Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 161 (2010) 2585–2596

FUZZY sets and systems

www.elsevier.com/locate/fss

Design of fuzzy wavelet neural networks using the GA approach for function approximation and system identification

Shian-Tang Tzeng

Department of Electronic Engineering, Kao Yuan University, No. 1821 Chungshan Road, Luchu, Kaohsiung 821, Taiwan

Received 8 May 2009; received in revised form 30 April 2010; accepted 7 June 2010
Available online 12 June 2010

Abstract

In this paper, an efficient method is proposed to design fuzzy wavelet neural network (FWNN) for function learning and identification by tuning fuzzy membership functions and wavelet neural networks. The structure of FWNN is based on the basis of fuzzy rules including wavelet functions in the consequent parts of rules. In order to improve the function approximation accuracy and general capability of the FWNN system, an efficient genetic algorithm (GA) approach is used to adjust the parameters of dilation, translation, weights, and membership functions. By minimizing a quadratic measure of the error derived from the output of the system, the design problem can be characterized by the proposed GA formulation. Moreover, the solution is directly obtained without any need for complicated computations. The performance of our approximation is superior to that of existing methods. Several numerical design examples are likewise presented to demonstrate the design flexibility and usefulness of this presented approach.

Keywords: Fuzzy wavelet neural network; GA formulation

1. Introduction

As a result of rapid industrial developments, soft computing methodologies, such as fuzzy logic, neural network, and GA technique, are getting more and more important. The concepts of fuzzy logic, wavelet technology, neural network, and GA method have received a lot of attention in recent years [1–4]. Fuzzy controller is often used to deal with complex nonlinear systems with ill-defined conditions and uncertain factors. These control rules are based on expert knowledge. However, the knowledge may not be enough for some complicated systems. Thus, there are some methods used to generate IF–THEN rules [5,6]. On the other hand, there are several characteristics of neural networks such as learning ability, generalization, and nonlinear mapping [4]. These properties of neural networks are used to deal with signal processing, control system, decision making, and so on. However, the main problem of neural networks is that they require a large number of neurons to deal with the complex problems. Moreover, they also result in slow convergence and convergence to a local minimum. In order to overcome these disadvantages, wavelet technology is integrated into neural networks.

Recently, based on the combination of feed-forward neural networks and wavelet decompositions, wavelet neural network (WNN) has received a lot of attention and has become a popular tool for function learning [3]. The wavelet function is used as the nonlinear transformation in the hidden layer of neural network. The time–frequency property of wavelet is incorporated into the learning abilities of neural networks. However, the main problem of WNN with

E-mail address: smtstn@cc.kyu.edu.tw.

0165-0114/$ - see front matter
doi:10.1016/j.fss.2010.06.002

fixed wavelet bases is the selection of wavelet transform because the dilation and translation parameters of wavelet basis are fixed and only the weights are adjustable. The appropriate wavelet transform will result in the accuracy of approximation. Therefore, there are several different methods proposed to solve the problems [7–11], and their algorithms are given. According to some simple threshold rules shown in [7,8], new nodes are gradually added by a node-configuration strategy. Constructing and storing wavelet frames for large dimension problems entails prohibitive costs. In order to deal with these problems, it is important to note the sparse training data. On the other hand, by using radial wavelet frames with natural single-scaling characteristic, the algorithm of wavelet network has been constructed [9]. Moreover, a large number of wavelet neurons are required for WNN with fixed wavelet bases to improve the function approximation. Other problems also arise, such as a large complex network structure and overfitting. By initializing the WNN as truncated wavelet frames and training with a back-propagation algorithm, the linear and nonlinear components of the system can be modeled by waveARX neural network [10] and wavelet networks [11], respectively.

The complexity and uncertainty of the system can be also reduced and handled by the concepts of fuzzy logic. The local details of nonstationary signals can be analyzed by wavelet transforms. The approximation accuracy of the plant can be improved by the self-learning capabilities of neural networks. Therefore, there are many papers that discuss the synthesis of a fuzzy wavelet neural inference system for signal processing, control problems, reinforcement hybrid evolutionary learning, and pattern recognition [3,4,12–14,19,23,24]. For example, the fuzzy wavelet networks are constructed for function learning [3]. The model proposed by Abiyev and Kaynak [4,19] has been approved to be good and useful to solve the identification and control of dynamic plants very well. It makes use of the gradient decent method to update the parameters. A fuzzy model with wavelet transform is introduced in [12]. The wavelets in softcomputing are proposed in [13]. The chaotic time-series prediction using adaptive wavelet-fuzzy inference system is presented in [14]. A recurrent wavelet-based neuro-fuzzy system (RWNFS) with the reinforcement hybrid evolutionary learning is proposed in [23]. Based on the improved particle swam optimization, the neural-fuzzy networks are used to deal with the pattern recognition [24].

In this paper, the concepts of fuzzy logic are used in combination with WNN for arbitrary nonlinear function approximation and identification. Furthermore, it is important to adjust the required parameters in the design of dynamic systems. In order to avoid trial-and-error and time-consuming processes, the GA method is used as an optimization technique for complex problems. A self-tuning process that uses the GA approach is proposed to determine significant parameters such as dilation, translation, weights, and membership functions. Moreover, compared with the existing methods, it can reduce the number of fuzzy rules. The efficient method is expected to have good performance without requiring any derivatives or other auxiliary knowledge [15]. It converges toward the global solution for evaluating many points in a search space. The GA approach is briefly described in Section 2. In Section 3, the concepts of FWNN are explained. Several examples are presented to demonstrate the design capability and efficiency of the proposed approach in Section 4. Finally, the conclusions are given in Section 5.

