Bulgarian Chemical Communications, Volume 47, Number 3 (pp. 904 – 912) 2015

Adaptive mutation particle swarm optimized BP neural network in state-of-charge estimation of Li-ion battery for electric vehicles

Feng Jin[1,2], He Yong-ling[1]

[1] School of Transportation Science and Engineering of Beijing University of Aeronautics and Astronautics, Beijing 100191;

[2] Department of Automobile Engineering, Guilin University of Aerospace Technology, Guilin 541004

Submitted May 12, 2014

The state of charge (SOC) of Li-ion battery on electric vehicle (EV) is highly nonlinear. The randomly selected initial parameters of BP neural network can cause significant inaccuracy and long training time. In the study presented in this paper, an optimized BP neural network, with its initial parameters optimized by adaptive particle swarm optimization (PSO) algorithm, was used to estimate the Li-ion battery's state of charge (SOC). The performance on BP neural network estimation, as well as the optimized performance with adaptive mutation PSO was analyzed. A model for adaptive mutation PSO- BP neural network was established for battery SOC estimation. Experimental results show that: using BP neural network optimized by adaptive mutation PSO for SOC estimation of Li-ion battery of EV, can overcome the shortcomings of easily trapped to local optimum, long training time and so on. It also reduces the estimation deviation.

Key words: Adaptive mutation; particle swarm optimization; BP neural network; state of charge

INTRODUCTION

State of charge (SOC) estimation is a key technology of battery management system (BMS) of electric vehicles (EV). Accurate estimation of SOC not only avoids the danger of over-charging and over-discharging which may damage the battery, but also helps creating a better battery control strategy, which allows more effective control and more precise prediction of driving range. This will help approaching the goal of energy saving, environmental protection, and battery life extension for EV [1, 2].

SOC of battery cannot be measured directly. Instead, it can be estimated from some physical properties of the battery, such as terminal voltage, current, temperature, etc. The accuracy of estimation is affected by many factors, i.e. voltage, charge-discharge rate, power, temperature, life cycle, internal resistance, internal pressure, self-discharge rate, etc. These factors have strong nonlinear relationship with SOC. Therefore it is difficult to establish an accurate mathematical model [3, 4].

The commonly used methods for SOC estimation include Ah counting method, open circuit voltage (OCV) method, the linear model method; neural network method and Kalman filter (KF) method [2]. Ah counting method can get the battery charge and discharge electricity by the integral of current times the time. If the initial SOC is known, this method can be approached on-line SOC testing. But the algorithm also has some drawbacks such as, the Coulomb efficiency is difficult to be measured accurately and the accumulated sampling error is large, and so forth. It is not suitable for the occasions where the voltage and current change dramatically. Therefore, Ah counting method does not meet the requirement of EV for long-term use [5]. Some studies proposed improved Ah counting method that an equivalent Coulomb efficiency was defined to alleviate these problem with an SOC estimation method combined with the open circuit voltage method, Kalman filter, and Ah counting method. The SOC estimate error using this method relative to a discharge test was only 2.3%, satisfies the 8% SOC estimate precision requirement of EV. However, this method also comprises the problem of high requirement of model accuracy, large amount of calculation, high requirement of hardware, and couldn't meet the requirements for commercialization [6]. The most obvious drawback of OCV method is that battery must be relaxed for a long time before each measurement to eliminate the battery polarization effects which affects accuracy of voltage value. So this method is not suitable for online estimation of battery' SOC. The most effective use of OCV method is in initial SOC estimation after EV's long time standing so it is often used in combination with the Ah counting method. A recent study [7] proposed an equivalent circuit network to describe the polarization effect of the battery in OCV method. The recursive least square algorithm with

* To whom all correspondence should be sent:
E-mail: daewoo_feng@126.com

forgetting was applied to implement the on-line parameter calibration. The maximum and mean relative errors are 1.666% and 0.01% respectively, in a hybrid pulse test. The linear model method which is based on linear equation established by the relationship between current, voltage, SOC variation and the former SOC value, is suitable for the low-current situation, which makes it only applicable to the lead-acid battery. The KF method is a useful tool for optimal state estimation of systems. Using the temporal transfer relationship of a system, this approach estimates the state of system with a set of recursive formula. It is suitable for noise filter in the harsh environment like EV driving process. However, the KF needs a proper equivalent circuit model of the battery to describe the characteristics of charging and/or discharging, of which the internal parameters are often difficult to determine. Meanwhile, for large amount of calculation, the system requires a higher speed processor which means higher cost [8].

The neural network method presumes a highly non-linear system, applicable to SOC estimation of all kinds of battery. But, it needs a large number of experiment data for training [9, 10]. Since the initial weights and thresholds of neural network are selected randomly, each training result of the network is different and the range of deficiency is large. Finding the proper network parameters takes a great amount of time. However, using power battery testing equipment, training samples all-inclusive for covering the entire work range can be collected and use to train the neural network. On this basis, the forecast accuracy of neural network can be improved as long as the proper network can be constructed and the initial weights and thresholds could be optimized [11].

