Asian Journal of Chemistry; Vol. 26, No. 11 (2014), 3339-3343

ASIAN JOURNAL OF CHEMISTRY

http://dx.doi.org/10.14233/ajchem.2014.17522

Multi-Fault Diagnosis of Aluminum Electrolysis Based on Modular Fuzzy Neural Networks

JIEJIA LI*, PENG ZHOU and JINXIANG PIAN

School of Information and Control Engineering, Shenyang Jianzhu University, Shenyang 110168, P.R. China

*Corresponding author: E-mail: ljj_0123@sjzu.edu.cn

Received: 12 February 2014; *Accepted*: 15 April 2014; *Published online*: 25 May 2014; AJC-15244

Aluminum electrolysis is a nonlinear process with the characteristics of multi-variable, strong coupling, time-varying and large time-delay. There are different types of faults that occur frequently in it. According to the fault characteristics of the aluminum electrolysis, a multi-fault diagnosis method of aluminum electrolysis which is based on modular integrated fuzzy neural network is proposed. Considering the shortages of a single network applied in multi-fault diagnosis, a multi-fault diagnosis platform with two layers of sub-network and decision fusion network is constructed in multi-fault diagnosis of aluminum electrolysis, combining fuzzy logic and neural network by the application of the concept of modular integration. Mixed particle swarm optimization algorithm is adopted in the paper so that the convergence speed and accuracy of the network can be increased to some extent. Simulation results show that the proposed method can improve the accuracy rate of fault prediction and give the prediction advance.

Keywords: Aluminum electrolysis, Modular, Fault diagnosis, Fuzzy neural network, Integrated neural network.

INTRODUCTION

The aluminum electrolysis process is with complicated electrolysis equipment, harsh operating conditions during and various kinds of faults. Sometimes, multi-failures happen at the same time. So, an effective fault diagnosis is the key point to ensure the safety and stability. The advantages of adopting the single neural network to realize the multi-fault diagnosis are as follows:

• Quite a few fault samples are required in the network and it is hard to obtain the eigenvectors that refer to the various faults.

• The number of the input knots is huge in the vast fault-diagnosis network which contributes to the difficulty in training. By contrast, the accuracy of the fault alarm will be decreased for the less fault information if the number of inputs is cut down.

• As the magnitudes of the input information differ sharply, the data with small magnitudes will be drowned in the vast magnitudes of noise and the fault information can not be reflected. So the effect of the fault diagnosis will certainly be influenced.

Given the above drawbacks in the single neural network, a new method called modular integrated fuzzy neural network of multi-fault diagnosis solution method is put forward.

EXPERIMENTAL

Setup of the network model: We adopt the modular integrated fuzzy neural-network multi-fault diagnosis model (including diagnosis of sub-networks anode effect hot tank and cold tank diagnostic sub-networks and the data-fusion sub-networks) based on the modulation and data fusion. As is shown in Fig. 1[1].

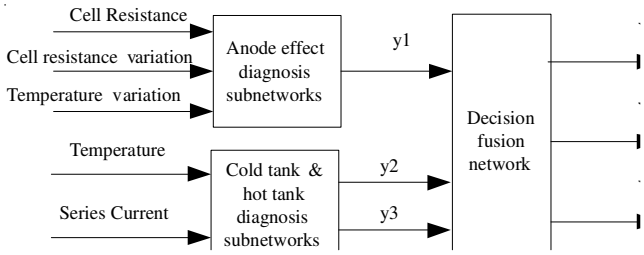

Fig. 1. Structure of aluminum electrolysis fault diagnosis network

While designing the fault diagnosis sub-network, the closely related characteristic variables are chosen as the input variables of the sub-network, cell resistance, cell resistance change rate and the change rate of the temperature. Temperature and the current series[1] are chosen in hot tank and cold tank fault diagnosis sub-network.

The fault probability is output by the each fault diagnosis sub-network and its initial diagnosis of fault is finished. In order to improve the accuracy in fault diagnosis and to detect at least two faults at the same time, two diagnosis modes are utilized and these modes add up to the decision fusion web, analyzing the fault information and giving their conclusions on the fault diagnosis.