2. The GA approach

The usual form of GA was described by Goldberg [16]. However, the binary representation traditionally used in GAs has some shortcomings when used to solve multidimensional and high-precision numerical problems [17]. Extensive computation is required. Genetic algorithms perform poorly at solving such problems. Therefore, in this study, these chromosomes are set to real-values rather than binary bit strings. Also, the design procedure of the GA approach is described as follows:

Step 1: Initial real-valued chromosomes of the population size (PS) are randomly generated.

Step 2: Crossover combines two parents to produce the children in the next generation. In order to produce improved offspring, the direction-based operator [18] is introduced into the genetic operation. It generates a single offspring B' from two parents B_1 and B_2 according to the following rule:

$$B' = r \cdot (B_1 - B_2) + B_1 \tag{1}$$

where r is a random number between 0 and 1. It also assumes that parent B_1 is not worse than B_2; that is $\mathit{fitness}(B_1) \leq \mathit{fitness}(B_2)$ for minimization problems. Consequently, it produces PS/2 offspring.

Step 3: Mutation rules randomly modify individual parents to produce children. The mutation rate is set to 1%.

Step 4: Evaluate these chromosomes (PS+PS/2+1%PS).

Step 5: The best PS chromosomes are selected among parents and offspring to form a new population for the next generation.

Step 6: Does the number of generations reach the specified value?

If the answer is YES, stop the process; otherwise, go to Step 2.

After numerous generations, the algorithm converges to the best chromosome, which represents the best solution to the problem. The population size critically represents the solutions to the problem at hand. If it is too small, numerous chromosomes that would have been useful are never tried out. However, a large computational effort is required if the population size is too large. In this work, the proper population size is set to 100.

3. FWNN

For the sake of convenience, only the multi-input–single-output (MISO) is taken into account in this paper. Therefore, $x = [x_1, x_2, \ldots, x_n]$ and y are set to be the input and output of the system, respectively. The basic concepts of fuzzy wavelet neural network method, originally presented by Abiyev and Kaynak [4,19], are briefly introduced in this section. The wavelet form is defined as follows:

$$\psi_{ij}(x_i) = \frac{1}{\sqrt{|d_{ij}|}}\psi(z_{ij}) \quad \text{for } d_{ij} \neq 0 \tag{2}$$

where

$$z_{ij} = \frac{x_i - t_{ij}}{d_{ij}} \tag{3}$$

Here, d_{ij}, t_{ij}, and $\psi_{ij}(x)$ stand for the dilation parameters, translation parameters, and family of wavelets obtained from the single $\psi(x)$ function, respectively. The subscript of *ij* denotes the *i*-th input and *j*-th output of the wavelet function. Moreover, a mother wavelet $\psi(x)$ is localized in both time space and frequency space. Wavelet functions included in the neural networks are the neurons of the hidden layer of the network. Therefore, the output of the *j*-th wavelet neural network can be represented as

$$\theta_j = \sum_{i=1}^{n} w_j \psi_{ij}(x_i) \tag{4}$$

where

$$\psi_{ij}(x_i) = \frac{1}{\sqrt{|d_{ij}|}}(1 - z_{ij}^2)\exp\left(-\frac{z_{ij}^2}{2}\right) \tag{5}$$

and w_j are the weighting coefficients. Notice that the Mexican Hat function $\psi(x) = (1 - x^2)\exp(-x^2/2)$ is used as the wavelet transform in this paper.

Furthermore, the fuzzy systems can be represented as a linear combination of the fuzzy basis functions introduced in [1,5,20]. Now suppose that there are M fuzzy IF–THEN rules in the following form:

$$R_j : \text{IF } x_1 \text{ is } A_{1j} \text{ and} \cdots \text{and } x_n \text{ is } A_{nj}, \quad \text{THEN } y \text{ is } \theta_j \tag{6}$$

where A_{ij} is a linguistic term characterized by a fuzzy membership function $\mu_{A_{ij}}(x_i)$ for $j = 1, \ldots, M$. The fuzzy basis function form with center average defuzzifier, product-inference rule, singleton fuzzifier, and Gaussian membership function is used here. The Gaussian membership function is a popular method for specifying fuzzy sets. Its curve has the advantage of being smooth and nonzero at all points. Therefore, the output of defuzzified inference is given as follows:

$$f(x) = \frac{\sum_{j=1}^{M} \theta_j (\prod_{i=1}^{n} \mu_{A_{ij}}(x_i))}{\sum_{j=1}^{M} (\prod_{i=1}^{n} \mu_{A_{ij}}(x_i))} \tag{7}$$

where $\mu_{A_{ij}}(x_i)$ is the Gaussian membership function defined by

$$\mu_{A_{ij}}(x_i) = \exp\left[-\frac{1}{2}\left(\frac{x_i - c_{ij}}{\sigma_{ij}}\right)^2\right] \tag{8}$$

where c_{ij} and σ_{ij} stand for the center and width parameters, respectively.

