Integration of Neuro-Fuzzy and Genetic Algorithms for System Identification

Chuen-Jyh Chen [#1], Shih-Ming Yang [*2], Shih-Guei Lin [*3]

[#] *Department of Aviation Service Management, Aletheia University*
Matou, Tainan, Taiwan, R.O.C.

[1]cjchen@mail.au.edu.tw

[*] *Institute of Aeronautics and Astronautics, National Cheng Kung University*
No.1, University Road, Tainan City 701, Taiwan, R.O.C.

***Abstract*—It is known that neuro-fuzzy system is easily stuck in local minimum. To improve these drawbacks, a two-stage algorithm combining the advantages of neuro-fuzzy and genetic algorithms (GA) is integrated in system identification. Genetic algorithms are general purposed optimization algorithms with adaptive reproduction, crossover, and mutation operators that provide a method to search optimal parameters. The purpose of this paper is mainly using genetic algorithms individually to tune weights and membership functions of neuro-fuzzy system. Integrating neuro-fuzzy system and genetic algorithms is shown the better performances comparisons than neuro-fuzzy system in system identification. By applying the neuro-fuzzy system with genetic algorithms to system identification in this paper have been very successful.**

***Keywords*—Neuro-fuzzy system, Genetic algorithms, System identification**

1. INTRODUCTION

Since Zadeh proposed the fuzzy set in 1965, applications of fuzzy logic system have had increasing attention. A fuzzy system that contains human thinking and reasoning is able to deal with inexact information. Most applications are aimed at fuzzy modeling and fuzzy logic control [1-3]. The former uses linguistic description to establish a logic-based system model with fuzzy predicates, while the latter is a knowledge-based system operating in linguistic, rule-based structure. One of the advantages of fuzzy logic controller is that it does not require mathematical model. However, its performance strongly depends on the selection of input and output membership functions and the fuzzy rules, which, conventionally, are determined by experts' knowledge or experiences. For systems with practical complexity and/or uncertainty, it is often difficult to extract the "inside" knowledge so as to determine the adequate fuzzy structure, membership functions, and logic rules.

Many effective learning algorithms for neuro-fuzzy systems are developed to optimize the input/output membership functions and fuzzy logic rules. Shi and Mizumoto [5,6] developed a neuro-fuzzy learning algorithm for tuning fuzzy rules by using training input-output data. Gorrostieta and Pedraza [6] developed a neuro fuzzy modeling of control systems. Zaheeruddin and Garima [7] proposed a neuro-fuzzy approach for prediction of human work efficiency in noisy environment. Cheng [8] applied a neuro-fuzzy inference system for system modeling. Banakar and Azeem [9] also presented an identification and prediction of nonlinear dynamical plant using TSK and wavelet neuro-fuzzy model. A cluster-based self-organizing neuro-fuzzy system (SO-NFS) for control of unknown plants by Li and Lee [10]. Castellano *et al*. [11] proposed a neuro-fuzzy approach for the extraction of a recommendation model from usage data encoding user navigational. Esfahanipour and Aghamiri [12] proposed a Takagi–Sugeno–Kang (TSK) type fuzzy rule based system for stock price prediction by using a neuro-fuzzy inference system. Javadi-Moghaddam and Bagheri [13] presented an adaptive neuro-fuzzy sliding-mode-based genetic algorithm control system for a remotely operated vehicle with four degrees of freedom. Based on the model of Li and Lee [10], Hadavandi *et al.* [14] emphasized the learning algorithm and finding fuzzy rules, and Yang *et al.* [15, 16] and Chen *et al.* [17-19] implemented a neuro-fuzzy system with three phases in the five-layer feedforward network using Mamdani's fuzzy model. The above learning algorithms are of the neuro-fuzzy systems based on the Mamdani/Sugeno fuzzy model that contains a number of IF-THEN rules with antecedent and consequent parts in linguistic terms.

The main interest of fuzzy systems is to construct a fuzzy relationship model expressed by a set of linguistic representations from the experience of the expert or observed input-output data. In practice, however, it is almost impossible to establish effectively and efficiently such a fuzzy relationship model. Genetic algorithm (GA) is an optimization algorithm based on the mechanism of natural selection and genetics, it has been demonstrated to be an effective global optimization tool. Sural [20] proposed a genetic algorithm used for feature selection with a feature quality index metric. The feature vectors by defining fuzzy sets on hough transform of character pattern pixels. Hadavandi *et al.* [14] proposed a novel approach based on genetic fuzzy systems and self-organizing map clustering for building a stock price forecasting expert system, with the aim of improving forecasting accuracy. Shi and Mizunoto [21] presented a hybrid model for bankruptcy prediction by using genetic algorithm, fuzzy c-means algorithm, MARS. Bhattacharya and Das [22] proposed a method uses genetic algorithm and adaptive neuro fuzzy techniques to implement the recognition scheme for feature selection. It is further illustrated that a neuro-fuzzy model can be used in a genetic algorithm based control scheme to increase the efficiency of fuzzy control by Lin and Zheng [23]. Sivapathasekaran *et al*. [24] proposed an artificial neural network modeling and genetic algorithm based medium optimization for the improved production of marine biosurfactant. Talaat *et al*. [25] presented the design and implementation of a decentralized power system stabilizer for a wide range of variations in system parameters and/or loading conditions by using a decentralized GA-optimized neuro-fuzzy model. Li *et al*. [26] proposed a double chains quantum genetic algorithm for the direction of rotation angle of quantum rotation gates by using a normalized neuro-fuzzy controller. Mehrabi *et al*. [27] proposed an FCM-based neuro-fuzzy inference system and genetic algorithm-polynomial neural network model for the thermal conductivity of alumina–water nanofluids. Shahlaei *et al*. [28] presented an expert system based on genetic algorithm–adaptive neuro-fuzzy inference system for the cathepsin K inhibitory activity of studied compounds to construct the nonlinear quantitative structure–activity relationship model. In this paper, a neuro-fuzzy based on five-layers and three-phase learning scheme is developed by using genetic algorithms for system identification.

