

Fuzzy Wavelet Neural Networks for Identification and Control of Dynamic Plants—A Novel Structure and a Comparative Study

Rahib Hidayat Abiyev, *Member, IEEE*, and Okyay Kaynak, *Fellow, IEEE*

Abstract—One of the main problems for effective control of an uncertain system is the creation of the proper knowledge base for the control system. In this paper, the integration of fuzzy set theory and wavelet neural networks (WNNs) is proposed to alleviate the problem. The proposed fuzzy WNN is constructed on the base of a set of fuzzy rules. Each rule includes a wavelet function in the consequent part of the rule. The parameter update rules of the system are derived based on the gradient descent method. The structure is tested for the identification and the control of the dynamic plants commonly used in the literature. It is seen that the proposed structure results in a better performance despite its smaller parameter space.

Index Terms—Control, fuzzy wavelet neural network (FWNN), identification, wavelet.

I. INTRODUCTION

IN MOST technological processes, dynamic systems are encountered that are characterized with uncertainties in terms of structure and parameters. These uncertainties cannot adequately be described by deterministic models, and therefore, conventional control approaches based on such models are unlikely to result in the required performance. Under such conditions, the use of soft computing methodologies can be a viable alternative.

Fuzzy technology is an effective tool for dealing with complex nonlinear processes that are characterized with ill-defined and uncertain factors. In this context, fuzzy controllers are often used in achieving a desired performance; the rule base of which is created on the base of the knowledge of human experts. However, for some complicated processes, this knowledge may not be sufficient, and several approaches [1], [2] have been proposed for the generation of the IF–THEN rules. Nowadays, for this purpose, the use of neural networks (NNs) has taken more importance. In this paper, the combination of fuzzy logic, NNs, and wavelet technology are used to solve identification and control of dynamic systems.

In literature, numerous different neural and fuzzy structures are proposed for solving identification and control problems

Manuscript received November 27, 2007; revised March 31, 2008. First published April 25, 2008; last published July 30, 2008 (projected). This work was supported by Near East University and Bogazici University.

R. H. Abiyev is with the Department of Computer Engineering, Near East University, Lefkosa, Mersin 10, North Cyprus (e-mail: rahib@neu.edu.tr).

O. Kaynak is with the Department of Electrical and Electronics Engineering, Bogazici University, Istanbul 34342, Turkey (e-mail: okyay.kaynak@boun.edu.tr).

Digital Object Identifier 10.1109/TIE.2008.924018

[2]–[10], and their parameter update algorithms are given. A well-known structure is the adaptive neuro-fuzzy inference system [2], another one is known as neural fuzzy controller [3], both being available for implementation under MATLAB/SIMULINK. In [4], a variable structure system theory-based training procedure is proposed, and in [5], its use in a neuro-adaptive scheme for the control of electrical drives is described, and experimental results are presented. In [6] and [7], fuzzy NNs are used for direct adaptive control of dynamic plants and for robust adaptive control of robot manipulators, respectively. Some of the neuro-fuzzy structures proposed utilize recurrent NNs. In [8], a recurrent fuzzy network is used for nonlinear modeling. In [9], a Takagi–Sugeno–Kang (TSK)-type recurrent neuro-fuzzy NN (TRFN) is developed, and in [10], the use of random set theory in a fuzzy scheme for identification of dynamic plants is considered.

The key factors for the use of NNs in the control field are the properties that they have such as: learning and generalization abilities, nonlinear mapping, parallelism of computation, and vitality. Due to these characteristics, NNs have become very popular in areas such as artificial behavior, artificial intelligence, control, decision making, identification, robotics, etc. However, for solving the majority of approximation problems, NNs require a large number of neurons. Furthermore, the NNs may get stuck on a local minimum of the error surface, and the network convergence rate is generally slow. A suitable approach for overcoming these disadvantages is the use wavelet functions (WFs) in the network structure. WF is a waveform that has limited duration and an average value of zero. A wavelet NN (WNN) has a nonlinear regression structure that uses localized basis functions in the hidden layer to achieve the desired input–output mapping. The integration of the localization properties of wavelets and the learning abilities of NN results in the advantages of WNN over NN for complex nonlinear system modeling [11], [12], and some researchers [13]–[17] have used such structures for solving approximation, classification, prediction, and control problems.

