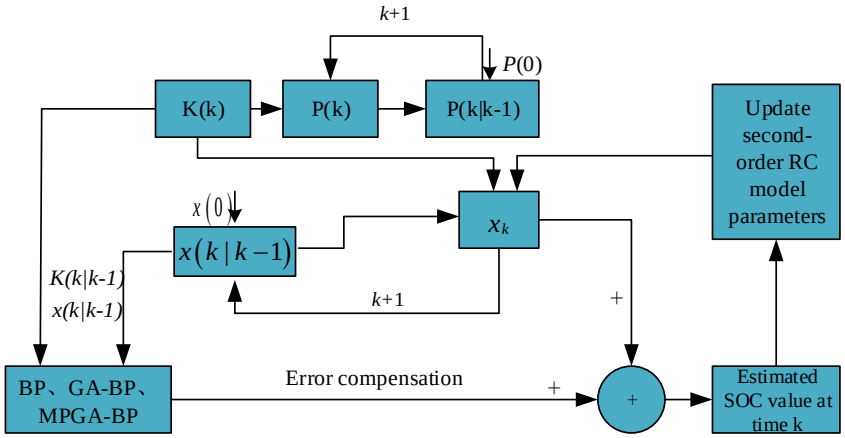


Figure 9. Overall flow chart

In the flow chart of SOC estimation, the SOC nonlinear error compensation formula is shown in Equation (23). Nonlinear error compensation for EKF can further improve the accuracy of SOC and reduce the requirement for model accuracy.

$$SOC_r = SOC_{EKF} + SOC_{BP} \qquad (23)$$

3. EXPERIMENTAL VERIFICATION AND RESULT ANALYSIS

The acquisition of experimental working condition data is shown in Figure 10.

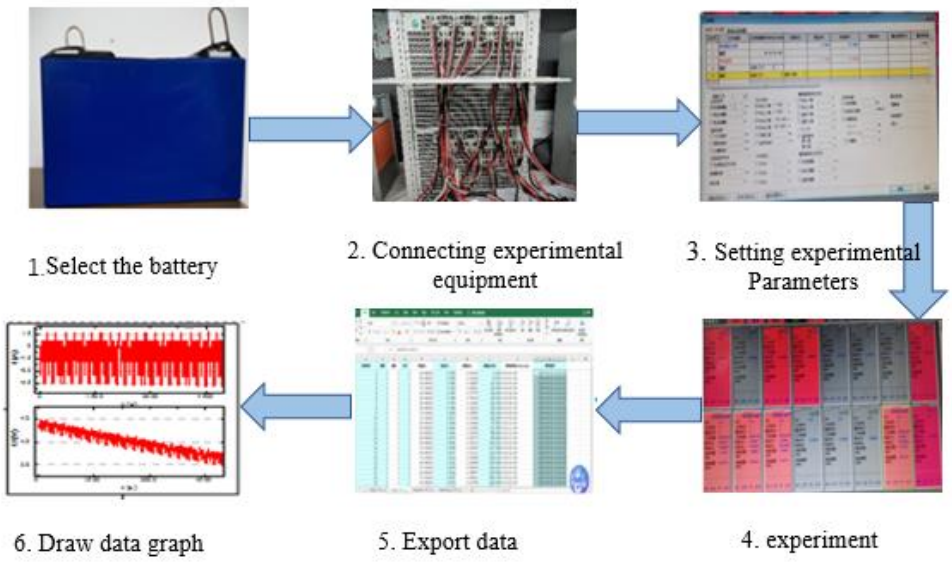


Figure 10. Platform building

The factory capacity of the selected lithium battery is 70Ah, but the actual measured capacity is 69.27ah.The data read interval is 0.1s. The upper limit of voltage is 4.5V, and the lower limit is 2.5V.

The current upper limit is 100A, and the current lower limit is -100A. The battery data were obtained through bTS200-100-104 experimental platform provided by Shenzhen Sub-Keyuan Technology Co., LTD.