2. Neuro-Fuzzy System and Genetic Algorithm

2.1. Neuro-Fuzzy System

A neuro-fuzzy system with five-layer feedforward and the fuzzy inference of Mamdani fuzzy model is developed as shown in Fig. 1. By determining the fuzzy logic rules and optimizing the membership functions through the connective weights, a valid neuro-fuzzy system is established. Layer 1 defines the input nodes and layer 5 indicates the output nodes. Layer 2 and layer 4 are the term nodes of membership function to express the linguistic terms. Layer 3 defines the nodes representing the fuzzy rules. A series-parallel identification model for nonlinear system can be written as

$$\hat{y}(k+1) = f\,(y(k), y(k-1),\ldots,y(k-n+1); \quad u(k), u(k-1),...,u(k-m+1)) \tag{1}$$

where $\hat{y}(k+1)$ is the estimated output of the neuro-fuzzy model at time step k+1, [$u(k)$, $y(k)$] represents the input/output pair of the plant at time k, and n and m are the maximum lags in the input and output respectively. Equation (1) indicates that $\hat{y}(k+1)$ is a function of the past n values of the plant output $y(k-i)$, i = 0,1,…, n-1, and the past m values of the input $u(k-i)$, i = 0,1,…, m-1. By determining the fuzzy logic rules and optimizing the membership functions through the connective weights, a neuro-fuzzy model is established.

The design of a neuro-fuzzy system is by the three- phase learning process to locate the initial membership functions in phase 1, find the fuzzy rules in phase 2, and tune the membership functions of the input/output variables in phase 3. In phase 1, the center and the width of the initial membership function are determined by the feature-map algorithm:

$$\| x(k) - m_c(k) \| = \min\{\|x(k) - m_i(k)\|\} \tag{2}$$

$$m_c(k+1) = m_c(k) + \alpha(x(k) - m_c(k)) \tag{3}$$

$$m_i(k+1) = m_i(k) \text{ for } m_i \neq m_c \tag{4}$$

where $x(k)$ and $m_i(k)$ are the input and the center of membership function, respectively. The subscript c indicates the associative closest value and α is a decreasing rate. This adaptive formula runs independently for each input and output linguistic variables. Once $m_i(k)$ is calculated, the width $\sigma_i(k)$ can be determined by the first-nearest-neighbor heuristic,

$$\sigma_i = (m_i - m_c)/r \tag{5}$$

where r is the overlap parameter. After the membership functions of σ_i and m_i have been calculated, the backpropagation learning algorithm is to find the fuzzy rules in phase 2. The output of layer 2 is transmitted to layer 3 to find the firing strength of each rule node. Based on the firing strength and the node output in layer 4, the correct consequence-link for each node can be determined by using error backpropagation to minimize the error function $E = (d(k) - y(k))^2 / 2$, where $d(k)$ is the desired output and $y(k)$ is the current output. The weight of the links from layer 3 to 4 is tuned via the update rule,

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k) \tag{6}$$

where $\Delta w_{ij}(k) = \eta(d(k) - y(k)) O_{j3} w_{ij} \sigma_{i4}$

$$\frac{m_{i4}(\sum \sigma_{i4} O_{i4}) - (\sum m_{i4} \sigma_{i4} O_{i4})}{(\sum \sigma_{i4} O_{i4})^2} \text{ if } j = \hat{r}$$

and $\Delta w_{ij}(k) = 0$, otherwise, $\hat{r} = Arg \max_j (O_{j3}(w_{ij})^2)$ and η is the learning rate. By adjusting the weight, the correct consequent link of each rule node is determined, and for every antecedent clause, the centroid of all the possible consequent is calculated. Only the dominant rule whose consequent has the highest membership value is selected.

After the fuzzy rules have been deduced, a supervised learning is applied to tune optimally the membership functions in phase 3. Initially, the links between rule nodes in layer 3 and consequent nodes in layer 4 are fully connected, for the consequence of rule nodes are not decided yet. Starting at the output node, a backward pass is to compute the gradient of the error function for all hidden layer nodes. In layer 5, the center and the width of each Gaussian membership function are the adjustable parameters. The error propagated to the proceeding layer is

$$\delta_5(k) = d(k) - y(k) \tag{7}$$

By using Eq.(4) and the gradient of center m_{i4}, the center is updated via

$$m_{i4}(k+1) = m_{i4}(k) + \eta(d(k) - y(k)) \cdot \sigma_{i4} O_{i4} / \sum \sigma_{i4} O_{i4} \tag{8}$$

Similarly, the width parameter is

$$\sigma_{i4}(k+1) = \sigma_{i4}(k) + \eta(d(k) - y(k)) \cdot O_{i4} \frac{m_{i4}(\sum \sigma_{i4} O_{i4}) - (\sum m_{i4} \sigma_{i4} O_{i4})}{(\sum \sigma_{i4} O_{i4})^2} \tag{9}$$

The error signal in layer 4 is derived as

$$\delta_{i4}(k) = (d(k) - y(k)) \cdot \sigma_{i4} \frac{m_{i4}(\sum \sigma_{i4} O_{i4}) - (\sum m_{i4} \sigma_{i4} O_{i4})}{(\sum \sigma_{i4} O_{i4})^2} \tag{10}$$

By the same token, only the error signal δ_{i3} is needed and it is identical to δ_{i4}. In layer 2, the center and width parameter are updated by

$$m_{i2}(k+1) = m_{i2}(k) - 2O_{i2}\eta \frac{(O_{i1} - m_{i2})}{(\sigma_{i2})^2} \sum_k q_k \tag{11}$$

$$\sigma_{i2}(k+1) = \sigma_{i2}(k) - 2O_{i2}\eta \frac{(O_{i1} - m_{i2})^2}{(\sigma_{i2})^3} \sum_k q_k \tag{12}$$

where $q_k = 1$ when O_{i2} = min(input of the k_{th} rule node) and $q_k = 0$ for the others. The above learning algorithm highlights the computation procedures in the design of neuro-fuzzy model.