In this paper, the BP neural network optimized with adaptive mutation PSO would be proposed for estimation of battery SOC. First, the characteristics of BP (Back Propagation) neural network and the modeling process will be introduced. Then, the initial weights and thresholds of BP neural network optimized by adaptive mutation PSO will be taken into the BP neural network to establish the SOC estimator. Finally, the proposed method will be tested in UDDS cycles and the simulation results are compared with the actual values.

ESTABLISHMENT OF BP NEURAL NETWORK

BP Neural Network

BP network is a multi-layer forward network with hidden layer and error feedback. It has good learning and adaptive capacity as to solve the learning problem of the connection weights of implied unit in a multi-layer network. As of today it is the most widely used neural network [12]. The basic principle of BP neural network algorithm is the gradient steepest descent method, which can minimize the total errors by adjusting the network weights. That is, the gradient search technology minimizes the error of the mean square value of actual output. In fact, multi-layer network using BP learning algorithm contains the forward and reverse spread of two stages. The input information from the input layer is propagated through the hidden layer to output layer and processed layer-by-layer during the forward propagation process [11,13]. The structure of BP neural network is showed as Fig.1.

The input of the *i-th* neuron in hidden layer is showed as below under the sample *p*:

$$o_i = f(\sum_{j=1}^{m} w_{ij} x_j - \theta_i) \tag{1}$$

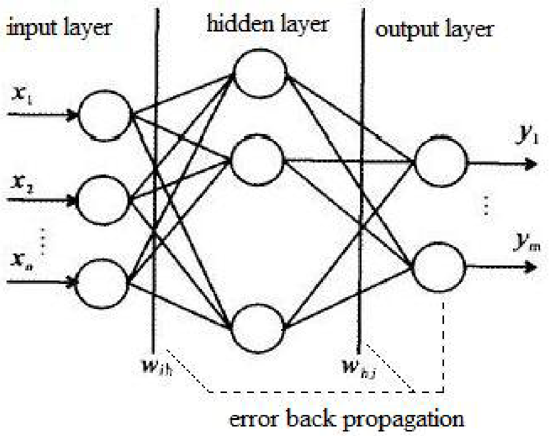

Fig.1. The structure of BP neural network.

Where o_i is the output value of the *i-th* neuron in the hidden layer, w_{ij} is the connection weight between the *j-th* neuron in input layer and the *i-th* neuron in the hidden layer, x_i is the input value of the *j-th* neuron in input layer, θ_i is the threshold of the *j-th* neuron of input layer, *f* is the activation function of the hidden layer.

The output of the *k-th* neuron in the output layer is:

$$y_k = g(\sum_{i=1}^{q} w_{ki} o_i - \theta_k) \tag{2}$$

Where, y_k is the output value of the *k-th* neuron in output layer, w_{ki} is the connection weight between the *k-th* neuron in output layer and the *i-th* neuron in hidden layer, θ_k is the threshold of the *k-th* neuron of output layer, *g* is the activation function of the output layer.

If the output value is not the expected, back propagation then begins. The error signal is returned along the original connection channel, then, the value of connection weights of each layers are modified to make the error signal less at the same time.

The error function *J* can be expressed as:

$$J=\frac{1}{2}\sum_{k=1}^{L}(t_k-o_k)^2 \tag{3}$$

Where, *L* is the number of output layer, t_k is the target value.

Weight coefficient of output layer adjustment

The weight coefficient is adjusted according to the opposite direction of function gradient, which make the network gradually converge. According to the gradient method, the correction formula of each neuron weight coefficient of output layer is as follows:

$$\Delta w_{ki}=-\eta\frac{\partial J}{\partial w_{ki}}=\eta o_k(1-o_k)(t_k-o_k)o_i \tag{4}$$

Similarly, the correction formula of the each neuron weights coefficient of hidden layer is as follow:

$$\Delta w_{ij}=-\eta\frac{\partial J}{\partial w_{ij}}=\eta o_i(1-o_i)(\sum_{k=1}^{L}((t_k-o_k)o_k(1-o_k)w_{ki}))o_j \tag{5}$$

Collection of testing sample

When using BP neural network algorithm for SOC estimation of Li-ion battery, the first thing is the collection of training samples and testing samples. The number of training samples should be large and all-inclusive for covering the entire work range. There are many impact factors in SOC estimation. Considering the purpose of this research is to verify the rationality of the algorithm, only current and voltage’s influence are taken into account [14].

Simulation software ADVISOR (Advanced Vehicle Simulator) is developed by National Renewable Energy Laboratory (NREL) of the United States for the management of the development of hybrid drive systems. Because the battery data in this software is from experiment done by NREL, its data is relatively accurate and comprehensive. In this paper, experiment samples would be acquired under different working conditions using a virtual EV which contains a 6Ah Li-ion battery manufactured by SAFT Company of the Unite State. The working conditions include constant speed of 8 km h^{-1}, constant speed of 72kmh^{-1}, constant speed of 144 km h^{-1}, FTP cycle and UDDS cycle. The relationship curve between speed and time is shown as Fig. 2 in the example of UDDS cycle.