3.1 Verify adaptive FFRLS

Online parameter identification was carried out for the improved Thevenin equivalent model. The relationship of SOC-OCV was obtained by using Hybrid Pulse Power Characterization experiments (HPPC). The HPPC test process: First, the lithium battery is charged with a constant current of 1C（69.27A）and a constant voltage of 4.2V. After charging, the battery is put on hold for 40mins. A constant discharge current of 69.27A was carried out for 10s, and the battery was static for 40s after stopping the discharge. The battery was charged at A constant current rate of 69.27A for 10s, and the battery stopped charging and stood for 5min. Constant discharge was carried out at 69.27A rate for 6min and set for 40 mins. Repeat the previous three steps until the battery discharge is complete. HPPC working conditions is shown in FIG. 11.

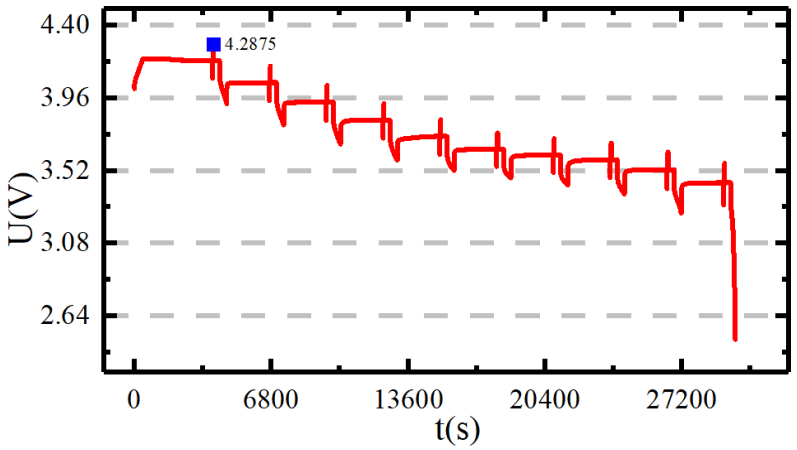


Figure 11. HPPC experimental voltage change diagram

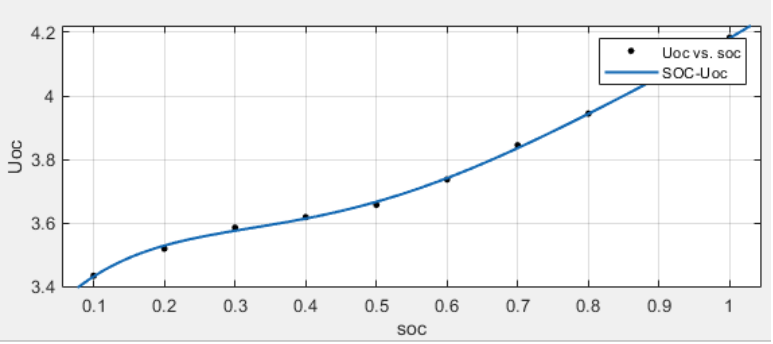
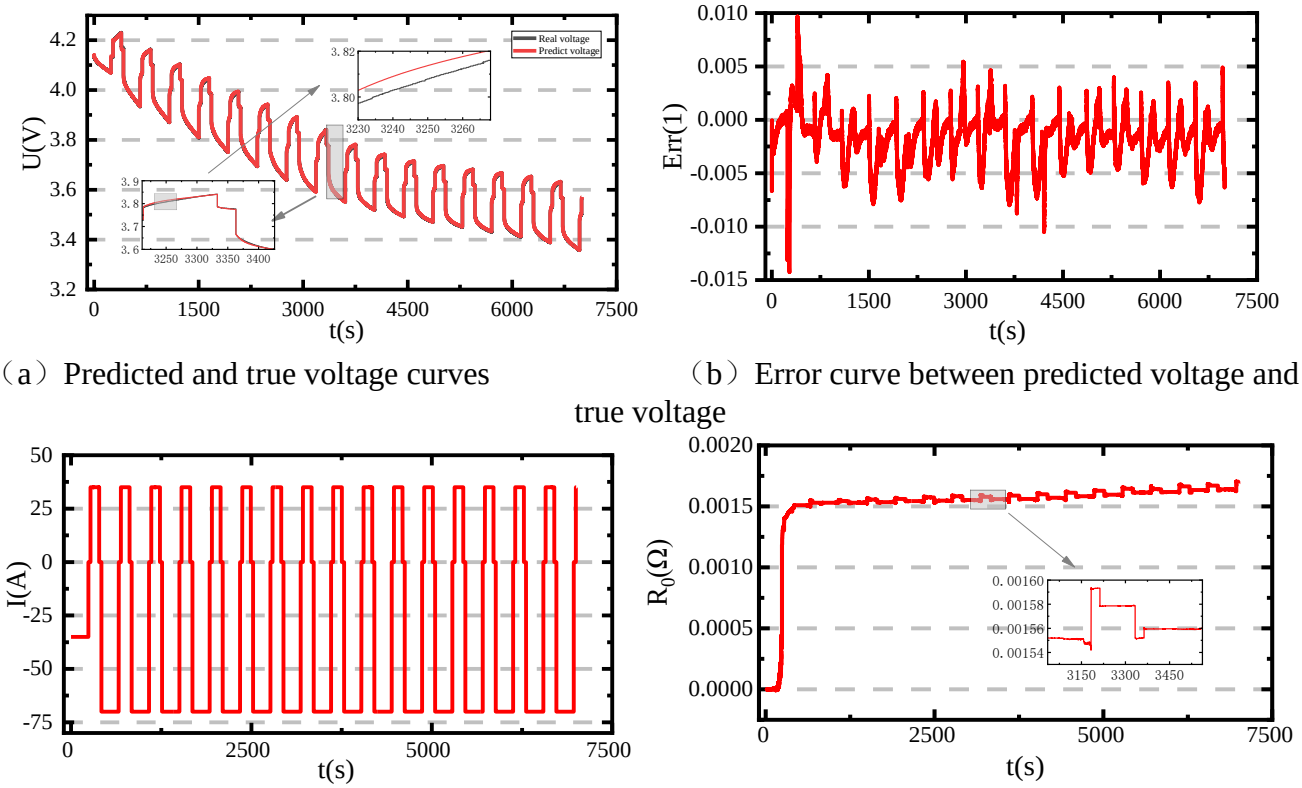