2.2. Genetic Algorithm

Genetic algorithm is an optimization technique developed by Holland (1975). It is a computational model inspired by population genetics for optimization based on the mechanism of natural selection and natural genetics. The coding method allows genetic algorithm to handle multiparameters or multimodel type of optimization problems easily, which is rather difficult or impossible to be treated by classical optimization method. The genetic algorithm evolves a multiset of elements called a population of individuals, also called strings or chromosomes. Each individual v_i (i = 1,…,n) of the population V represents a trial solution of the problem. Chromosomes are usually represented by string of variable, and each element of which is called a gene. Every gene controls the inheritance of one or several characters. The value of a gene is called allelic value, and it usually ranges within [0, 1]. A genetic algorithm is capable of maximizing a given fitness function computed for each individual of the population. If the problem is to minimize a given objective function f, this is equivalent to maximizing a function g where $g = -f$, i.e.,

$$min\, f(x) = max\, g(x), \tag{13}$$

where x is the variable of objective function. The structure of genetic algorithm is shown in Fig. 2. The algorithm can be described by reproduction, crossover, and mutation. Reproduction is a process by which the most highly rated chromosomes in the current generation are reproduced in the new generation. For the selection process, a roulette wheel with slots sized according to fitness is used widely. The roulette wheel process is to calculate the fitness value for each chromosome $q_i = \sum_{j=1}^{i} p_j$, $i = 1, \ldots,$ s, where s is population size, and to find the total fitness of the population

$$f = \sum_{i=1}^{s} e(v_i) \tag{14}$$

where $e(v_i)$ is fitness function. Calculate the probability of a selection p_i for each chromosome v_i

$$p_i = \frac{1}{f} e(v_i) \tag{15}$$

Calculate a cumulative probability q_i for each chromosome v_i

$$q_i = \sum_{j=1}^{i} p_j \tag{16}$$

Generate a random number r within (0, 1). If $r < q_i$, then select the first chromosome v_i , otherwise select the I chromosome v_i ($2 \leqq i \leqq s$) such that $q_{i-1} < r \leqq q_i$. Crossover provides a mechanism for chromosomes to mix and match by random processes in the follow way. For each chromosome in the (new) population. Generate a random number r within the range of (0, 1). If $r < pc$, where pc is the probability of crossover, select the given chromosome for crossover. For each pair of coupled chromosomes, an integer random number pos is generated from the rang [0, …, m-1] where m is number of bits. The number pos indicates the position of the crossing point. For example, two chromosomes (1, …, 1_{pos+1} 1_{pos+1}, …,1) and (0, …, 0_{pos} 0_{pos+1}, …, 0) are replaced by a pair of their offspring (1, …, 1_{pos+1} 0_{pos+1}, …, 0) and (0, …, 0_{pos} 0_{pos+1}, …, 1). Mutation is a random alteration of some gene value in a chromosome. Every bit has an equal chance undergoing mutation, i.e., change from 0 to 1 or vice versa. For each chromosome in the current population and for each bit within the chromosome, generate a random number r from the range $0 < r < 1$. If $r < pm$, where pm is the probability of mutation, mutate the bit. The binary genetic algorithms can be applied to find the optimal centers, widths and weights of a neuro-fuzzy. The neuro-fuzzy system trained by binary genetic algorithm will be applied to system identification.

3. APPLICATION OF NEURO-FUZZY SYSTEM WITH GENETIC ALGORITHM

The schematic diagram of the identifier training process is shown in Fig. 3. After the learning process is completed, the membership functions and the fuzzy rules of a fuzzy system identifying this nonlinear system are constructed by the neuro-fuzzy with genetic model. The structure of the network can be easily interpreted in terms of a fuzzy system. Nodes in layers 2 and 4 act as membership functions to express the input-output fuzzy linguistic terms. The identification procedure of the proposed neuro-fuzzy model using a five-layer feedforward network consists of three learning phases. The first is to find the initial centers and widths of the membership functions by a self-organized learning technique that is analogous to statistical clustering. Secondly, the error backpropagation learning algorithm by batch learning is used to train the connection weights representing the certainty factors of the corresponding fuzzy rules. After the learning is completed, redundant rules are deleted by a pruning process for obtaining a concise fuzzy rule-base. Finally, a supervised learning scheme is used to optimally adjust the parameters of the membership functions of input and output variables. After the identification procedure is completed, the performance of the neuro-fuzzy model is validated by many other sets of input. The model will produce the output $\hat{y}(k)$ very close to the actual system output $y(k)$ if the training is successful, and the mean of square error (MSE) is used to index the performance difference:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2, \tag{17}$$

where $\hat{y}_i(k)$ and $y_i(k)$ are the desired and actual system outputs of the i^{th} output node. N represents the total number of outputs.