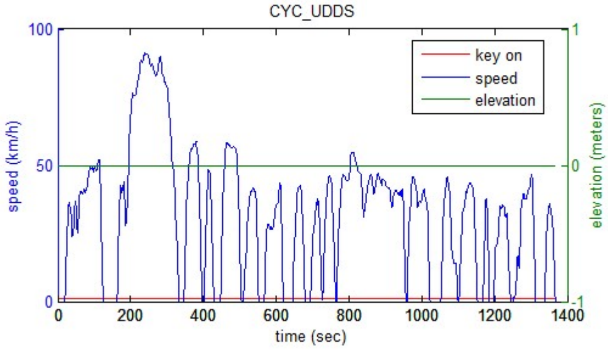

Fig. 2. EPA Urban Dynamometer Driving Schedule (UDDS)

The above working condition simulates the situations of EV at low speed, medium speed, high speed and urban road, covering the entire typical driving pattern and has strong representation. In order to collect sufficient data for the neural network training and testing, each of simulation working condition was looped to execute from SOC being 1 until SOC being 0. Each parameter of EV battery including voltage value and current value was sampled at the frequency of 1 during the simulation process and there were total of 12925 sets of data. The 2386 sets of data sampled during UDDS cycle were for testing, the other data were grouped as 160-set samples according to the principles of uniform distribution for training [15, 16]. Fig. 3 to Fig.5 shows the sample value when virtual EV drives during one period of UDDS cycle.

Sample preprocessing

From above figure we can see that current and voltage samples have difference in the order of magnitude. In order to avoid this problem which would make the network error larger, samples should be normalized first. Meanwhile, samples normalization can also help the convergence of network’s training speed accelerate. The common samples normalization methods includes maximum and minimum method and average variance method. For the reason to simplify the problem, maximum and minimum method was employed in this paper.

$$x_k=(x_k^*-x_{\min})/(x_{\max}-x_{\min}) \tag{6}$$

Where, x_k is the *k-th* sample factor after normalization, x_k* is the *k-th* sample factor before normalization, x_{max} is the maximum value of sample factor before normalization and x_{min} is the minimum value of sample factor before normalization.

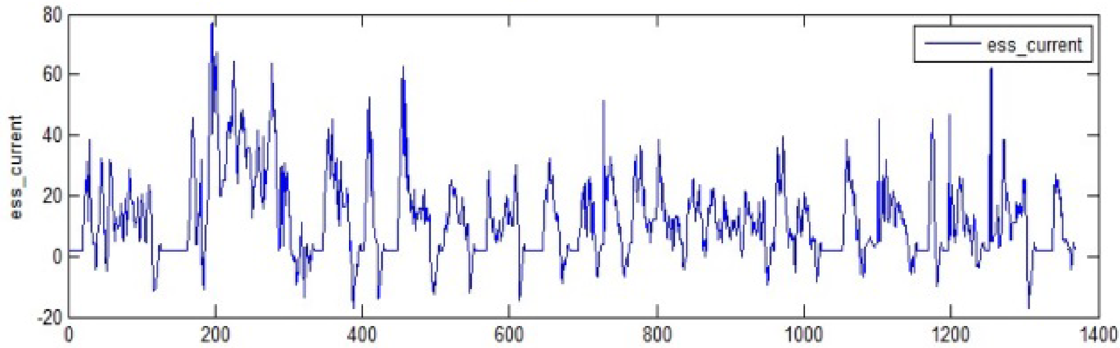

Fig. 3. Current profile sampled during UDDS cycles.

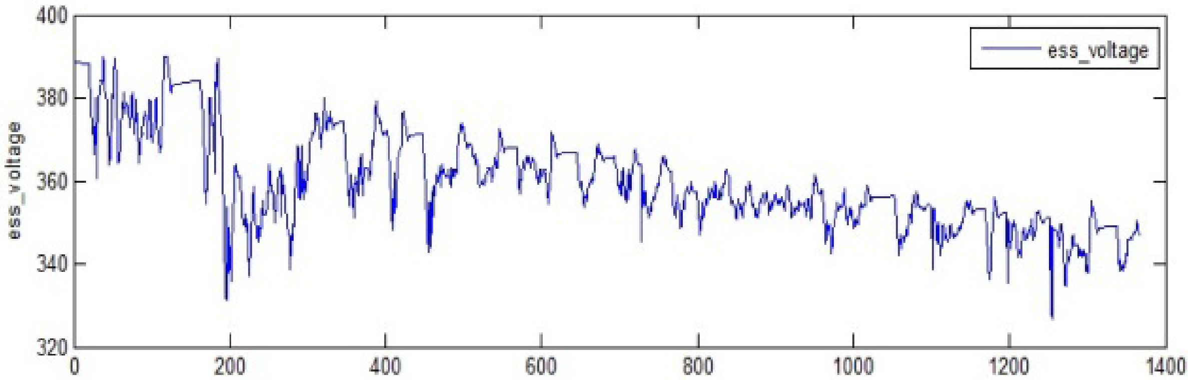

Fig. 4. Voltage profile sampled during UDDS cycles.