Figure 12. SOC-OCV fitting curve

The OCV corresponding to different SOC time can be obtained by HPPC condition. The SOC-OCV curve was obtained by curve fitting on MATLAB. The curve is shown in figure 12.

AFFRLS is verified by DST condition. The DST experiment procedure: First, the lithium battery is charged with a constant current of 1C （69.27A）and a constant voltage of 4.2V. After charging, the battery is put on hold for 40mins. Then the battery was continuously discharged at 34.35A current for 4min, and the battery was stopped for 40 seconds after discharging. The battery was charged at 34.35A constant current for 2min, and stood for 40s after stopping charging. The battery is continuously discharged at 1C for 4min, and the battery is stationary for 40 seconds. Repeat three steps until the battery discharge is complete. The predicted and real voltages identified online by DST are shown in Figure 13.



（a）Predicted and true voltage curves

（b）Error curve between predicted voltage and true voltage

（c）DST operating current change curve

（d）DST internal resistance R_0 curve

Figure 13. results of DST conditions

According to the error curve in FIG. 13, the maximum error is less than 0.015. The error curve is also stable. The error is stable within 0.005. According to the R_0 change curve in FIG. 13, the internal resistance R_0 tends to zero in the range of 0-240s because the current changes in the range of 0-240s are very small. According to the basic principle of RLS, the size of R_0 is derived from the change of historical current I, so the internal resistance is basically zero in the range of 0-240s. Therefore, it can explain the large error of analog voltage error in the initial interval. At about 435s, the internal resistance R_0 basically reaches a stable value, which proves that the adaptive FFRLS correction performance is good. After that,

the internal resistance R_0 changes periodically, and the change law is basically the same as the current change law, indicating that FFRLS can well track the change of current and voltage, and the identification effect is good.

3.2 Error evolution curve

In the process of BP optimization by GA and MPGA, the optimal individual (the optimal individual in this paper is the individual with the smallest norm of error matrix between predicted value and expected value) is selected from the population of each generation, and the minimum error matrix norm of each generation is recorded. The error curve is shown in FIG. 14 and FIG. 15.