This simulation is verified from Yang *et al.* [15, 29] in which the plant to be identified is given by the second-order highly nonlinear difference equation

$$y(k) = \frac{y(k-1)y(k-2)\big(y(k-1)+2.5\big)}{1+y(k-1)^2+y(k-2)^2} + u(k). \tag{18}$$

Training data of 500 points are generated from the plant model, assuming a random input signal uniformly distributed in the interval [-2, 2] for building a neuro-fuzzy model for this plant. The neuro-fuzzy model has three input: $u(k)$, $y(k\text{-}1)$, $y(k\text{-}2)$ and one output $y(k)$. The inputs $u(k)$ and $y(k\text{-}1)$ are partitioned into seven fuzzy linguistic spaces {negative large (NL), negative medium (NM), negative small (NS), zero (ZE), positive small (PS), positive medium (PM), positive large (PL)}, and the input $y(k\text{-}2)$ is partitioned into five fuzzy linguistic spaces {NL, NS, ZE, PS, PL}. The output $y(k)$ is partitioned into eleven fuzzy spaces {NVL, NL, NM, NS, NVS, ZE, PVS, PS, PM, PL, PVL}. The first phase of learning algorithm described in the previous section is used to determine the initial center and width of each membership function of the input-output variables in the fuzzy model. The overlap parameter is set as $r = 1.5$. Fig. 4(a) and (b) show the membership functions of input $u(k)$, $y(k\text{-}1)$ and $y(k\text{-}2)$ and output $y(k)$ after finishing this learning phase. Note that the selection of membership function, the Gaussian membership function has the advantage of being smooth and nonzero at all points, and it is popular method for specifying fuzzy sets. The training parameters are set as: learning rate (η) is 0.01, moment coefficient (α) is 0.9, increasing learning rate is 1.05 and decreasing learning rate is 0.7. According to the structure of neuro-fuzzy network, there are 19 nodes in layer 2, and the number of rule node in layer 3 is $7 \times 7 \times 5 = 245$. The phase-two learning is used to find the correct fuzzy rule(s). Then the phase-three learning is applied to optimize the parameters of each membership function. Finding the correct fuzzy rule(s) and the parameters of each membership function to minimize a cost function subject to multiple constrains specified is so complex that it cannot be analytically or numerically solved. Binary genetic algorithm is therefore employed in this work to tune the neuro-fuzzy structure. The methods are individually shown by the two following steps: (i) In phase two, the number of weights describe w_{ij} is 11×245 among layer 3 and layer 4. Thus, a total of 2695 parameters are needed to be tuned by genetic algorithm. (ii) In phase three, it has 7+7+5+11=30 membership functions. Each Gaussian membership function is defined by two parameters (the center m and the width σ). The optimizing 30×2=60 parameters of the membership functions can be determined by genetic algorithm. The structure of phase two and three of neuro-fuzzy system are individually adjusted by genetic algorithm in this paper. The results under different parameters of genetic algorithm are shown in Fig. 5(a) and (b). It is obvious that crossover rate equal to 0.65 can obtain the best fitness function in Fig. 5(a), and the best fitness function can be acquired when population size is 80 in Fig. 5(b). In the above two steps, the algorithm parameters are set as follows: population size = 80, generation number = 1500, crossover rate = 0.65, mutation rate = 0.05, bit number = 10. Fig. 6 (a) and (b) show the fitness value $e(v_i)$ variations by the generations of genetic algorithm in phase two and phase three. The change of mean-squared error after the phase-two and the phase-three learning are shown in Fig. 7(a) and (b), where they are reduced effectively by selecting the correct fuzzy logic rules. The performance of the neuro-fuzzy model is tested by 200 data points obtained by using a sinusoid input signal $u(k)=sin(2\pi k/25)$. The resultant membership functions of input and output variables after finishing the learning phase-three are shown in Fig. 8(a) and (b). It shows the membership functions of the optimal center (mean) and width (variance) after the phase-three learning. The output of both the neuro-fuzzy model and the actual plant are shown in Fig. 9. After sufficient learning, the performance of the neuro-fuzzy model is validated by comparing its response to that of the system without identification.

To demonstrate the learning capability of the proposed method, the same case under different operating conditions are performed. The tests are repeated with the neuro-fuzzy [29], neural network [16], and the results are compared with the neuro-fuzzy with the genetic algorithm. Fig. 9 shows the simulate results between those methods. By using genetic algorithm in a five-layer neuro-fuzzy model, the performance comparison between the membership function adjustment and the weight adjustment has been demonstrated on the same case. The simulate result is closer to the benchmark and has a good approach. Both steps have the ability of tuning efficient for system identification. Instead of needing experts' experiences and knowledge, the five-layer neuro-fuzzy in different methods can be constructed just from the input-output training data. The criterion test on a nonlinear system with three inputs and one output reveals that the membership function and the weight can be adjusted reasonably and optimally.

4. CONCLUSIONS

1. The neuro-fuzzy based on five-layers and three-phase learning scheme is developed using genetic algorithms for system identification further. Genetic algorithms are general purpose optimization algorithms with a probabilistic component that provide a method to search optimal parameters. The combination of neuro-fuzzy system and genetic algorithms are integrated in system identification.
2. The three-phase learning scheme that combines a self-organized learning algorithm and a supervised learning algorithm to find the optimal structure and parameters of the neuro-fuzzy network is complex. In this paper, a performance comparison of the same case between adjusting the membership function and the weight in the neuro-fuzzy model which has a structure of five-layer is proposed by genetic algorithm. The simulation result of the two steps tuning method demonstrates that it is closer to the benchmark and has a good approach.
3. In this paper, integrating neuro-fuzzy system with genetic algorithm is superior than neuro-fuzzy and neural network systems in system identification. The mean-squared error can be successfully reduced by binary genetic algorithm. In addition, other type genetic algorithms such as real genetic algorithm can be developed to further improve the simulation results.

ACKNOWLEDGMENT

The authors gratefully acknowledge the reviewers for enhancing the style and clarify of this paper. This work was supported in part by the Aletheia University, Taiwan, R.O.C. under contract AU-AR-102-013.

REFERENCES

[1] Chen, T., and Lin, Y. C. “A Fuzzy-Neural System Incorporating Unequally Important Expert Opinions for Semiconductor Yield Forecasting,” *International Journal of Uncertainty, Fuzziness, and Knowledge-based Systems*, Vol. 16, No. 1, pp. 35–58, 2008.