A fuzzy wavelet NN (FWNN) combines wavelet theory with fuzzy logic and NNs. The synthesis of a fuzzy wavelet neural inference system includes the determination of the optimal definitions of the premise and the consequent part of fuzzy IF–THEN rules. Several researchers [18]–[25] have used a combination of fuzzy technology and WNN for solving signal processing and control problems. In [18], a fuzzy system with a linear combination of basis function is proposed, and in

[19]–[21], the wavelet network model of a fuzzy inference system is proposed. In [19], the membership functions are chosen from a family of scaling functions, and the fuzzy system is developed by using wavelet techniques. A fuzzy wavelet network that includes the combination of three subnets: pattern recognition subnet, fuzzy reasoning subnet, and control synthesis subnet is presented in [22]. The use of such multilayer structures complicates the architecture of the system. An FWNN structure that is constructed on the base of a set of fuzzy rules is proposed in [23] and used for the approximation nonlinear functions. Other wavelet-based approaches include FWNN structures developed for control of dynamic plants [24] and for time-series prediction [25].

Wavelet transform has the ability to analyze nonstationary signals to discover their local details. Fuzzy logic allows us to reduce the complexity of the data and to deal with uncertainty. Their combination allows us to develop a system with fast learning capability that can describe nonlinear systems that are characterized with uncertainties. NNs have self-learning characteristics that increases the accuracy of the model. In this paper, these methodologies are combined to construct a fuzzy wavelet neural inference system to solve identification and control problems. In the following section, its seven-layer structure is explained. In Section III, the parameter update rules based on the gradient descent method are derived. In Section IV, the simulation studies are presented for both identification and control cases. The plant models considered are taken from the literature to enable a direct comparison of the proposed network with other approaches reported.

II. FWNN

Wavelets are defined in the following form:

$$\Psi_j(\mathbf{x}) = |\mathbf{a}_j|^{-\frac{1}{2}} \psi\left(\frac{\mathbf{x} - \mathbf{b}_j}{\mathbf{a}_j}\right), \quad \mathbf{a}_j \neq 0 \quad (1)$$

where $\Psi_j(\mathbf{x})$ represents the family of wavelets obtained from the single $\psi(x)$ function by dilations and translations, where $\mathbf{a}_j = \{a_{1j}, a_{2j}, \dots, a_{mj}\}$ and $\mathbf{b}_j = \{b_{1j}, b_{2j}, \dots, b_{mj}\}$ are the dilation and the translation parameters, respectively. $\mathbf{x} = \{x_1, x_2, \dots, x_m\}$ are the input signals, $j = 1, \dots, n$. $\psi(\mathbf{x})$ is localized in both time space and frequency space and is called a mother wavelet.

Wavelet networks include WFs in the neurons of the hidden layer of the network. The output of WNN is calculated as

$$\mathbf{y} = \sum_{j=1}^k \mathbf{w}_j \Psi_j(\mathbf{x}) = \sum_{j=1}^k \mathbf{w}_j |\mathbf{a}_j|^{-\frac{1}{2}} \psi\left(\frac{\mathbf{x} - \mathbf{b}_j}{\mathbf{a}_j}\right) \quad (2)$$

where $\Psi_j(\mathbf{x})$ is the WF of the j th unit of the hidden layer, w_j are weight coefficients between the input and the hidden layers, a_i and b_j are the parameters of WF as described above. WNN has good generalization ability, can approximate complex functions to some precision very compactly, and can be easily trained than other networks, such as multilayer perceptrons and radial-based networks [12], [15]. A good initialization of the parameters of WNNs enables to obtain fast convergence. A

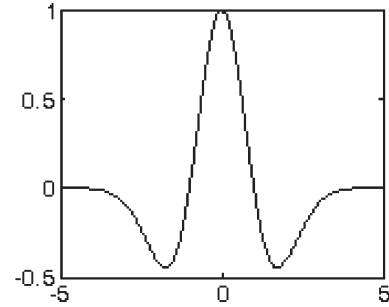


Fig. 1. Mexican Hat wavelet.

number of methods are proposed in literature for the initialization of the wavelets, such as the orthogonal least square procedure [12] and the clustering method [15]. An optimal initial choice of the dilation and the translation parameters of the wavelet increases the training speed and results in fast convergence. The approximation and convergence properties of WNN are presented in [14].