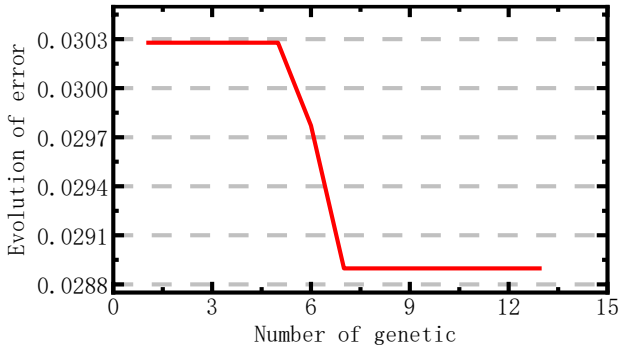


Figure 14. Evolution curve of MPGA

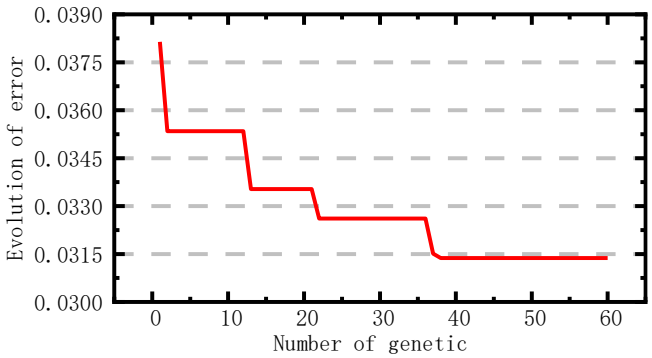


Figure 15. Evolution curve of GA

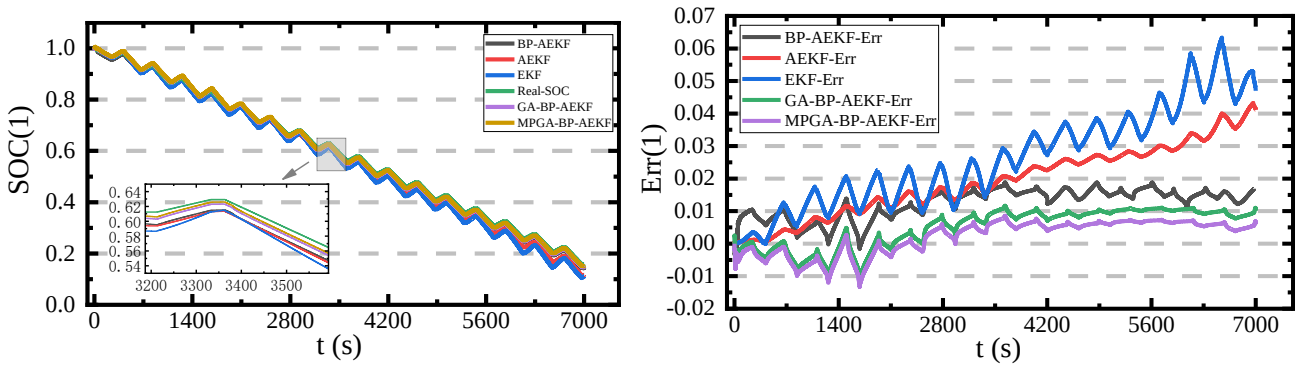
As can be seen from FIG. 14 and 15 the error gradually decreases with the increase of iteration times. MPGA has less iteration times and less final error. GA has more genetic algebra, and the final result is not as good as MPGA. GA not only has more genes, but the final result is not as good as MPGA. According to the error evolution curve in FIG. 15, MPGA can quickly find the optimal value from the 5th generation to the 7th generation, which reflects the strong global and local search ability of MPGA. GA needs more genetic algebra to slowly find the optimal value. Putting the optimal initial value of the

optimization into BP not only increase the prediction ability of the neural network, but also make the results of each operation of the BP, thus increasing the stability of the system.

3.3 DST condition verification

In the practical application of battery, the current of lithium battery is complex and changeable, which requires high dynamic performance of battery, so it is difficult to accurately estimate battery SOC. DST working condition data is obtained through the experimental platform shown in FIG. 10 to verify the reliability of the algorithm. Through the theoretical analysis of each algorithm, the experimental environment of each algorithm is built on MATLAB.