Sub-fault diagnosis model: Elman neural-network structure is adopted in the sub-fault diagnosis model, anode effect sub-diagnostic network composed of hot sink and cold sink. Set the anode effect diagnostic fault sub-network for example, the Elman neural-network is used in this network, which is mainly made up of the input layer, hidden layer, structure layer and the output layer. Its structure model is given below:

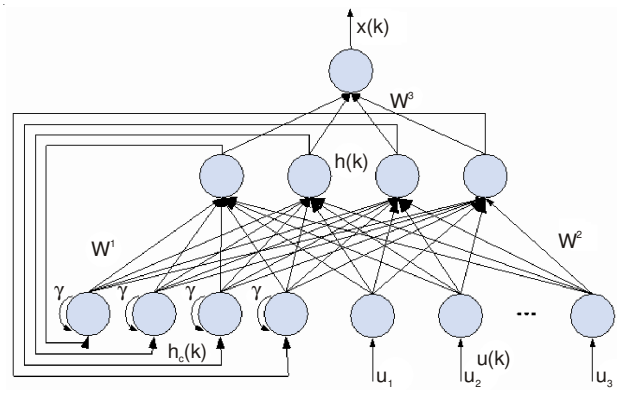

Fig. 2. Anode effect sub-fault diagnostic network model

The role of input layer is to allocate the input signals of the system. Its quantity is determined by the mechanism of the fault diagnosis. The inputs of the sub-diagnostic network for anode effect are the sink resistance, the rate of change for the resistance and the change rate for the temperature (three input variables). Similarly, the inputs of the fault diagnostic sub-network are the currents and temperatures, which means that there are two inputs in it.

$$y_i^u(k) = u_i(k) \tag{1}$$

y_i^u is the ith component in the input layer; $u_i(k)$ is the i^{th} component of the k^{th} output.

The role of the output layer is to give the sub-network fault diagnosis probability and these parameters are passed to the decision-fusion network to be given fused.

$$x(k) = W^3h(k) \tag{2}$$

Among them W^3 is the weight from the hidden layer to the output layer.

Recursive algorithm is used in the structure layer to derive the output of the last time in the hidden memory.

$$h_c(k) = \gamma h_c(k-1) + h(k-1) \tag{3}$$

among them, γ is the self-connected feedback gain factor.

Hidden layer: These neurons are used to extract more valuable information gradually from the input mode, to approach any function, enabling the network to get a better understanding of the complex tasks. The accuracy of the fault diagnosis is largely impacted by the number of the hidden layer. For example, the node optimization is obtained through the genetic algorithm in the anode effect sub-network.

$$h(k) = f[W^1h_c(k) + W^2u(k-1)] \tag{4}$$

W^1 is the weight from the structure layer to the hidden layer. W^2 is the weight from the input layer to the hidden layer. Activation function is the sigmoid function as follows:

$$f(x) = \frac{1}{1+e^{-x}} \tag{5}$$

To determine the number of the hidden layer neurons: The pre-training method to determine network nodes in the hidden neural-network is used. According to the Kolmogorov theorem, the optimization of the nodes in the hidden layer is given below:

$$L \le \sqrt{m+n+a} \tag{6}$$

L is the quantity of the nodes in the hidden layer, m is the quantity of the nodes in the output layer, a is the constant between 0 to 10. In the practical problem, we choose the approximate range of the number of nodes in the hidden layer by referring to the formula to set the best number within the model. Set the anode effect fault diagnosis sub-network as example, input number is 3, output number is 1, which we get from the formula 7 that L < 2 + 10 = 12. To get the better test results, we give L as the upper limit to set up the neural-network and then change the genetic algorithm to optimize it.

Coding: To determine the optimal structure in the Elamn model, the mixed coding way float number code way to code the neural-network weights. Binary symbol string $d_1d_2\ldots d_i\ldots d_n$ are the hidden node codes in the neural-network, among which n is the upper limit of the hidden layer. For example, we set four neurons and the node is effective when $d_i = 1$ and invalid until $d_i = 0$. Set the upper limit in the hidden layer as 4 and when the code is 1011, the first node second one and the third one are chosen. At the same time, the nodes chosen are the effective weights and the thresholds.