Assume that there are N input–output pairs: $(x(l), y(l))$, $l = 1, \ldots, N$. In this paper, our task is to design the fuzzy basis function expansion such that the error between $f(x(l))$ and $y(l)$ is minimized. Therefore, the fitness of k-th chromosome is defined as follows:

$$E_k = \sum_{l=1}^{N} |f_k(x(l)) - y(l)|^2 \tag{9}$$

where

$$f_k(x(l)) = \frac{\sum_{j=1}^{M} \theta_j^k \left[\prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i(l) - c_{ij}^k}{\sigma_{ij}^k}\right)^2\right)\right]}{\sum_{j=1}^{M} \left[\prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i(l) - c_{ij}^k}{\sigma_{ij}^k}\right)^2\right)\right]} \tag{10}$$

and

$$\theta_j^k = \sum_{i=1}^{n} w_j^k \psi_{ij}^k(x_i) = \sum_{i=1}^{n} w_j^k \frac{1}{\sqrt{|d_{ij}^k|}} \left(1 - \left(\frac{x_i - t_{ij}^k}{d_{ij}^k}\right)^2\right) \exp\left(-\frac{1}{2}\left(\frac{x_i - t_{ij}^k}{d_{ij}^k}\right)^2\right) \tag{11}$$

Therefore, the k-th chromosome is represented as

$$B_k = [c_{ij}^k \;\; \sigma_{ij}^k \;\; t_{ij}^k \;\; d_{ij}^k \;\; w_j^k]^T \quad \text{for } i = 1, \ldots, n \text{ and } j = 1, \ldots, M \tag{12}$$

where the superscript T denotes the vector transpose operation. Thus,

$$c_{ij}^k = [c_{11}^k \cdots c_{1M}^k \cdots c_{n1}^k \cdots c_{nM}^k] \tag{13}$$

$$\sigma_{ij}^k = [\sigma_{11}^k \cdots \sigma_{1M}^k \cdots \sigma_{n1}^k \cdots \sigma_{nM}^k] \tag{14}$$

$$t_{ij}^k = [t_{11}^k \cdots t_{1M}^k \cdots t_{n1}^k \cdots t_{nM}^k] \tag{15}$$

$$d_{ij}^k = [d_{11}^k \cdots d_{1M}^k \cdots d_{n1}^k \cdots d_{nM}^k] \tag{16}$$

and

$$w_j^k = [w_1^k \cdots w_M^k] \tag{17}$$

are all free design parameters. There are $(4n+1)M$ parameters to be updated in our FWNN model. The development of the GA method has provided another efficient approach for adjusting the required parameters in the design of FWNN structure. After applying the GA approach as stated in Sections 2, the best chromosome (solution) corresponding to the smallest fitness value can be obtained. The constructed FWNN is used to approximate the complex nonlinear functions more effectively. Moreover, based on our FWNN model, the network size is smaller and the learning speed is faster compared with other modeling structures.

4. Design examples

In order to evaluate the effectiveness and efficiency of the proposed algorithm for constructing the nonlinear dynamic system, the performance of our method is compared with those of waveARX neural network [10], wavelet networks [11], stable adaptive fuzzy control [21], and orthogonal least squares [22].

Example 1. The piecewise function is used to compare the performance of our FWNN with those of other wavelet-based networks [10,11]. This function is defined as

$$f(x) = \begin{cases} -2.186x - 12.864, & -10 \le x < -2 \\ 4.246x, & -2 \le x < 0 \\ 10\exp(-0.05x - 0.5)\sin[(0.03x + 0.7)x], & 0 \le x \le 10 \end{cases} \tag{18}$$

Obviously, this piecewise function is continuous and analyzable. However, it is inefficient owing to the use of the conventional tools. The wide-band information is not only hidden at the turning point but also has coexisting linearity and nonlinearity. In this example, 100 input–output pairs are randomly sampled in the region $[-10, 10]$ for training data. After applying the GA approach with 5000 generations for $n = 1$ and $M = 4$, the best chromosome corresponding to the smallest fitness value at each generation is presented in Fig. 1. It shows that the best fitness value improves rapidly in the early generations when the individuals are farther from the optimum. However, the best fitness value improves more slowly in later generations, whose populations are closer to the optimal point. When the best fitness value does not decrease any more for a long time, it means that the proposed approach finds the best chromosome (solution) of this problem. Therefore, the best solution is obtained in the 4758th generation. The approximation of piecewise function is presented in Fig. 2. The solid line and dashed line denote the desired output and actual output, respectively.