[2] Hsiao, F. H., Xu, S. D., Lin, C. Y., and Tsai, Z. R. “Robustness design of fuzzy control for nonlinear multiple time-delay large-scale systems via neural-network-based approach,” *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, Vol. 38, No. 1, pp. 244–251, 2008.

[3] Chen, T., and Wang, Y. C. “A Hybrid Fuzzy and Neural Approach for Forecasting the Book-to-Bill Ratio in the Semiconductor Manufacturing Industry,” *International Journal of Advanced Manufacturing Technology*, 2010.

[4] Shi, Y. and Mizunoto, M., “Some Considerations on Conventional Neuro-Fuzzy Learning Algorithms by Gradient Descent Method,” *Fuzzy Sets and Systems*, Vol. 112, pp. 51-63, 2000.

[5] Shi, Y. and Mizumoto, M., “A New Approach of Neuro-fuzzy Learning Algorithm for Tuning Fuzzy Rules,” *Fuzzy Sets and Systems*. Vol. 12, No. 1, pp. 99-116, 2000.

[6] Gorrostieta, E. and Pedraza, C., “Neuro Fuzzy Modeling of Control Systems,” *IEEE Conference on Electronics, Communications and Computers*, pp. 23, 2006.

[7] Zaheeruddin and Garima, “A Neuro-Fuzzy Approach for Prediction of Human Work Efficiency in Noisy Environment,” *Applied Soft Computing Journal*, Vol. 6, No. 3, pp. 283-294, 2006.

[8] Cheng, K. H., “Hybrid Learning-Based Neuro-Fuzzy Inference System: A New Approach for System Modeling,” *International Journal Systems Science*, Vol. 39, No. 6, pp. 583-600, 2008.

[9] Banakar, A. and Azeem, M. F., “Identification and Prediction of Nonlinear Dynamical Plants Using TSK and Wavelet Neuro-Fuzzy Models,” *IEEE Conference on Intelligent Systems*, pp. 617-620, 2006.

[10] Li, C. and Lee, C. Y., “Self-Organizing Neuro-Fuzzy System for Control of Unknown Plants,” *IEEE Transactions on Fuzzy Systems*, Vol. 11, No. 1, pp. 135-150, 2003.

[11] Castellano, G., Fanelli, A. M. and Torsello, M. A., “Recommendation Rule Extraction by a Neuro-Fuzzy Approach,” *Eighth International Conference on Hybrid Intelligent Systems*, pp. 758-763, 2008.

[12] Esfahanipour, A. and Aghamiri, W., “Adapted Neuro-Fuzzy Inference System on Indirect Approach TSK Fuzzy Rule Base for Stock Market Analysis,” *Expert Systems with Applications*, Vol. 37, pp. 4742-4748, 2010.

[13] Javadi-Moghaddam, J. and Bagheri, A., “An Adaptive Neuro-Fuzzy Sliding Mode Based

Genetic Algorithm Control System for Under Water Remotely Operated Vehicle," *Expert Systems with Applications*, Vol. 37, pp. 647-660, 2010.

[14] Hadavandi, E., Shavandi, H., and Ghanbari, A., "Integration of Genetic Fuzzy Systems and Artificial Neural Networks for Stock Price Forecasting," *Knowledge-Based Systems*, Vol. 23, pp. 800-808, 2010.

[15] Yang, S. M., Tung, Y. J. and Liu, Y. C., "A Neuro-Fuzzy System Design Methodology for Vibration Control," *Asian Journal of Control*, Vol. 7, No. 4, pp. 393-400, 2005.

[16] Yang, S. M., Chen, C. J. and Huang, W. L., "Structural Vibration Suppression by a Neural-Network Controller with a Mass-Damper Actuator," Journal of Vibration and Control, Vol. 12, Iss. 5, pp. 495-508, 2006.

[17] Chen C. J., Yang, S. M. and Wung, Z. C., "A Neuro-Fuzzy System and Its Application to Vibration Suppression," *Journal of Aeronautics Astronautics and Aviation*, Series A, Vol. 41, No. 3, pp. 195-202, 2009.

[18] Chen C. J., Yang, S. M. and Wung, Z. C., "System Identification by Neuro-Fuzzy Model with Sugeno and Mamdani Fuzzy Rules," *Journal of Aeronautics Astronautics and Aviation*, Series A, Vol. 41, No. 4, pp. 263-270, 2009.

[19] Chen, C. J., Yang, S. M. and Chen, C. Y., "Development of a Rule Selection Mechanism by Using Neuro-Fuzzy Methodology for Structural Vibration Suppression," *Journal of Intelligent & Fuzzy Systems*, Vol. 25, No. 4, pp. 881-892, 2013.

[20] Sural, S., "A Genetic Algorithm for Feature Selection in a Neuro-Fuzzy OCR System," *IEEE Computer Society*, pp. 987-991, 2001.

[21] Shi, Y. and Mizunoto, M., "A Hybrid Model for Bankruptcy Prediction Using Genetic Algorithm, Fuzzy C-Means and MARS," *International Journal on Soft Computing*, Vol. 2, No. 1, pp. 12-24, 2011.

[22] Bhattacharya, M. and Das, A., "Genetic Algorithm Based Feature Selection in a Recognition Scheme Using Adaptive Neuro Fuzzy Techniques," *International Journal* of Computers, Communications & Control, Vol. 5, No. 4, pp. 458-468, 2010.

[23] Lin, J., and Zheng, Y. B., "Vibration Suppression Control of Smart Piezoelectric Rotating Truss Structure by Parallel Neuro-Fuzzy Control with Genetic Algorithm Tuning," *Journal of Sound and Vibration*, Vol. 331, pp. 3677-3694, 2012.

[24] Sivapathasekaran, C., Mukherjee, S., Ray, A., Gupta, A., and Sen, R., "Artificial Neural Network Modeling and Genetic Algorithm Based Medium Optimization for the Improved Production of Marine Biosurfactant," *Bioresource Technology*, Vol. 101, pp. 2884-2887, 2010.