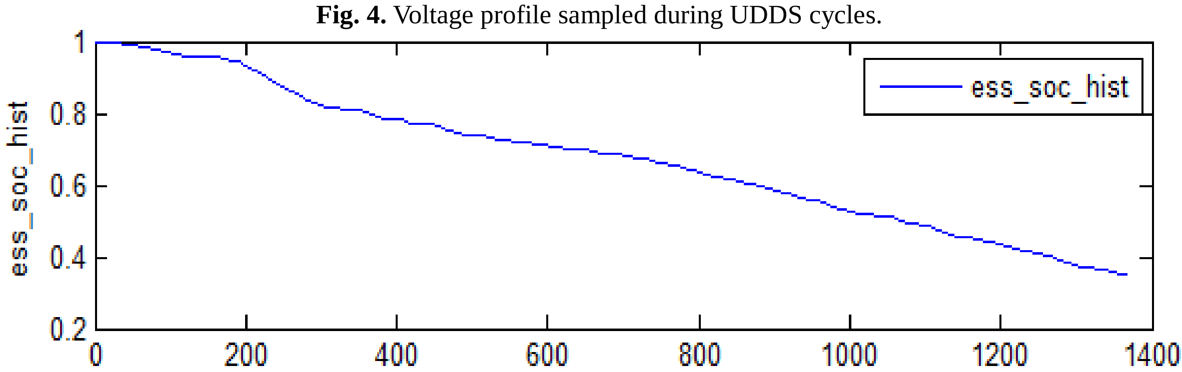

Fig. 5. SOC profile sampled during UDDS cycles.

The hidden layer of BP neural network

The number of nodes in hidden layer of BP neural network has a great impact on network's prediction accuracy. If the nodes number is too small, the network does not have sufficient learning and needs to increase the frequency of training, so the training accuracy may be less than desired. However, if the number of nodes is too much, it will make the training time too long and the network easy to over-fitting. The number of nodes has a direct relationship with the requirements, input and output nodes of the problem. The following two equations can be used as reference formula to select the optimum node number of hidden layer [17].

$$L=\sqrt{(m+n)}+a \tag{7}$$

Where L is the number of nodes of hidden layer, m is the number of node of output layer, n is the number of nodes of input layer, a is a constant between 0 and 10.

$$L=\log_2 n \tag{8}$$

Where n is number of nodes of input layer.

Base on the above conditions, the number of nodes of hidden layer should be determined through thorough testing. In this study, number of nodes of the hidden layer are 5. The established BP neural network is shown as Fig. 6.

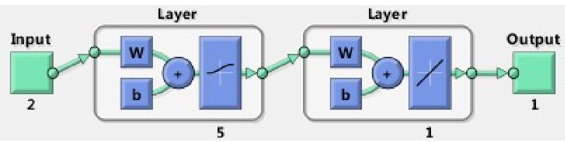

Fig. 6. The BP neural network.

Weight and threshold of bp neural network optimized by PSO

Since the initial weights and thresholds of BP neural network are selected randomly, each training result of the network is different and range of deficient is large. Finding the proper network parameters takes a great amount of time. If the initial weights and thresholds of BP neural network are optimized by particle swarm optimization algorithm, the forecast accuracy of network can be improved.

Particle swarm optimization (PSO) is a new evolutionary algorithm developed in recent years. Similar to genetic algorithm (GA), PSO find the optimal solution by iteration, starting at a random solution. However it does this in a simpler way. PSO find the global optimum by following the current optimal value without “crossover” or “mutation” like GA. Compared with generic algorithm, PSO is easier to implement, has enhanced global searching capability, higher precision and faster convergence. Using PSO to optimize the initial weights and thresholds of BP network can shorten the network training time, improve the convergence, enhance network generalization ability and reduce error [18, 19].

The algorithm assumes that there are a number of particles in a population, and each particle has a position vector and velocity vector. The position vector and velocity vector of the *i-th* particle can be expressed as:

$$X_i = [x_{i1}, x_{i2}, \ldots, x_{id}] \tag{9}$$

$$V_i = [v_{i1}, v_{i2}, \ldots, v_{id}] \tag{10}$$

Where d represents the dimension of the solution space and its value is also the possible solutions. Particles can find the optimal solution by iteration through constantly moving in search space. The basic formula is:

$$v_{ij}^{k+1} = \omega v_{id}^{k} + c_1 r_1 (p_{ij}^{k} - x_{ij}^{k}) + c_2 r_2 (p_{gj}^{k} - x_{ij}^{k}) \tag{11}$$

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1} \tag{12}$$

Where i = 1,2,…,N and j=1,2,…,d, v_{ij} are flight speed of the *i-th* particle, x_{ij} is position of the *i-th* particle, ω is the inertia factor, c_1 and c_2 are acceleration factor which is positive constant, r_1, r_2 are the random number on the interval ***[0,1]***, p_{ij} is the best position of the *i-th* particle currently find. p_{gj} is the best position of global population currently find.