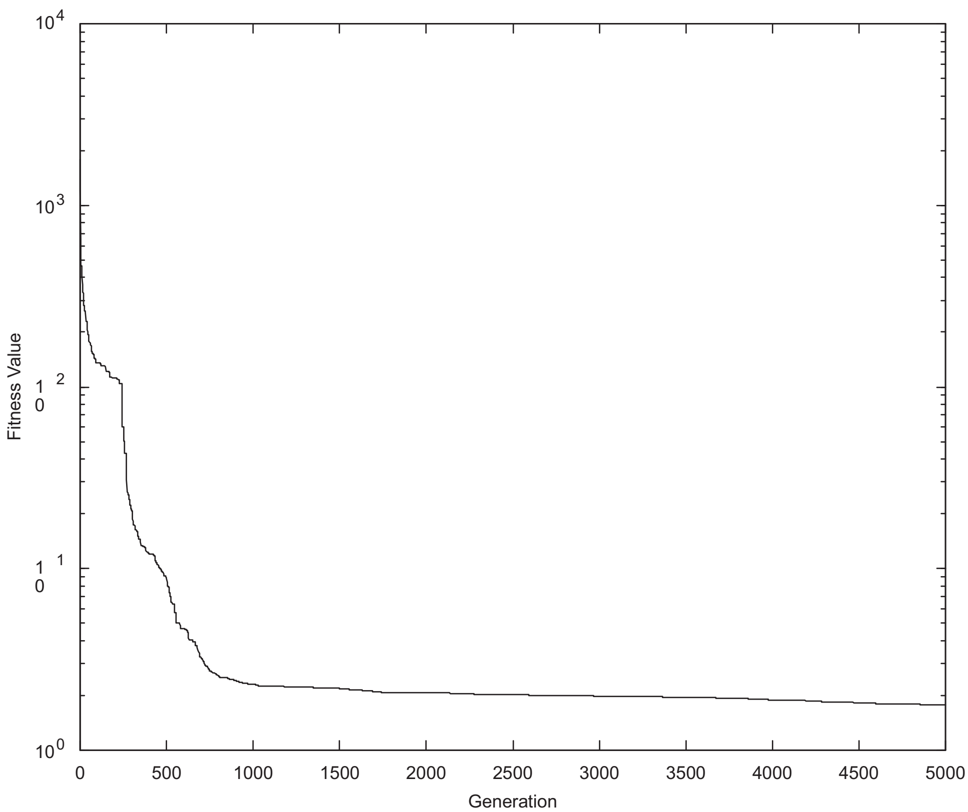

Fig. 1. Fitness.

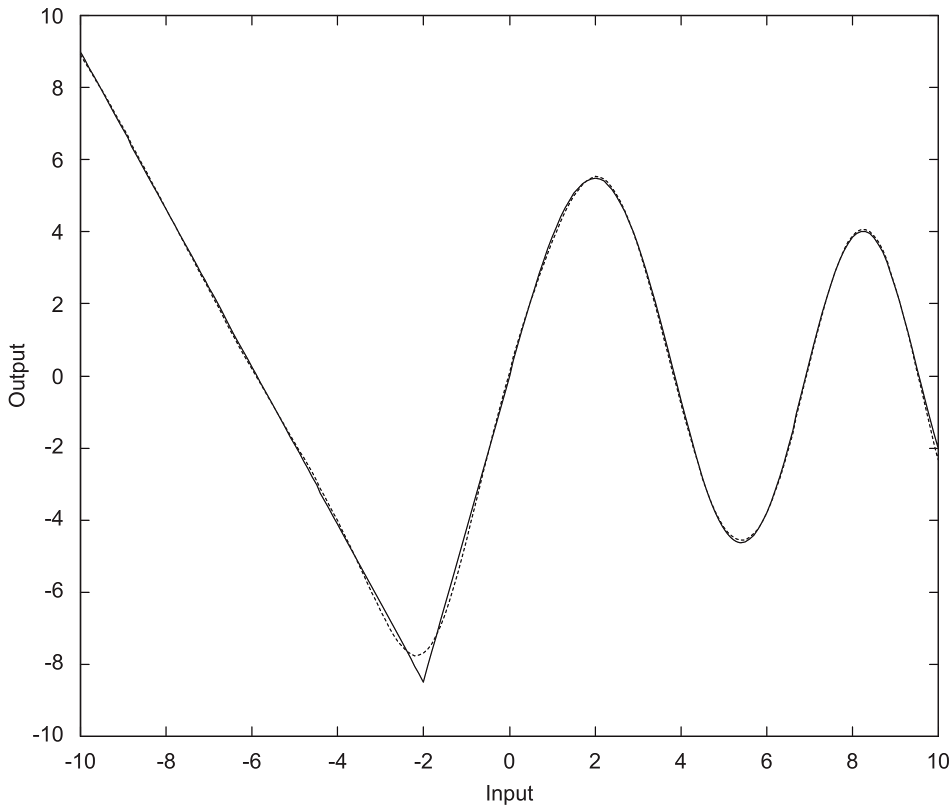

Fig. 2. Approximation of piecewise function.

In order to evaluate the performance of our model and others, the performance index [11] is defined as

$$J = \sqrt{\frac{\sum_{l=1}^{N} (y_l^d - y_l^a)^2}{\sum_{l=1}^{N} (y_l^d - \bar{y})^2}} \tag{19}$$

where y_l^d and y_l^a stand for the desired output and actual output, respectively. Here, $\bar{y}$ is represented as

$$\bar{y} = \frac{1}{N} \sum_{l=1}^{N} y_l^d \tag{20}$$

This example computes the uniformly sampled test data of 201 points. The performance of our FWNN compared with other wavelet-based method is shown in Table 1. The excellent performance of our FWNN model using the GA approach is presented.

Example 2. The adaptive FWNN controller that uses the GA approach is also used to regulate the plant [21]

$$\dot{x}(t) = \frac{1 - e^{-x(t)}}{1 + e^{-x(t)}} + u(t) \tag{21}$$

to the origin, $y = 0$. Obviously, the above plant is unstable without any control because $\dot{x} = (1 - e^{-x})/(1 + e^{-x}) > 0$ for $x > 0$ and $\dot{x} = (1 - e^{-x})/(1 + e^{-x}) < 0$ for $x < 0$. In order to design a suitable u to regulate the plant, the controller is defined as follows:

$$u = -\alpha x - \frac{1 - e^{-x}}{1 + e^{-x}} \tag{22}$$

Table 1
Comparison of FWNN with other wavelet-based networks.