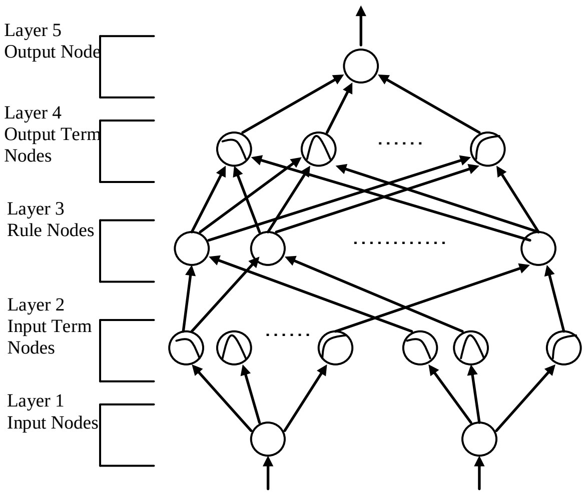
[25] Talaat, H. E. A., Abdennour, A., and Al-Sulaiman, A. A., "Design and Experimental Investigation of a Decentralized GA-Optimized Neuro-Fuzzy Power System Stabilizer," *Electrical Power and Energy Systems*, Vol. 32, pp. 751-759, 2010.

[26] Li, P. C., Song, K. P., and Shang, F. H., "Double Chains Quantum Genetic Algorithm with Application to Neuro-Fuzzy Controller Design," *Advances in Engineering Software,* Vol. 42, pp. 875-886, 2011.

[27] Mehrabi, M., Sharifpur, M., and Meyer, J. P., "Application of the FCM-Based Neuro-Fuzzy Inference System and Genetic Algorithm-Polynomial Neural Network Approaches to Modelling the Thermal Conductivity of Alumina–Water Nanofluids," *International Communications in Heat and Mass Transfer*, Vol. 39, pp. 971-977, 2012.

[28] Shahlaei, M., Madadkar-Sobhani, A., Saghaie, L., and Fassihi, A., "Application of an Expert System Based on Genetic Algorithm–Adaptive Neuro-Fuzzy Inference System (GA–ANFIS) in QSAR of Cathepsin K Inhibitors," *Expert Systems with Applications*, Vol. 39, pp. 6182-6191, 2012.

[29] Yang, S. M., Chen, C. J., Chang, Y. Y. and Tung, Y. Z., "Development of a Self-Organized Neuro-Fuzzy Model for System Identification," ASME *Transaction Journal on Vibration and Acoustics*, Vol. 129, No. 4, pp. 507-513, 2007.



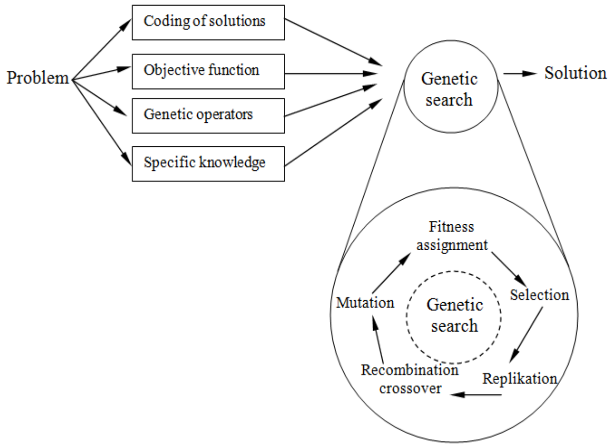


Figure 1 The structure of a self-organized, five-layer, neuro-fuzzy model.

Figure 2 Flow chart of the genetic algorithm.