The proposed algorithm is compared with other algorithms by using DST conditional data. Including the EKF algorithm based on AFFRLS proposed in references [30, 31]. Reference [32] proposed an algorithm based on AEKF to overcome noise changes. On the basis of reference [32], the algorithm combining BP and AEKF is adopted in reference [33] to increase the accuracy of the algorithm. Reference [34] proposed that GA was used to optimize the initial value of BP. GA-BP-AEKF algorithm formed on the basis of reference [34]. In order to overcome the immature phenomenon of GA, the algorithm in this paper is proposed on the basis of reference [34]. Under the condition that all initial values are equal, the SOC estimation results of the proposed algorithm and the appellate algorithm are compared. The experimental results are shown in Figure 16.



(a) Estimation of SOC of each algorithm (b) The error of each algorithm

Figure 16. DST SOC estimation results

Table 1 is obtained by collating the data from the graph.

Table 1. Table 1 Collated data of each algorithm

	Maximum Error	Average Error	RSME
EKF	0.06337	0.025995	0.030048
AEKF	0.0432	0.019141	0.02199
BP-AEKF	0.01885	0.01241	0.013179
GA-BP-AEKF	0.0137	0.006836	0.007659
MPGA-BP-AEKF	0.0121	0.004119	0.005564

As shown in Table 1, the three errors of MPGA are much smaller than those of the other four algorithms. The small maximum error indicates that MPGA has strong anti-interference ability and will not have a large mutation at a certain moment. The small mean error indicates that the error generated by MPGA in the whole process of SOC estimation is small. The proposed algorithm can track the actual SOC well in each stage of SOC. The small root mean square error indicates that MPGA has good stability and will not have large fluctuations.

According to figure 16, GA-optimized BP has higher accuracy in SOC estimation than BP neural network alone, and it is not like EKF in the stage of SOC<0.4 that the error increases gradually with the decrease of SOC. BP optimized by MPGA is better than GA in estimating SOC, indicating that MPGA increases the diversity of species under the action of multiple populations and makes the genetic effect better, which is consistent with the error evolution curve.

4. CONCLUSION

1) Based on the improved Thevenin model, this paper adopts the adaptive FFRLS algorithm to carry out online parameter identification. The maximum error of the parameter identification error curve is 0.015 and the error is stable at 0.05. It has strong real-time tracking ability and good online identification effect.

2) In this paper, GA is used to find the initial weight of BP, which makes the prediction result of BP deterministic and greatly increases the accuracy of BP prediction ability. By introducing the concept of MPGA, the immature phenomenon of MGA is solved. Through the evolutionary error curve and SOC error curve, it can be concluded that the final evolutionary error of GA is larger than that of MPGA. The three errors of SOC estimation are also small, indicating that the BP effect of MPGA optimization is better than that of GA optimization.

3) The error of SOC estimation by different methods was compared through DST test of lithium battery measurement. The results show that the three errors of method E are smaller under DST condition. The maximum error is 0.0123. The average error is 0.004119, and the root mean square error is 0.005564. The overall error is also more stable than that of other algorithms. Method E can track the actual SOC of lithium battery more accurately and stably.

FUND PROJECT
National Natural Science Foundation of China(61801407); Quality Education Reform (Youth Development Research) Special Project of SWUST(18xnsu12); Natural Science Foundation of SWUST(17zx7110，18zx7145).

References

1. L. Chen, Z. Z. Wang, Z. Q. Lu, J. Z. Li, B. Ji, H. Y. Wei and H. H. Pan, *Ieee Transactions on Power Electronics*, 33 (2018) 8797.
2. J. Y. Yong, V. K. Ramachandaramurthy, K. M. Tan and N. Mithulananthan, *Renewable & Sustainable Energy Reviews*, 49 (2015) 365.
3. S. Afshar, K. Morris and A. Khajepour, *Ieee Transactions on Control Systems Technology*, 27