Method	Number of design parameters	Performance index
FWNN	20	0.0303
WaveARX neural network [10]	23	0.0480
Wavelet networks [11]	22	0.0506

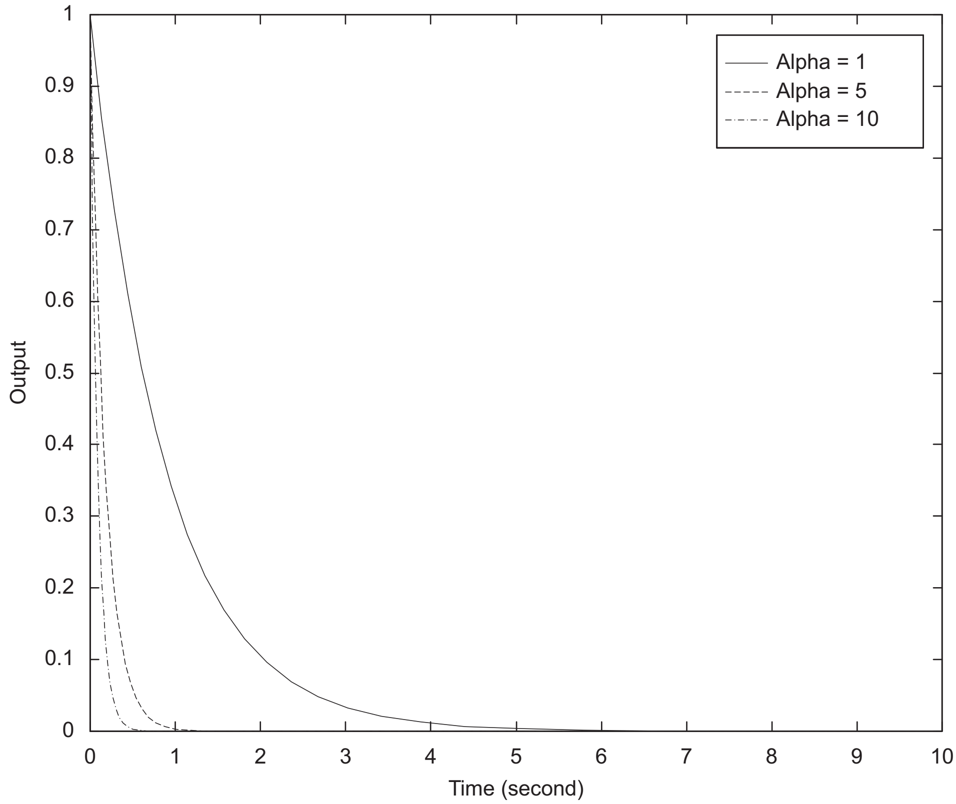

Fig. 3. Output of controlled system.

where α is the weighting constant that controls the relative speed of convergence in the control system. In general, the larger the parameter is, the faster the speed of convergence. In this example, u generates $N(x, u)$ pairs with randomly sampled x. Note that the "ode23" command of MATLAB software package is used to deal with the differential equation in this example. Moreover, the $f(x)$ with the FWNN structure was used as the controller u in (22). Namely, the $f(x)$ is used to regulate the plant instead of the original controller u.

After applying the GA approach with $\alpha = 1$, $n = 1$, $M = 2$, and $N = 100$, the output of the plant with solid line is shown in Fig. 3. Moreover, the output of $\alpha = 5$ and 10 are also shown in Fig. 3 with dashed line and dash-dot line, respectively. It demonstrates that the larger the weighting constant is, the faster the speed of convergence. The performance of our method, compared with that of stable adaptive fuzzy control [21], is also listed in Table 2 for reference. It is shown that the stable time and number of rules are less than that of [21]. It clearly presents that our approach is superior.

Table 2
Comparison of FWNN with stable adaptive fuzzy control.

Method	Stable time	Number of rules
FWNN	5.8 s ($\alpha = 1$) 1 s ($\alpha = 5$) 0.6 s ($\alpha = 10$)	2
Stable adaptive fuzzy control [21]	14 s[a] 18 s[b]	6

[a] With incorporating fuzzy control rules.
[b] Without incorporating any fuzzy control rules.