The initial positions and velocities of the particle swarm are generated randomly, and then they iterate according to the formula 11 and formula 12. Particles continue to modify their velocities and positions according to p_{ij} and p_{gj} in each of the iterations, so that particles approach to the global optimal solution. There are 21 (2*5+5+5*1+1) parameters that need to be optimized for a BP neural network with topology structure of ***[2,5,1]***.

Adaptive mutation algorithm establishment

PSO algorithm iteratively update by tracking the most optimal particle. Once a particle finds the optimal value, the other particles will quickly move close to it. However, the traditional algorithms have the problems of early convergence when the optimal value is trapped to local optima. It is necessary to enhance the basic PSO algorithm to avoid the premature convergence problem. The reason of premature convergence is large lost in population diversity. When algorithm escapes from local optima before convergence, it can continue searching in other area in solution space and finally find the global optima [20].

In this study, a random number in the iterative formula serves as mutation condition. Once the particles are greater than iteration threshold value, it mutates to be a random number. In this way, some of the particles can maintain the diversity for optimization from the current optimal conditions. The optimized algorithm with adaptive mutation can be established as:

$$x_{ij}^{k+1} = \begin{cases} x_{ij}^{k} + v_{ij}^{k+1} & r_1 \le m \\ rand & r_1 > m \end{cases} \tag{13}$$

Where, m is the mutation threshold.

The basic parameters of adaptive mutation PSO algorithm are set as the following: The population size is 20, the maximum number of iterations is 200, learning factor $c1 = c2 = 1.49$, Speed range is on interval ***[-1,1]***, the position range is on interval ***[-1,1]***, the adaptive mutation threshold m = 0.8. First, fitness value is compared between PSO with adaptive mutation and the basic PSO. Fig.7 shows the fitness curve without adaptive mutation and Fig.8 shows the fitness curve with adaptive mutation.

As can be seen in Fig.8, although the optimization process of PSO algorithm without adaptive mutation is obvious, it converged quickly at first, but stopped at the 120 generation and trapped in local optima. In contrast, the PSO algorithm with adaptive mutation continues finding the optimal solution during the whole evolution process as shown in Fig.9.

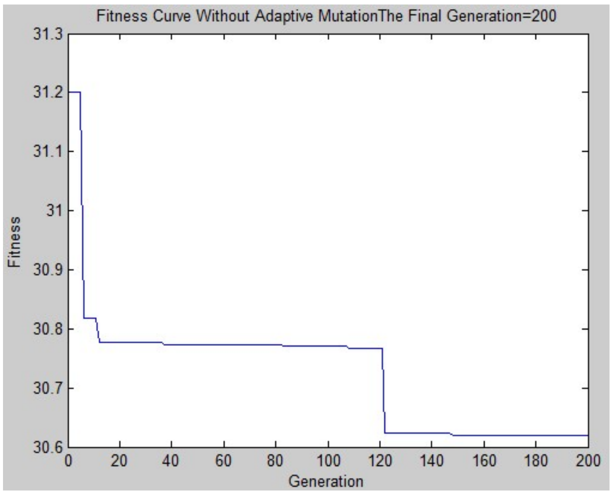

Fig. 7. Fitness curve without adaptive mutation.

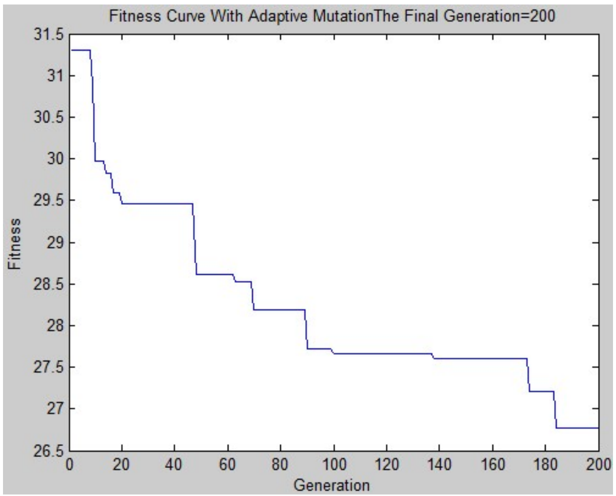

Fig. 8. Fitness curve with adaptive mutation.

EXPERIMENT

Experiment process

To verify the effect of BP neural network whose initial weights and threshold are optimized by adaptive mutation PSO, experiment and simulation were conducted. Experiments were conducted under UDDS conditions to illustrate the algorithm. Before starting the test, the Li-ion battery was fully charged (SOC=100%). Current and voltage value sampled from UDDS cycle in Advisor were imported into BP neural network for simulation. And then, the simulation result was compared with the experiment result done by NREL.