Population initialization: From step 1: we know that the number of the hidden layer neurons ranges from 1 to 12. For these 12 different groups of network structures, each produces 5 chromosomes and the population size is 60. The range of the weight initialization is [-1, 1], its crossover probability P_c =0.5, its mutation probability P_m = 0.001 and maximum iteration number G_{max} = 100. Its initialization chromosome structure is

$$w_{1,1}^1 \cdots w_{1,12}^1 w_{2,1}^1 \cdots w_{2,12}^1 \cdots w_{3,12}^1 w_{1,1}^2 \cdots w_{1,12}^2 w_{2,1}^2 \cdots w_{2,12}^2 \cdots w_{12,12}^2 w_1^3 \cdots w_{12}^3$$

The top 12 are structure codes of the effective neurons. And the others are weight codes when the hidden layer neurons are effective.

Calculation of individual fitness: n samples to optimize the network is input. Set y_k and t_k as the real outputs and the desired output, respectively (k = 123…N). Mean square error E_c is expressed as followed:

$$E_c = \frac{1}{N}\sum_{k=1}^{N}(t_k - y_k)^2 \tag{7}$$

According to the training result from the selected prediction model of fitness function evaluation and to simplify the network structure with the same performance in the same conditions, we set the fitness function as

TABLE-1
NUMBER OF HIDDEN LAYER NEURON IN SUB-NETWORK, THE AVERAGE OF ITS FITNESS AND THE NUMBER OF CHROMOSOMES DETECTED BY ANODE EFFECT

Hidden layer nodes	1	2	3	4	5	6	7	8	9	10	11	12
Average fitness	0.113	0.247	0.582	0.613	0.508	0.321	0.245	0.153	0.143	0.217	0.156	0.119
Number of chromosome	1	3	11	18	10	7	3	1	1	3	1	1

$$\text{fit} = \frac{1}{1 + E_c + CO} \tag{8}$$

Among which, the constant $C \in (0, 0.5)$ is the complexity factor in the neural networks. In this paper, we set C = 0.1. O is the number of the nodes in the hidden layer, that is to say, the number of one in the structure coding.

Judging whether the fitness value satisfies the setting requirements. If it meets the requirements, go to step 7, otherwise, go to step 6.

Selection crossover and mutation: Roulette wheel selection method is adopted and several individuals which have larger fitness are copied directly to the next generation. Those that need the crossover and mutation could be operated crossover and mutation according to the crossover probability P_c and the mutation probability P_m to get the next generation and then go to step 4 to evaluate the fitness values.

If it meets the preconditions or it meets the maximum number of the iteration steps, stop the circulating and obtain the optimized chromosome group. Then decode them and convert them into the nodes in the hidden layer. At last, we show the outcome from the hidden layer optimization in the anode effect sub-network as followed in the Table-1.

From the results, as it can be seen that the average fitness and the chromosome numbers are the biggest when the number of the hidden layer neurons is four. Hence, the number of the best hidden layer nodes is 4 in the anode effect predicted sub-network.

Secondary training: The secondary training for the weights of the hidden layer neurons is operated after the number of these neurons forming the fault diagnostic sub-network is determined. Hybrid particle swarm optimization is adopted to optimize the weights in the following chapters.

Decision fusion network design: Fuzzy neural network structure is applied in the decision fusion network. Its inputs are the outputs from the front sub-networks. Anode effect fault diagnosis sub-network has one output node and those form the cold and hot sink has two outputs each. And we get 3 input nodes for the decision fusion sub-network. This paper uses the fuzzy neural network to diagnose the faults in the aluminum electrolysis. The training sample data and the tested sample data are obtained from the production site. During the fault recognition process of the aluminum electrolysis, we find there are the anode effect hot and cold sink faults. Similarly, there exist some compound faults from the anode effect and the hot sink; anode effect and the cold sink. Decision network output modes have six types. Table-2 shows the fault categories from the various outputs.

The fuzzy neural network structure is adopted in the decision fusion network, whose structure is showed in the Fig. 3. It consists of input layer, fuzzy layer, rule layer and the output layer and the allocation, fuzzy work, fuzzy calculation

TABLE-2
OUTPUT VALUE OF THE NETWORK AND ITS CORRESPONDING FAILURE MODE

Fault class	Output		
	Y_3	Y_2	Y_1
Normal system	0	0	0
Anode effect	0	0	1
Cold trough	0	1	0
Hot trough	1	0	0
Anode effect and hot trough	0	1	1
Anode effect andcold trough	1	1	0