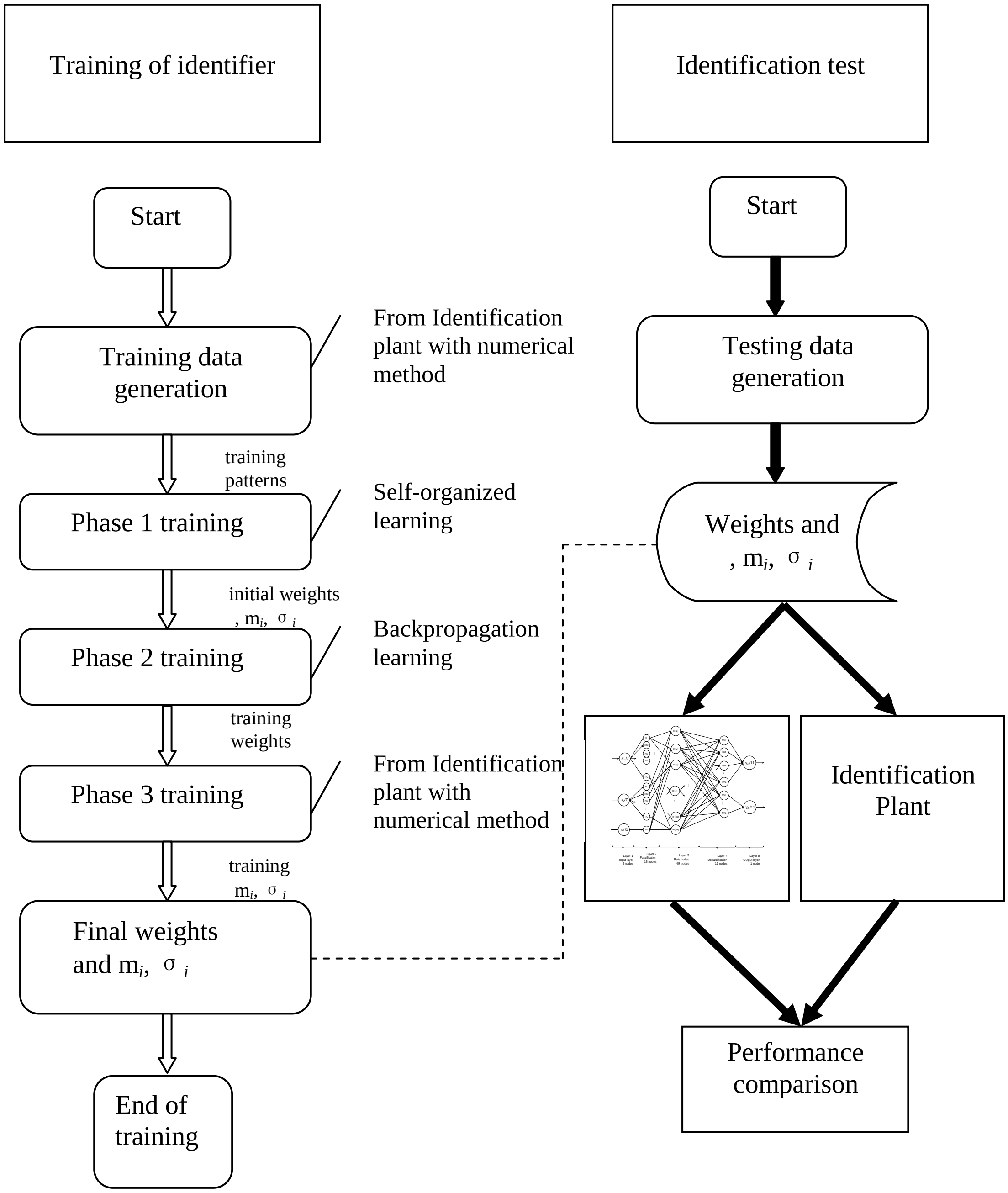
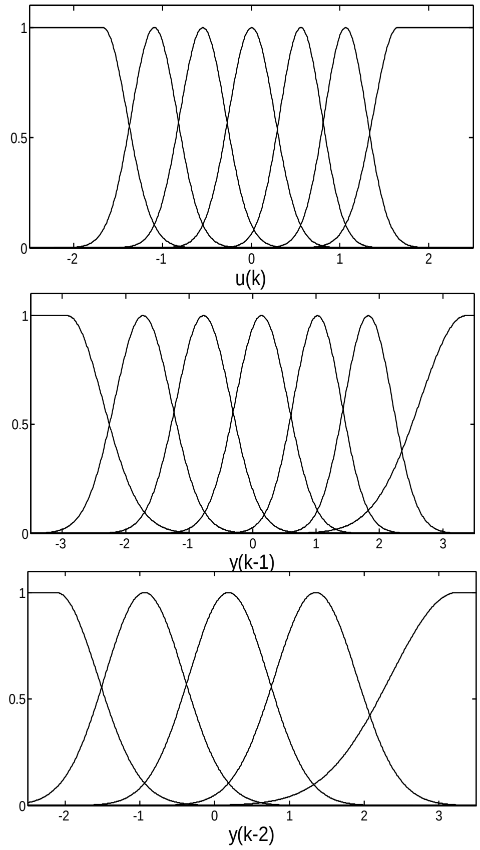
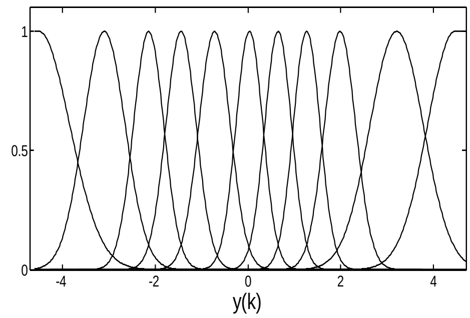


Figure 3 The schematic diagram of the identification training process.

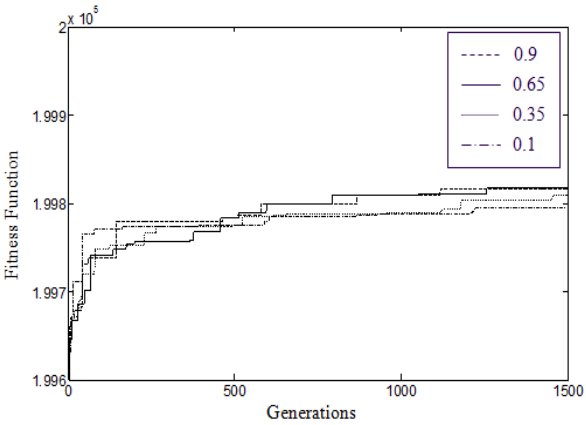


(a)

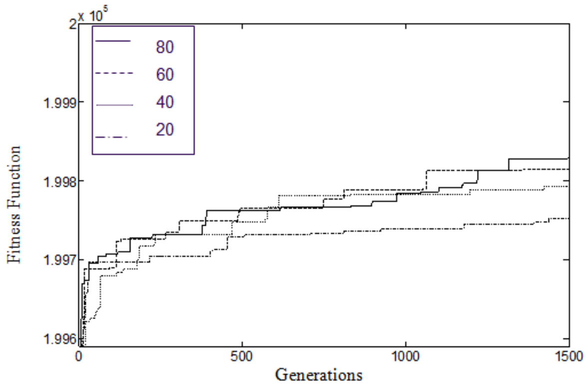


(b)

Figure 4 The membership functions for the example after the phase-one learning: (a) the input u(k), y(k-1) and y(k-2), (b) the output y(k) of the neuro-fuzzy model.