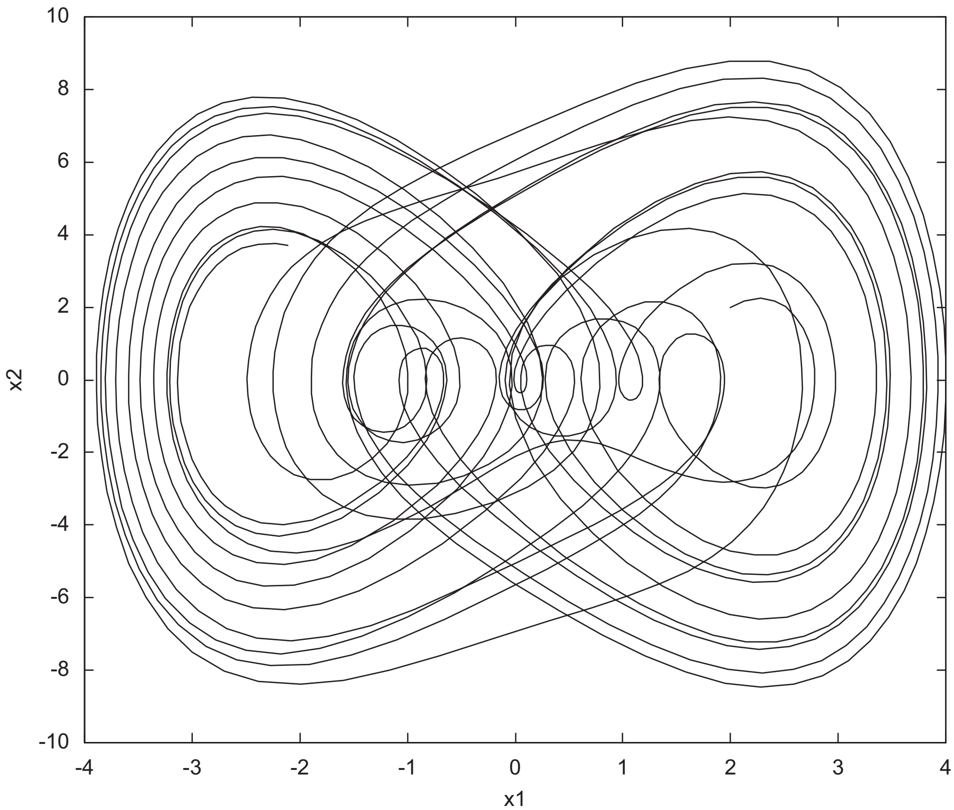

Fig. 4. Duffing forced-oscillation system.

Example 3. The Duffing forced-oscillation system [21] is taken into account in this example

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -0.1x_2 - x_1^3 + 12\cos(t) + u(t) \tag{23}$$

Clearly, the system is chaotic if there is no control. The trajectory of the system for $u(t) = 0$ is presented in Fig. 4 with the initial condition $x_1(0) = x_2(0) = 2$ and time period from $t_0 = 0$ to $t_f = 60$. The adaptive FWNN controller is also used to control the state x_1 to track the reference trajectory $y(t) = \sin(t)$. Note that this reference trajectory of the system is unit circle: $y^2 + \dot{y}^2 = 1$. In order to control the chaotic system, the controller is defined as follows:

$$u = -\sin(t) + 0.1x_2 + x_1^3 - 12\cos(t) \tag{24}$$

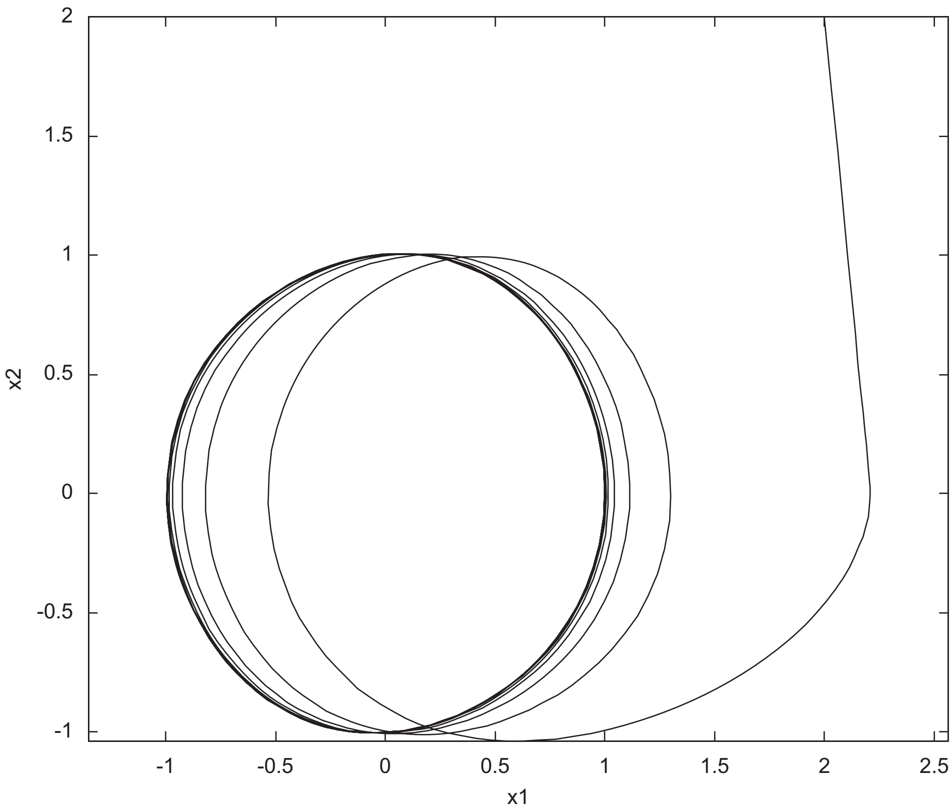

Fig. 5. Controlled chaotic system.

Table 3
Control of Duffing forced-oscillation system.

Method	Stable time (s)	Number of rules
FWNN	20	2
Stable adaptive fuzzy control [21]	50	6

Similarly, the $f(x)$ with the FWNN approach was used as the controller u in (23) and (24). After applying the efficient method with $n = 2$ and $M = 2$, the trajectory under the same initial conditions and time period is shown in Fig. 5. Moreover, the compared results are also listed in Table 3. It shows that our approach outperforms the others.

Example 4. In this example, the FWNN structure is used to approximate a controller for the nonlinear ball and beam system [20,22]. It is used as a controller to regulate the system to the origin from a certain range of initial conditions. The input–output linearization algorithm [22] is used to generate a set of state control pairs with randomly sampled points in a certain region of the state space. With arbitrarily chosen initial conditions, the proposed method is used to determine the parameters of FWNN.