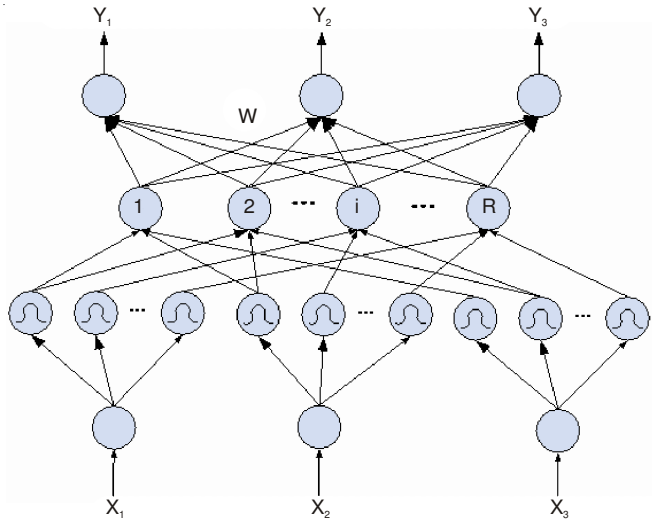

Fig. 3. Decision fusion network model

and the antifuzzy calculation are executed, respectively. From each neuron in the fuzzy layer, its network parameters are the fuzzy center values m, the width and the weight W from the rule layer to the output layer.

The first layer is the input layer, which passes the neuron output directly to the next layer, transition and distribution of signals.

$$y_m^{(1)} = u_m^{(1)} = x_m \tag{9}$$

From which, x_m is the mth component of outer input, which symbolizes the diagnostic outcome from the sub-network.

The second is the fuzzy layer, where the neuron realizes the input fuzzy with the fuzzy functions.

$$y_{mi}^{(2)} = \exp\left(-\frac{\left(u_{mi}^{(2)} - m_{mi}^{(2)}\right)^2}{2\left(\sigma_{mi}^2\right)^2}\right) = \exp\left(-\frac{\left(x_m - m_{mi}^{(2)}\right)^2}{2\left(\sigma_{mi}^2\right)^2}\right) \tag{10}$$

where, $u_{mi}^{(2)}$ is the input and $m_{mi}^{(2)}$ is the output of the ith component in the second layer from the mth neuron in the first layer. $u_{mi}^{(2)}$ is the fuzzy center value and $\sigma_{mi}^{(2)}$ is the width from the neuron. Each network input is divided into five parts in this network, such as negative large, negative small, zero, positive large and positive small. With the Gaussian Functions

the sub-network probability is classified fuzzily according to the range of zero to one.

The third layer is the rule layer to calculate the logic with the fuzzy results of the second layer. The fuzzy rules are shown as follows:

Set x the M-dimensional vector. The number of each term linguistic values of the dimension components are, I_1, I_2 …, I_M. y is the N-dimensional vector. The number of each term linguistic value of dimension components is J_1, J_2, ,…, J_N. The MA model of IF-THEN fuzzy inference rules are shown as follows:

Rule 1: IF x_1 is A_{11}, x_2 is A_{21}, … and x_M is A_{M1}, then y_1 is B_{11}, …, y_N is B_{N1}. Rule r: If x_1 is A_{1r}, x_2 is A_{2r}, … and x_M is A_{Mr}, then y_1 is B_{1r}, …, y_N is B_{Nr}. Rule R: If x_1 is, A_1I_1, x_2 is A_{2l_2}, … and x_M is A_{MI_M}, then y_1 is B_{1J_1}, …, y_N is B_{NJ_N}.

The network computing form is used to attain IF-THEN fuzzy inference rules of the MA model. In the first input layer/ the value of the input language layer we use the M input layer neuron/the value of the input language neurons, which corresponds to the IF-THEN fuzzy rules of the x vector of variables in each component. In the second term layer/the input term layer, I_1, I_2, …, I_M are used, respectively term neuron/ input term neurons, which corresponds to the IF-THEN fuzzy rules of the term language value $A_{11}, A_{12}, …, A_{MI_M}$. In the third layer, rule layer adopts the production operation to realize 'and' in the inference rules of the implementation. Expression is as follows:

$$y_r^3 = \prod_{k=1}^{K_r} u_{r_k}^{(3)} \tag{11}$$

which, K_r is the number of all input variables in the third layer of the r neurons.

The fourth layer is the output layer. The weighted squared method is used to antifuzzy.

$$y_i = \frac{\sum_{r=1}^{R} w_{nr} y_r^3}{\sum_{r=1}^{R} y_r^3} \tag{12}$$

where, y_i is the output of term fuzzy neural network.