This paper presents an FWNN that integrates WFs with TSK fuzzy model. The kernel of the fuzzy system is the fuzzy knowledge base that consists of the input–output data points of the system interpreted into linguistic fuzzy rules. The consequent parts of TSK-type fuzzy IF–THEN rules are represented by either a constant or a function. In most fuzzy and neuro-fuzzy models, the function is a linear combination of the input variables plus a constant term. TSK-type systems cannot, in general, model complex processes with desired accuracy using a certain number of rules. These systems do not have localizability, instead model the global features of the process, and their convergence is generally slow. The consequent part of these TSK-type fuzzy networks do not provide full mapping capabilities, and, in the case of modeling of complex nonlinear processes, may require a high number of rules in order to achieve the desired accuracy. Increasing the number of the rules leads to an increase in the number of neurons in the hidden layer of the network. In this paper, the use of wavelet (rather than linear) functions are proposed to improve the computational power of the neuro-fuzzy system. The rules used thus have the following form:

If x_1 is A_{11} and x_2 is A_{12} and ... and x_m is A_{1m}

$$\text{Then } y_1 = \sum_{i=1}^m w_{i1} (1 - z_{i1}^2) e^{-\frac{z_{i1}^2}{2}}$$

If x_1 is A_{21} and x_2 is A_{22} and ... and x_m is A_{2m}

$$\text{Then } y_2 = \sum_{i=1}^m w_{i2} (1 - z_{i2}^2) e^{-\frac{z_{i2}^2}{2}}$$

If x_1 is A_{n1} and x_2 is A_{n2} and ... and x_m is A_{nm}

$$\text{Then } y_n = \sum_{i=1}^m w_{in} (1 - z_{in}^2) e^{-\frac{z_{in}^2}{2}} \quad (3)$$

where x_1, x_2, \dots, x_m are the input variables, y_1, y_2, \dots, y_n are the output variables, A_{ij} is a membership function for i th rule

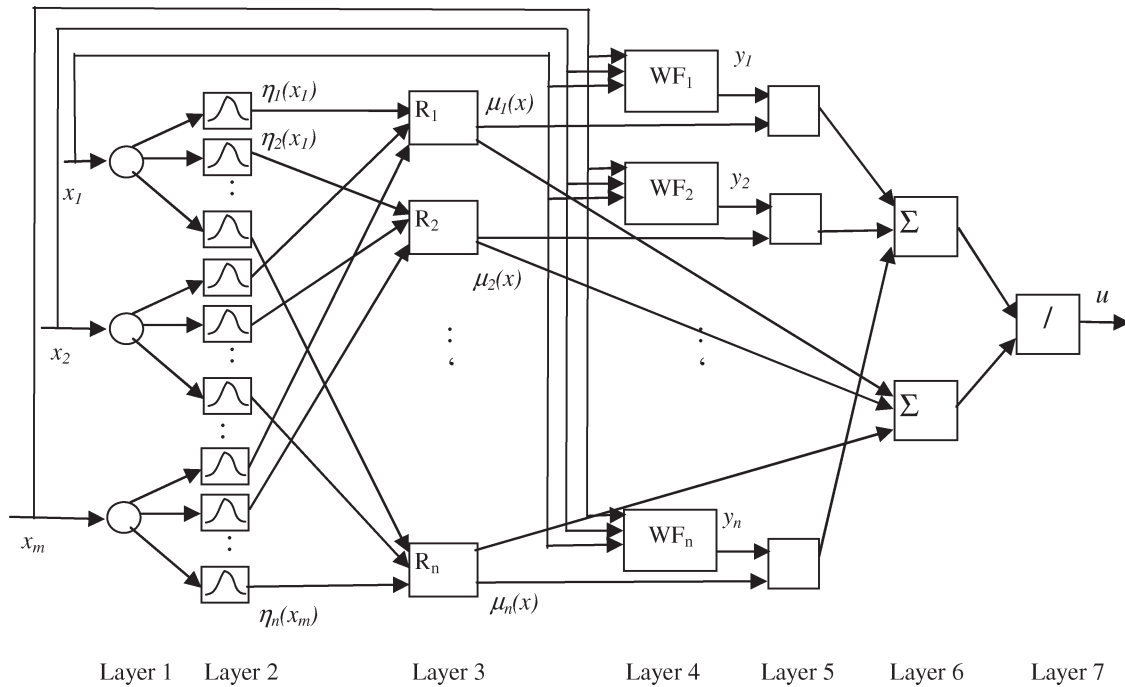


Fig. 2. Structure of FWNN.

of the j th input defined as a Gaussian function. Conclusion parts of the rules contain Mexican Hat WFs (Fig. 1). The use of wavelets with different dilation and translation values allows us to capture different behaviors and the essential features of the nonlinear model under these fuzzy rules. The proper fuzzy model that is described by the set of IF–THEN rules can be obtained by learning the dilation and the translation parameters of the conclusion parts and the parameters of the membership function of the premise parts. Here, because of the use of wavelets, the computational strength and the generalization ability of FWNN are improved, and FWNN can describe the nonlinear process with the desired accuracy.