UDDS stands for Urban Dynamometer Driving Schedule. It refers to a United States Environmental Protection Agency (EPA) mandated dynamometer test on fuel economy that represents city driving conditions, which is used for light duty vehicle testing. Each cycle time is 1369 seconds, 7.45 miles, with average speed of 31.52 kmh^{-1}. Conditions cycle was shown as Fig.2. In this paper, several cycles of UDDS were employed to verify the SOC estimation algorithm. The voltage and current profiles sampled during UDDS cycles were shown in Fig.3 and Fig.4.

From above figure we can see, the battery was in a rapidly changing dynamic process under the UDDS cycle. The current and voltage change very quickly. Simulation under this working cycle can test the generalization ability of BP network well.

Experimental result

To verify the performance of SOC estimation by adaptive mutation PSO-BP neural network for Li-ion battery of EV, we compared with standard BP neural network. Two kinds of model were trained with uniform training samples and set with uniform parameters, of which learning rate *lr* was 0.05, inertia factor *mc* was 0.9, number of iterations was 5000, error target was 10E-5. Meanwhile, relative errors between estimation value and experiment value were compared to illustrate their magnitude of error. Relative error was defined as follows:

$$Error = \frac{soc_s - soc_t}{soc_t} \tag{14}$$

Where, *Error* is the relative error, soc_s is the estimation value of SOC and soc_t is experiment value of SOC.

Experimental and simulation results were shown in Fig.9-12. Fig.9 and Fig.10 showed the actual and the estimated SOC during the entire charging process. Fig.11 and Fig.12 showed the relative error between the actual SOC and the estimated SOC.

From the estimation curve and error curve, estimated SOC by adaptive mutation PSO-BP algorithm matched the actual SOC well. It could follow the actual value trend. The relative error was small (about 8%) in the range of SOC from 1 to 0.15. However, the relative error became larger as the SOC decreases below 0.15. Considering the fact that SOC of EV's power battery is in the range from 0.2 to 0.8, the error was acceptable. Estimated SOC by standard BP algorithm can also follow the trend of experiment value. However, the relative error was large, more than 10% in the range of SOC from 1 to 0.15. Its estimation precision was lower than adaptive mutation PSO-BP algorithm. In summary, using adaptive mutation PSO-BP neural network has better precision in SOC estimation of EV's power battery than standard BP algorithm.

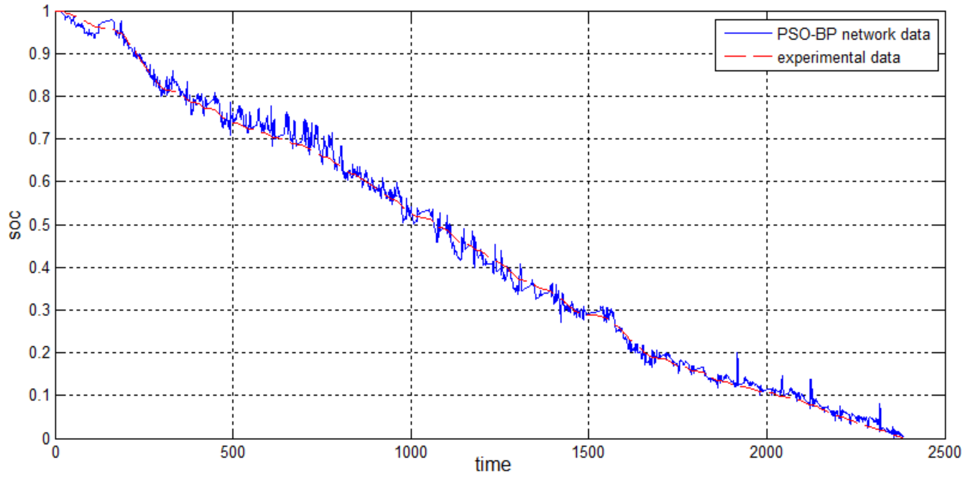

Fig. 9. SOC curves with adaptive mutation PSO-BP and experiment.