The ball and beam system is shown in Fig. 6. It consists of a ball without friction rolling on a beam; a motor generates a torque to tilt the beam to send the ball to a desired location. A FWNN controller that uses the GA approach is designed to generate an appropriate torque to achieve the control goal. Let $x = [r, \dot{r}, \theta, \dot{\theta}]^T$ and $y = r$ be the state vectors and

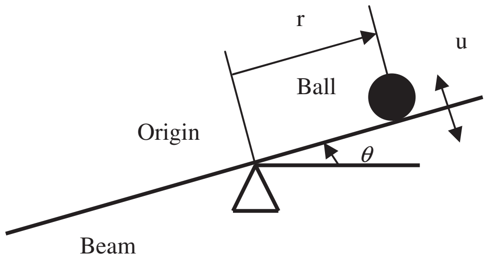

Fig. 6. Ball-and-beam system.

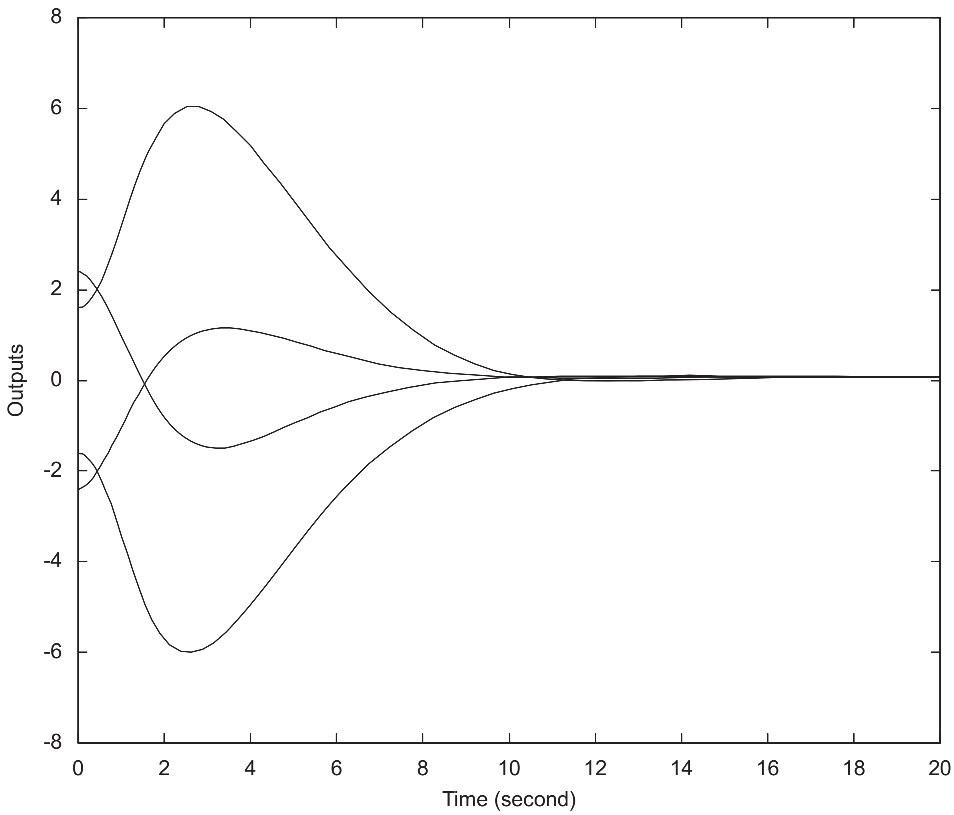

Fig. 7. Outputs with four initial conditions.

output of the system, respectively. Therefore, the system can be represented by the state space model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ B(x_1 x_4^2 - G \sin x_3) \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$
$$y = x_1 \tag{25}$$

The parameters of $B = 0.7143$ and $G = 9.81$ are defined in [22]. From arbitrary initial conditions, the system output y will converge to zero in a certain region through the controller u. The control law is determined by the input–output linearization algorithm [22]. It is represented in the following: for state x, compute $v = \alpha_3 BGx_4 \cos x_3 + \alpha_2 BG \sin x_3 - \alpha_1 x_2 - \alpha_0 x_1$; furthermore, the parameters of α_i are chosen such that $s^4 + \alpha_3 s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0$ is a Hurwitz polynomial. Then compute $a = -BG \cos x_3$ and $b = -BGx_4^2 \sin x_3$ such that $u = (v - b)/a$.

Table 4
Control of nonlinear ball and beam system.

Method	Stable time	Number of rules
FWNN	11 s ($N = 40$)	4
OLS algorithm [20]	16 s ($N = 200$)	
	27 s ($N = 40$)[a]	20

[a] Three initial conditions are under control and the other one is out of control.

The control $u = (v - b)/a$ is used to generate $N(x, u)$ pairs with state x randomly sampled in the region $U = [-5, 5] \times [-2, 2] \times [-\pi/4, \pi/4] \times [-0.8, 0.8]$. In this example, it is simulated for $n = 4$, $M = 4$ and $N = 40$. The $f(x)$ with the FWNN approach was used as the control u in (25) with four initial conditions: $x(0) = [2.4, -0.1, 0.6, 0.1]^T$, $[1.6, 0.05, -0.6, -0.05]^T$, $[-1.6, -0.05, 0.6, 0.05]^T$, and $[-2.4, 0.1, -0.6, -0.1]^T$, which were arbitrarily chosen in U.

The output of the nonlinear ball and beam system with four initial conditions are shown in Fig. 7. The FWNN controller gives the better overall performance. It means that the GA approach can determine the best parameters for the adaptive FWNN controller with a small number of sampling state control pairs. Furthermore, the performance of our method, compared with that of orthogonal least squares (OLS) algorithm [20], is listed in Table 4 for reference. It shows that the stable time and number of rules of the proposed algorithm are less than that of the OLS algorithm method. Obviously, it presents that our technique performs much better.