The structure of FWNN proposed in this paper is depicted in Fig. 2. It includes seven layers. In the first layer, the number of nodes is equal to the number of input signals. These nodes are used for distributing input signals. In the second layer, each node corresponds to one linguistic term. For each input signal entering into the system, the membership degree to the fuzzy set, which that input value belongs to, is calculated. To describe the linguistic terms, the Gaussian membership functions are used

$$\eta_j(x_i) = e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}}, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (4)$$

In (4), m is the number of external input signals, n is the number of fuzzy rules (hidden neurons in third layer), c_{ij} and σ_{ij} are the center and the width of the Gaussian membership functions of the j th term of i th input variable, respectively, and $\eta_j(x_i)$ is the membership function of the i th input variable for the j th term.

In the third layer, the number of nodes correspond to the number of rules R_1, R_2, \dots, R_n . Each node represents one

fuzzy rule. Here, to calculate the values of the output signals of the layer AND (min) operation is used. Below, \prod represents the min operation:

$$\mu_j(x) = \prod_i \eta_j(x_i), \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (5)$$

These $\mu_j(x)$ signals are the input signals for the next layer, which is the consequent layer. It includes n WFs. In the fifth layer, the output signals of the third layer are multiplied by the output signals of the WFs. The output of the l th wavelet is calculated as

$$y_l = w_l \Psi_l(z), \quad \Psi_l(z) = \sum_{i=1}^m |a_{il}|^{-\frac{1}{2}} (1 - z_{il}^2) e^{-\frac{z_{il}^2}{2}}. \quad (6)$$

Here, $z_{il} = (x_i - b_{il})/a_{il}$ and a_{il} and b_{il} are the parameters of the WF between the i th ($i = 1, \dots, n$) input and the l th output of ($l = 1, \dots, n$) the wavelet. In the sixth and the seventh layers, the defuzzification is made to calculate the output of the whole network. In this layer, the contribution of each wavelet to the output of the FWNN is determined

$$u = \frac{\sum_{l=1}^n \mu_l(x) y_l}{\sum_{l=1}^n \mu_l(x)}. \quad (7)$$

Here, y_l are the output signals of the wavelet NNs.

The number of parameters N to be updated in the FWNN structure is given by

$$\begin{aligned} N = & \text{Number of parameters of the Gaussians} \\ & + \text{Number of parameters of the wavelets} \\ & + \text{number of weights } (w). \end{aligned}$$

The number of linguistic terms for each input is equal to the number of fuzzy rules.

In [19]–[24], some fuzzy wavelet structures have been designed. In this paper, the consequent parts of fuzzy rules are computed using (6). In contrast to the other FWNN structures seen in the literature, in this paper, a variable z is used in the wavelet, defined as $z = (\mathbf{x} - \mathbf{b})/\mathbf{a}$, where \mathbf{a} and \mathbf{b} are the parameters of the WF, and \mathbf{x} is input signal of the network. That is to say, the difference between the input signal \mathbf{x} and the Mexican-hat wavelet center is calculated. In the existing literature, e.g., [18], [23], the input signal \mathbf{x} is directly used as a parameter of the wavelet.

The constructed FWNN structure has the following advantages: The FWNN is constructed on the basis of fuzzy rules that include WFs in the consequent parts of the rules. A wavelet is a nonlinear function of input linguistic variables that have local property. The use of such a structure allows the FWNN to approximate complex functions more effectively. Furthermore, the FWNN model has a smaller network size and faster learning speed than other modeling structures as will be demonstrated in this paper.

III. PARAMETER UPDATE RULES FOR LEARNING

In this paper, a gradient-based learning algorithm with adaptive learning rate is adopted. The latter guarantees the convergence and speeds up the learning of the network. In addition, a momentum is used to speed-up the learning process. The parameter to be updated are the parameters of the membership functions in the second layer of the network ($c_{ij}(t)$ and $\sigma_{ij}(t)$; $i = 1, \dots, n$, $j = 1, \dots, m$) and the parameters of wavelets ($a_{il}(t)$, $b_{il}(t)$, $w_l(t)$; $i = 1, \dots, m$, $l = 1, \dots, n$) in the consequent part. The initial values are generated randomly.