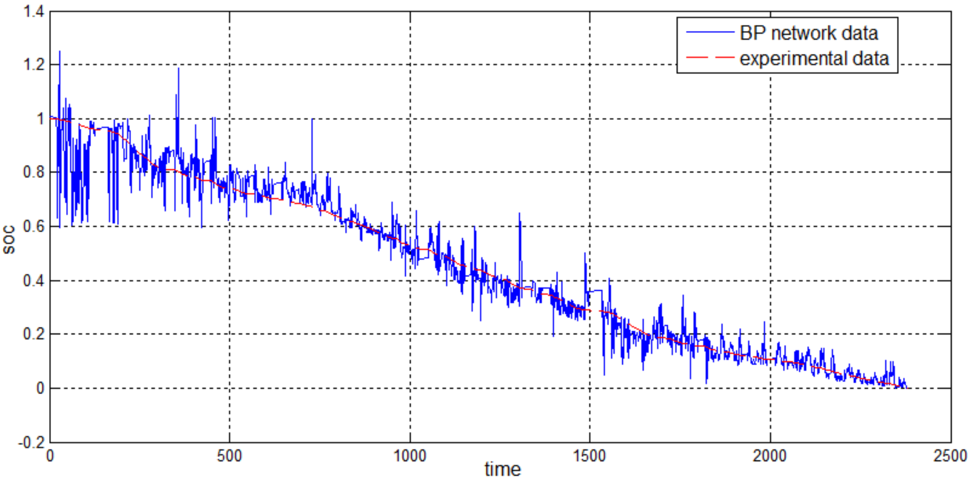

Fig. 10. SOC curves with BP network and experiment.

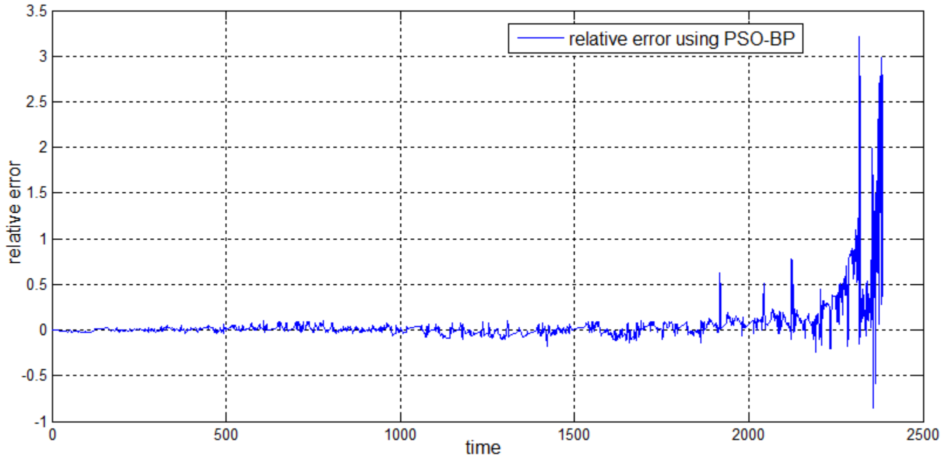

Fig. 11. Relative error between adaptive mutation PSO-BP and experiment.

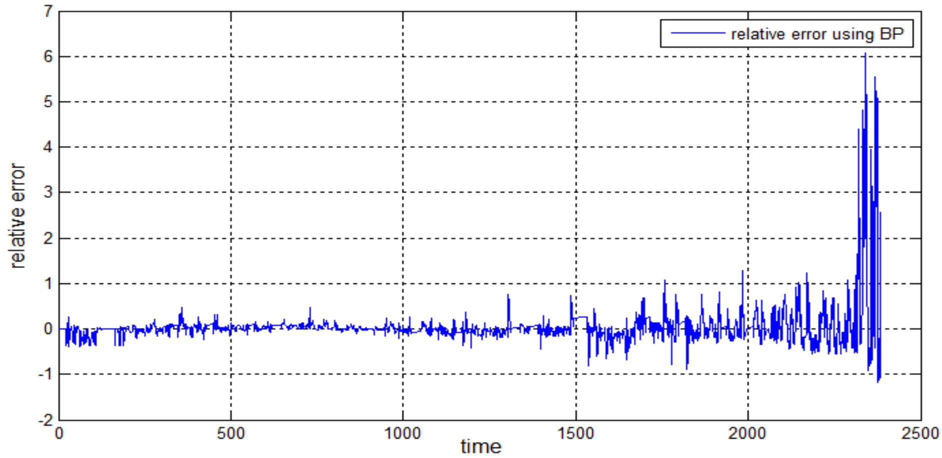

Fig. 12. Relative error between BP network and experiment.

CONCLUSION

Battery SOC estimation is one of the most important tasks in the EV's BMS. Not only is it the basic parameter to decide the vehicle control strategy, but also it helps drivers using battery power reasonably as to control and predict the driving range. In this paper, EV's power battery SOC estimation algorithm is proposed based on BP neural network whose initial parameters optimized by adaptive mutation PSO. Finally, the experiment demonstrated the basic performance of the algorithm. The results are as follows:

(1) The convergence rate of adaptive PSO-BP neural network is not only faster than BP neural network but also has strong ability of global optimization.

(2) Using BP neural network for EV's power battery SOC estimation is feasible. Furthermore, the algorithm with its initial parameters optimized by adaptive mutation PSO has better performance than basic BP neural network and has higher accuracy in the SOC estimation of EV's power battery. So it has application value.