(2019) 1907.
4. D. N. T. How, M. A. Hannan, M. S. H. Lipu and P. J. Ker, *Ieee Access*, 7 (2019) 136116.
5. H. W. He, R. Xiong, X. W. Zhang, F. C. Sun and J. X. Fan, *Ieee Transactions on Vehicular Technology*, 60 (2011) 1461.
6. T. Bruen and J. Marco, *Journal of Power Sources*, 310 (2016) 91.
7. X. Zhang, J. L. Lu, S. F. Yuan, J. Yang and X. Zhou, *Journal of Power Sources*, 345 (2017) 21.
8. H. W. He, R. Xiong and J. X. Fan, *Energies*, 4 (2011) 582.
9. C. F. Zou, X. S. Hu, S. Dey, L. Zhang and X. L. Tang, *Ieee Transactions on Industrial Electronics*, 65 (2018) 5951.
10. Y. Shi, S. Ahmad, Q. Tong, T. M. Lim, Z. B. Wei, D. X. Ji, C. M. Eze and J. Y. Zhao, *International Journal of Energy Research*, 45 (2021) 11206.
11. H. Miao, J. J. Chen, L. Mao, K. Q. Qu, J. B. Zhao and Y. J. Zhu, *International Journal of Energy Research*, 45 (2021) 9502.
12. P. Xu, J. Li, C. Sun, G. Yang and F. Sun, *Electronics*, 10 (2021) 122.
13. M. Li, Y. J. Zhang, Z. L. Hu, Y. Zhang and J. Zhang, *Sensors*, 21 (2021) 5698.
14. J. Duan, P. Wang, W. Ma, X. Qiu and S. Fang, *Energies*, 13 (2020) 4197.
15. Y. W. Li, C. Wang and J. F. Gong, *International Journal of Energy Research*, 42 (2018) 587.
16. H. He, H. Qin, Y. Shui and K. Oleksandr, *Energy Procedia*, 61 (2014) 468.
17. J. Peng, J. Luo, H. He and B. Lu, *Applied Energy*, 253 (2019) 113520.1.
18. Q. Zhu, L. Li, X. S. Hu, N. Xiong and G. D. Hu, *Ieee Transactions on Vehicular Technology*, 66 (2017) 10853.
19. C. Z. Liu, Q. Zhu, L. Li, W. Q. Liu, L. Y. Wang, N. Xiong and X. Y. Wang, *Ieee Transactions on Industrial Electronics*, 64 (2017) 8128.
20. M. S. H. Lipu, M. A. Hannan, A. Hussain, M. H. Saad, A. Ayob and M. N. Uddin, *Ieee Transactions on Industry Applications*, 55 (2019) 4225.
21. J. C. A. Anton, P. J. G. Nieto, C. B. Viejo and J. A. V. Vilan, *Ieee Transactions on Power Electronics*, 28 (2013) 5919.
22. F. F. Yang, X. B. Song, F. Xu and K. L. Tsui, *Ieee Access*, 7 (2019) 53792.
23. J. Chen, X. Feng, L. Jiang and Q. Zhu, *Energy*, 227 (2021) 120451.
24. Z. Ren, C. Q. Du, Z. Y. Wu, J. B. Shao and W. J. Deng, *International Journal of Energy Research*, 45 (2021) 13692.
25. L. Guo, J. Li and Z. Fu, *Energy Procedia*, 158 (2019) 2599.
26. P. Geng, J. Wang, X. Xu, Y. Zhang and S. Qiu, *International Core Journal of Engineering*, 6 (2020) 171.
27. N. Guo, Y. Fang, Z. Tian and S. Cao, *The Journal of Engineering*, 15 (2019) 576.
28. J. B. Li, M. Ye, K. P. Gao and X. X. Xu, *Iet Power Electronics*, 14 (2021) 2249.
29. P. Nian, S. Z. Zhang and X. W. Zhang, *Journal of Energy Storage*, 40 (2021) 102559.
30. M. Li, Y. J. Zhang, Z. L. Hu, Y. Zhang and J. Zhang, *Sensors*, 21 (2021) 151050.
31. X. Li, L. Qiu, L. Chang and Y. Wang, *Computer Measurement & Control*, 27 (2019) 208.
32. J. Feng, L. Wu, K. Huang, J. Lu and X. Zhang, *Energy Storage Science and Technology*, 10 (2021) 242.
33. X. Kong, W. Zhao and C. Wang, *Automotive Engineering*, 39 (2017) 648.
34. N. Feng, Z. Wang and Y. Pang, *Chinese Journal of Power Sources*, 35 (2011) 1586