5. Conclusions

In this paper, the structure of the FWNN model that uses the GA approach is introduced for function approximation from input–output pairs and nonlinear dynamic system identification. It integrates the advantages of fuzzy concepts, wavelet functions, and neural networks. The parameter update algorithm of the proposed model is based on the efficient GA method. The faster convergence speed of the presented FWNN is also shown. In the identification, the performance is much better with a small number of parameters. It means that the proposed model is more adaptive to new data. For the control of an uncertain plant, an accuracy mathematical model of the system is not required. Therefore, an appropriate FWNN system that uses the GA approach is automatically selected to ensure high performance. It also avoids time-consuming and trial-and-error procedures.

The presented FWNN illustrates not only the multiresolution capability of WNN but also the advantages of high approximation accuracy and good generalization performance. The simplicity and generality of this proposed approach make it attractive for a wide variety of applications such as function approximation, nonlinear dynamic system identification, signal processing, control of uncertain plants, and robotics. Furthermore, the performance of the obtained system is superior to those of the existing methods. The simulation results are presented to demonstrate the efficiency and effectiveness of the proposed approach.

Acknowledgements

The author is deeply indebted to the reviewers for their interests, careful reviews of this manuscript, observations, and valuable suggestions. Responding to their constructive comments helped the author improve this paper.

References

[1] L.X. Wang, A Course in Fuzzy Systems and Control, Prentice-Hall International, Inc., Englewood Cliffs, NJ, 1997.
[2] O. Cordsn, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, Fuzzy Sets and Systems 141 (2004) 5–31.
[3] D.W.C. Ho, P.A. Zhang, J. Xu, Fuzzy wavelet networks for function learning, IEEE Transactions on Fuzzy Systems 9 (1) (2001) 200–211.
[4] R.H. Abiyev, O. Kaynak, Fuzzy wavelet neural networks for identification and control of dynamic plants—a novel structure and a comparative study, IEEE Transactions on Industrial Electronics 55 (8) (2008) 3133–3140.
[5] L.X. Wang, Adaptive Fuzzy Systems and Control: Design and Stability Analysis, Prentice-Hall International Inc., Englewood Cliffs, NJ, 1994.

[6] J.S.R. Jang, C.T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice-Hall Inc., Upper Saddle River, NJ, 1997.
[7] M. Cannon, J.J.E. Slotine, Space-frequency localized basis function networks for nonlinear system estimation and control, Neurocomputing 9 (3) (1995) 293–342.
[8] M.R. Sanner, J.J.E. Slotine, Structurally dynamic wavelet networks for adaptive control of robotic systems, International Journal of Control 70 (3) (1998) 405–421.
[9] Q. Zhang, Using wavelet networks in nonparametric estimation, IEEE Transactions on Neural Networks 8 (1997) 227–236.
[10] J. Chen, D.D. Bruns, WaveARX neural network development for system identification using s systematic design synthesis, Industrial & Engineering Chemistry Research 34 (1995) 4420–4435.
[11] Q. Zhang, A. Benveniste, Wavelet networks, IEEE Transactions on Neural Networks 3 (1992) 889–898.
[12] C.K. Lin, S.D. Wang, Fuzzy modeling using wavelet transform, Electronic Letters 50 (6) (1996) 1317–1334.
[13] M. Thuillard, Wavelets in Softcomputing, World Scientific, Singapore, 2001.
[14] Y. Lin, F.Y. Wang, Predicting chaotic time-series using adaptive wavelet-fuzzy inference system, in: Proceedings of 2005 IEEE Intelligent Vehicles Symposium, Las Vegas, Nevada, USA, June 2005, pp. 888–893.
[15] K.F. Man, K.S. Tang, S. Kwong, Genetic Algorithms, Springer-Verlag, London, 1999.
[16] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[17] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Third, Revised, and Extended Edition, 1996.
[18] M. Gen, R. Cheng, Genetic Algorithms and Engineering Design, John Wiley & Sons Inc., New York, 1997.
[19] R.H. Abiyev, O. Kaynak, Identification and control of dynamic plants using fuzzy wavelet neural networks, 2008 IEEE International Symposium on Intelligent Control, San Antonio, Texas, USA, September 3–5, 2008, pp. 1295-1301.
[20] L.X. Wang, J.M. Mendel, Fuzzy basis functions, universal approximation and orthogonal least squares learning, IEEE Transactions on Neural Networks 3 (5) (1992) 807–814.
[21] L.X. Wang, Stable adaptive fuzzy control of nonlinear systems, in: Proceedings of 31st IEEE Conference on Decision and Control, Tucson, 1992, pp. 2511–2516.
[22] J. Hauser, S. Sastry, P. Kokotovic, Nonlinear control via approximate input–output linearization: the ball-and-beam example, IEEE Transactions on Automatic Control 37 (3) (1992) 392–398.
[23] C.J. Lin, Y.C. Hsu, Reinforcement hybrid evolutionary learning for recurrent wavelet-based neurofuzzy systems, IEEE Transactions on Fuzzy Systems 15 (4) (2007) 729–745.
[24] C.J. Lin, J.G. Wang, C.Y. Lee, Pattern recognition using neural-fuzzy networks based on improved particle swam optimization, Expert Systems with Applications 36 (3) (2009) 5402–5410.