Learning algorithms of the network: In the network, the parameters of the neural network are updated with particle optimization algorithm. The network uses the global PSO to find out the approximate results and then use the local PSO to continue the specific search. In the iteration process, we need to get the distance between each particle and other particles. For each particle, use the following formula to get the ratio:

$$\frac{\|X_a - X_b\|}{d_{max}} \tag{13}$$

where, ||Xa - Xb|| is the current distance between particle a and particle b. d_{max} is the maximum distance between any two particles in iterative methods. The threshold è varies according to the number of iterations. When another particle b fits the following formula,

$$\frac{\|X_a - X_b\|}{d_{max}} < \eta \tag{14}$$

Then, the b becomes the neighborhood of this particle. All the particles satisfy the conditions making up a set Ni. Using the improved neighborhood rules, the threshold θ varies according to the number of iterations. θ is expressed as follows,

$$\theta = \frac{3t + 0.6i}{i_{max}} \tag{15}$$

If θ > 0.9, the global particle swarm algorithm is adopted to renew the velocity and location of the particle. Otherwise, the local particle swarm algorithm will be adopted to renew the velocity and location of the particle.

RESULTS AND DISCUSSION

The data aluminum plant provides before and after the fault occurrence is taken to do some fault diagnosis simulation. Take data in 100 min of each state cell in the electrolysis slot (sampling interval is 20 s), a total of 300 inputs are used to train the system aiming for testing the network. Simulation research and analysis are carried on the three parts of the system -the training convergence rate simulation, single fault prediction research simulation and effectiveness of multi-fault prediction simulation.

Results of particle swarm algorithm and its optimization: Repeat the original PSO and improved PSO learning process 10 times, respectively. The statistical comparison results of the learning process are shown in Table-3.

TABLE-3
LEARNING RESULTS COMPARISON BETWEEN ORIGINAL PSO AND IMPROVE PSO

Performance index	Original PSO	Improve PSO
Iteration times	6531.7	3532.6
Single iteration time (ms)	0.591	0.901
Learning time (s)	3.389	3.183

The results show that single-iteration time grows, although the learning coefficient Metropolis in the improved PSO and the PSO classification rules increase the amount of computation. However, this method makes the discovery of PSO global optimum as soon as possible and reduces the number of iterations of global optimization and the total optimization time.

Simulation curve analysis

Comparison of single fault prediction test results: The decision fusion network adopts Gaussian fuzzy membership function. The simulation results are shown in Fig. 4.

Fig. 4 is the anode effect fault diagnosis simulation curve. When the electrolysis cell works normally the cell voltage is generally about 4.2 V. From the simulation curve point of view, before 80 min, the simulation curve dose not change significantly which means the electrolysis is in normal working condition. At 80 min, the cell voltage suddenly increased to 30 V, which indicates the anode effect occurred. At 89 min, cell voltage has back to normal working condition, which means the anode effect is lift. From the point of the parameters change of fault diagnosis model, the cell voltage value is about before 26 min, the fault diagnosis model outputs y_1, y_2 and y_3 are changes between 0 and 0.2, which is relatively stable and indicate the electrolysis cell is in normal working condition;