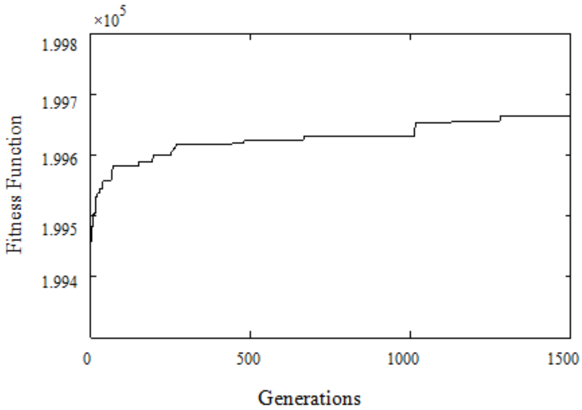


(a)

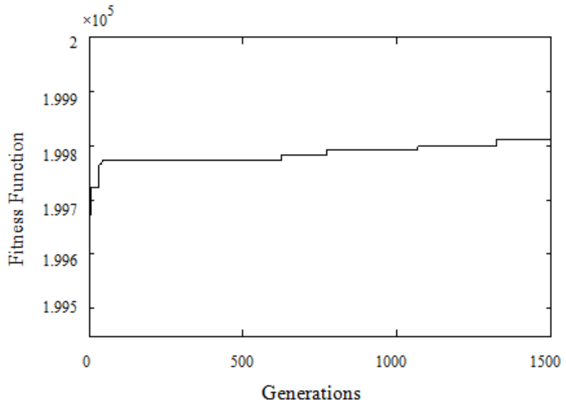


(b)

Figure 5 The results of fitness function under different (a) crossover rates (b) population sizes of genetic algorithm.

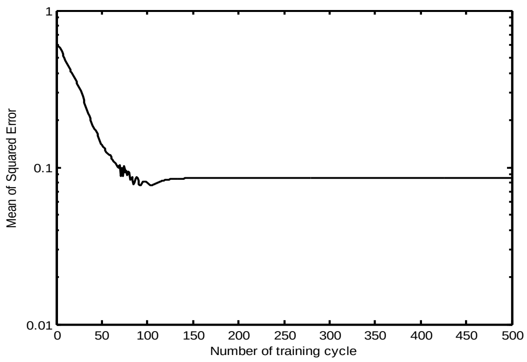


(a)

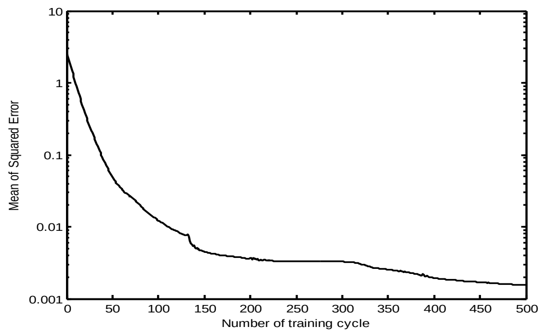


(b)

Figure 6 Fitness value variations by the generations of genetic algorithm in (a) the phase-two and (b) the phase-three.

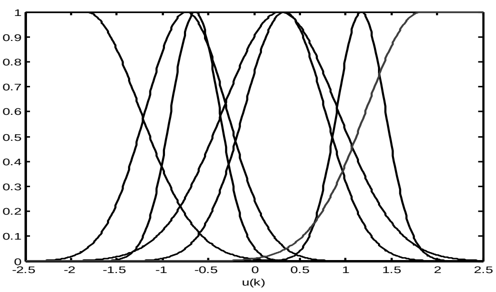


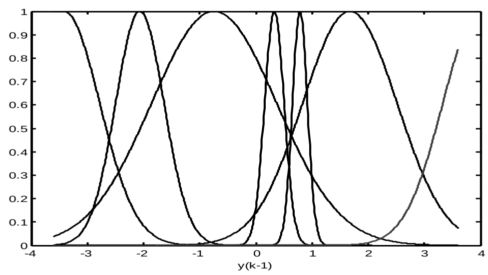
(a)

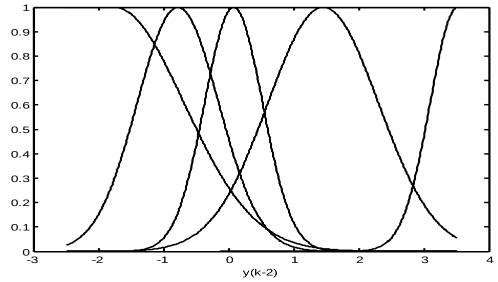


(b)

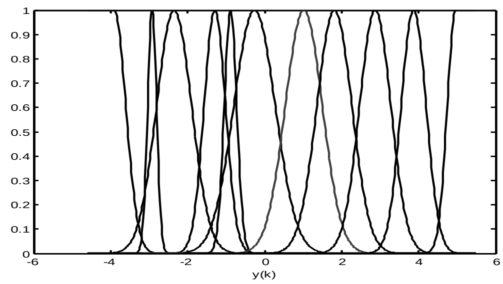
Figure 7 Variations of the mean-squared error by using genetic algorithm (a) the phase-two learning process and (b) the phase-three learning process.







(a)



(b)

Figure 8 The membership function using the two steps in three phase training: (a) the input u(k), y(k-1) and y(k-2), (b) the output y(k) of the neuro-fuzzy model.

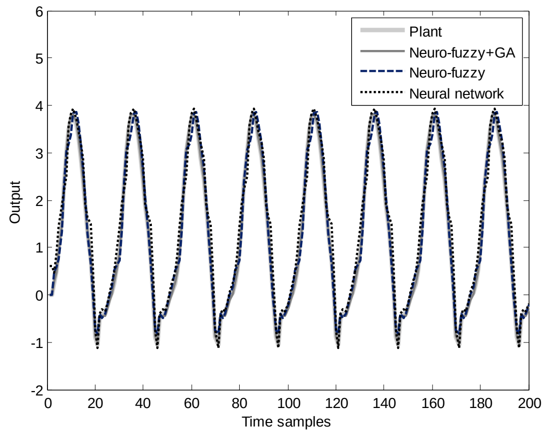


Figure 9 Performance comparison with different identification methods by using a sinusoidinput signal $u(k)=sin(2\pi k/25)$.