Acknowledgments. *The authors gratefully acknowledge the support of the Beijing University of Aeronautics and Astronautics and the Guilin University of Aerospace Technology. The work is supported by the project of the Guilin University of Aerospace Technology under Grant No. X12Z002.*

REFERENCES

1. Feng Jin, He Yong-ling. *Advanced Material Research*, **490-495**, 3854 (2012).
2. Feng Jin, He Yong-ling, *J. Theor. Appl. Inform. Technol.*,**48**, 1398 (2013).
3. Lin Chen-tao, Wang Jun-ping, CHEN Quan-shi, *Battery Bimonthly*, **134**, 376, (2004) (in Chinese).
4. Yinjiao Xing, Wei He, M. Pecht, Kwok Leung Tsui, *Appl. Energy*, **113**,106 (2014).
5. Wu Hong-jie, Qi Bo-jin, Zheng Min-xin, Liu Yong-zhe, *J. Beijing Univ. Aeronautics Astronautics*, **33,** 945 (2007) (in Chinese).
6. Lin Cheng-tao, Chen Quan-shi, Wang Jun-ping, HuanG Wen-hua, Wang Yan-chao, *J. Tsinghua Univ. (Sci & Tech)*, **46,** 247 (2006) (in Chinese).
7. Hu Xiao-song, Sun Feng-chun, ZoU Yuan, *J. Central South Univ. Technol.*, **18**, 1525 (2011).
8. Xia Chao-ying, Zhang Shu, Sun Hong-tao. *Chinese J. Power Sources*, **31,** 414 (2007) (in Chinese).
9. M. Charkhgard, M. Farrokhi, *IEEE Trans. Industr. Electronics*, **57**, 4178 (2010).
10. Jaemoon Lee, Oanyong Nam, B.H. Cho, *J. Power Sources*, **174**, 9 (2007).
11. V. Valdez, A. Jojutla, IEEE Neural networks conference, **39**, 193 (2006).
12. G. Das, P.K. Pattnaik, S.K. Pattnaik, S.K. Padhy, *Expert Systems with Applications*, **41**, 3491 (2014).
13. Yu Chang-guan. Modern control theory and applications, second ed., Harbin Institute of Technology, Press. Harbin, 2007 (in Chinese).
14. Liu Rui-hao, Sun Yu-kun, Chen Kun-hua, *Electrical Measurement&Instrumentation*, **48**, 34 (2011) (in Chinese).
15. Lei Xiao, Chan Qing-quan, Liu Kaipei, Ma Li, *Trans. China Electrotech. Soc.*, **23**, 81 (2008) (in Chinese)
16. G. L. Plett, *J. Power Sources*, **134,** 277 (2004).
17. Liu Qiu-li, Ma Xiao-jun, Yuan Dong, Su Jian-qiang, *Computer Eng*., **38**, 143 (2012) (in Chinese).
18. M. Carvalho, T.B. Ludermir, Sixth International conference on hybrid intelligent systems, **6**, 2 (2006).
19. Gan Xu-sheng, Duanmu Jing-shun, Meng Yue-bo, CONG Wei, *J. Central South Univ. Technol.*, **20**, 1592 (2013).
20. A. Alfi, *Acta Automatica Sinica*, **37**, 541 (2011).

ОЦЕНКА НА ЗАРЕЖДАНЕТО НА ЛИТИЕВО-ЙОННИ БАТЕРИИ С ПОМОЩТА НА АДАПТИВНА МУТАЦИЯ И ОПТИМИЗАЦИОНЕН АЛГОРИТЪМ С РОЯК НА ЧАСТИЦИ ПРИ НЕВРОННИ МРЕЖИ С ОБРАТНО РАЗПРОСТРАНЕНИЕ

Фенг Джин[1,2]，Хе Йонг-линг[1]

[1] *Училище по транспорт и инженерство при Университета по аеронавтика и астронавтика в Бейджин, Бейджин 100191, Китай*

[2] *Департамент по автомобилно инженерство, Университет по космически технологии, Гуйлин 541004, Китай*

Постъпила на 12 май, 2014 г.

(Резюме)

Състоянието на зареждане (SOC) на литиево-йонните батерии за елктромобилите (EV) е силно нелинейно. Произволният избор на началните параметри на невронните мрежи с обратно разпространение (BP) може да причини значителна неточност и дълго време за трениране. В настоящето изследване се въвежда BP оптимизирана невронна мрежа с начални параметри, оптимизирани чрез алгоритъм, основаващ се рояк на частици (PSO) с цел оценяване на състоянието на зареждане на литиево-йонна батерия (SOC). Анализирани са поведението на BP-невронната мрежа, както и оптимизираното поведение с адаптивна мутация. Съставен е модел на адаптивна мутация PSO-BP невронна мрежа, описващ състоянието на зареждане на батерията SOC. Експерименталните резултати показват, че чрез използването на BP-невронната мрежа, оптимизирана чрез адаптивна мутация PSO за оценка на SOC на литиево-йонните батерии за електромобили се преодоляват недостатъците от попадане на целевата функция в локален минимум, дълги времена на трениране и пр.