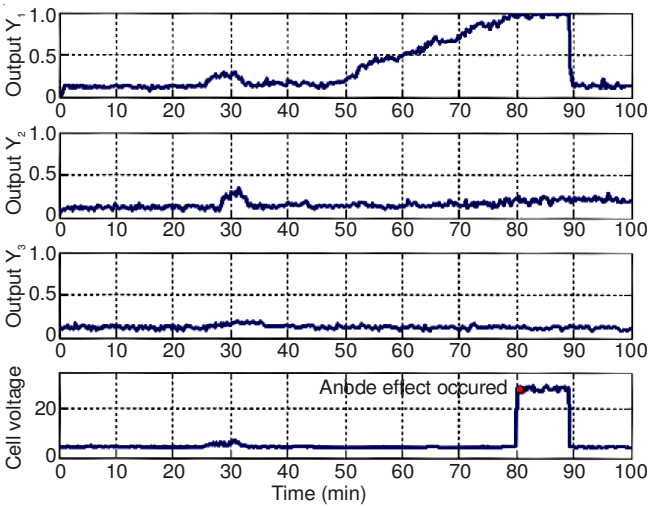

Fig. 4. Simulation curve of anode effect fault diagnosis

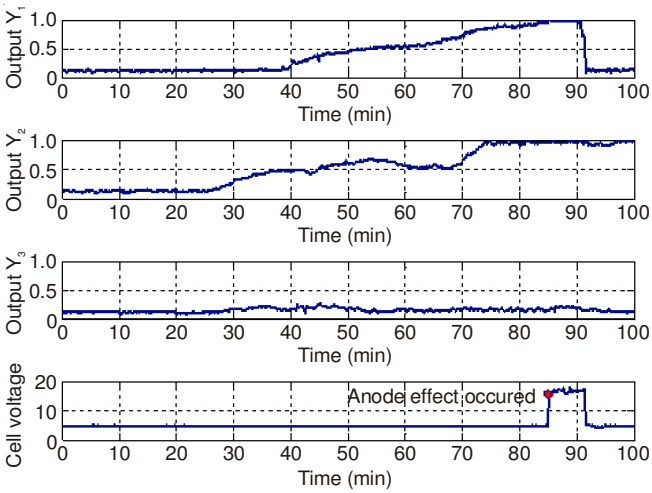

Fig. 5. Simulation diagram of complex fault

During 26-33 min, y_3 slightly increased, but the y_1 and y_2 output value increases, which is mainly due to outside interference or electrolytic process operating time make the network input changes; Y_1 output value increases between 50 and 60 min, but it is useless to reach the limits of the fault prediction. After 1 h, the fault diagnosis model output values y_1 is greater than 0.5, which illustrating the anode effects of electrolysis cell, that making the anode effect prediction permitted; after 80 min, the fault diagnosis model output y1 value is above 0.9, close to 1. That can be seen the electrolysis cell anode effect and the forecast is in advance the amount of about 20 min.

Multi-fault simulation prediction analysis: Fig. 5 shows the experimental simulation curves simultaneously for multiple fault diagnosis. From the simulation curve it can be seen that the model output y_3 does not change significantly and the changing range is from 0.1-0.25, then y_1 and y_2 change. At the first 25 min, Y_1 and y_2 output are about 0.1, which illustrates the electrolysis is no exception in normal working condition; Y_2 output increases between 25 to 45 min and the maximum increasing value is not more than 0.5. After 45 min, y_2 output value is more than 0.5 which achieves the maximum hot tank limit predicted value and that indicates fault of the hot tank will take place, at this time the warning signal of the hot tank is shown. At the 55th min, y_1 and y_2 are more than 0.5 limits, which may be issued prediction value for multi-fault prediction limits (anode effect and the hot tank occurred simultaneously); At 73 min y_2 output value reaches to 0.9, which indicates the hot tank occurred. At about 85 min, the anode effect failure occurs and about 91 min to lift, y_1 output is reduced to about 0.2, which means the hot tank is continued.

Conclusion

A modular integrated fuzzy neural network electrolytic fault diagnosis method is put forward in the paper and the two layer the fault diagnosis of networks is built which is sub-fault diagnosis and decision-making fusion diagnosis. Combining the modular and integrated network (such as fuzzy and neural network, subnet and information decision-making fusion network organically), make the sub-fault diagnosis network modularly are all mixed at this platform.. Using the functions of neural network nonlinear mapping ability, the capacity of associative memory, fuzzy reasoning and computing capabilities, fault diagnosis network has a better fault tolerance, logical reasoning and adaptive ability. In the end, the simulation results show that by using the method of multi-fault diagnosis of aluminum electrolysis based on modular fuzzy neural networks there will be higher accuracy of fault prediction in advance volume, which verify the validity of the method of fault diagnosis.

REFERENCES

1. Z.-Q. Zhao, J. Gao, H. Glotin and X.D. Wu, *Appl. Mathemat. Model.*, **34**, 3884 (2010).
2. X. Xia, *J. Shenyang Jianzhu Univ*., **26**, 399 (2010).
3. T.Z. Tan, C. Quek, G.S. Ng and E.Y.K. Ng, *Expert Syst. Appl*., **33**, 652 (2007).
4. N. Chen, *J. Computer-Aided Design & Computer Graphics*, **24**, 443 (2012).
5. B. Ma and N. Li, *J. Shenyang Jianzhu Univ*., **28**, 375 (2012).
6. N. Li, *Inf. Control*, **41**, 81 (2012).