At the first step, the value of the following cost function is calculated:

$$E = \frac{1}{2} \sum_{i=1}^O (u_i^d - u_i)^2. \quad (8)$$

Here, O is the number of output signals of the network (in our case, $O = 1$), and u_i^d and u_i are the desired and the current output values of the network, respectively. The parameters w_l , a_{il} , b_{il} , ($i = 1, \dots, m$, $l = 1, \dots, n$) of the wavelet NN and the parameters of the membership functions c_{ij} and σ_{ij} ($i = 1, \dots, m$, $j = 1, \dots, n$) of the neuro-fuzzy structure are adjusted by using the following formulas:

$$\begin{aligned} w_l(t+1) &= w_l(t) - \gamma \frac{\partial E}{\partial w_l} + \lambda (w_l(t) - w_l(t-1)) \\ a_{il}(t+1) &= a_{il}(t) - \gamma \frac{\partial E}{\partial a_{il}} + \lambda (a_{il}(t) - a_{il}(t-1)) \\ b_{il}(t+1) &= b_{il}(t) - \gamma \frac{\partial E}{\partial b_{il}} + \lambda (b_{il}(t) - b_{il}(t-1)) \\ c_{ij}(t+1) &= c_{ij}(t) - \gamma \frac{\partial E}{\partial c_{ij}}; \quad \sigma_{ij}(t+1) = \sigma_{ij}(t) - \gamma \frac{\partial E}{\partial \sigma_{ij}} \end{aligned} \quad (9)$$

$$(10)$$

where γ is the learning rate, λ is the momentum, m is the number of input signals of the network (input neurons), and n is the number of rules (the hidden neurons). $i = 1, \dots, m$, $j = 1, \dots, n$, and $l = 1, \dots, n$.

The values of derivatives in (9) can be calculated by the following formulas:

$$\begin{aligned} \frac{\partial E}{\partial w_l} &= \frac{\partial E}{\partial u} \frac{\partial u}{\partial y_l} \frac{\partial y_l}{\partial w_l} = (u(t) - u^d(t)) \cdot \mu_l \cdot \psi(z_l) \Big/ \sum_{i=1}^n \mu_i \\ \frac{\partial E}{\partial a_{il}} &= \frac{\partial E}{\partial u} \frac{\partial u}{\partial y_l} \frac{\partial y_l}{\partial \psi_l} \frac{\partial \psi_l}{\partial z_{il}} \frac{\partial z_{il}}{\partial a_{il}} \\ &= \delta_l (3.5z_{il}^2 - z_{il}^4 - 0.5) e^{-\frac{z_{il}^2}{2}} \Big/ \left(\sqrt{a_{il}^3} \right) \\ \frac{\partial E}{\partial b_{il}} &= \frac{\partial E}{\partial u} \frac{\partial u}{\partial y_l} \frac{\partial y_l}{\partial \psi_l} \frac{\partial \psi_l}{\partial z_{il}} \frac{\partial z_{il}}{\partial b_{il}} \\ &= \delta_l (3z_{il} - z_{il}^3) e^{-\frac{z_{il}^2}{2}} \Big/ \left(\sqrt{a_{il}^3} \right) \end{aligned} \quad (11)$$

where $\delta_i = (\partial E/\partial u)(\partial u/\partial y_l)(\partial y_l/\partial \psi_l) = (u(t) - u^d(t)) \cdot \mu_l \cdot w_l / \sum_{i=1}^n \mu_i$, $i = 1, \dots, m$, $l = 1, \dots, n$.

The derivatives in (10) are determined by the following formulas:

$$\frac{\partial E}{\partial c_{ij}} = \sum_j \frac{\partial E}{\partial u} \frac{\partial u}{\partial \mu_j} \frac{\partial \mu_j}{\partial c_{ij}}; \quad \frac{\partial E}{\partial \sigma_{ij}} = \sum_j \frac{\partial E}{\partial u} \frac{\partial u}{\partial \mu_j} \frac{\partial \mu_j}{\partial \sigma_{ij}} \quad (12)$$

$$\frac{\partial E}{\partial u} = u(t) - u^d(t); \quad \frac{\partial u}{\partial \mu_j} = \frac{y_j - u}{\sum_{j=1}^n \mu_j} \quad (13)$$

$$\frac{\partial \mu_j(x_i)}{\partial c_{ij}} = \begin{cases} \mu_j(x_i) \frac{2(x_i - c_{ij})}{\sigma_{ij}^2}, & \text{if } i \text{ node is connected} \\ 0, & \text{to rule node } j \\ & \text{otherwise} \end{cases} \quad (14)$$

$$\frac{\partial \mu_j(x_i)}{\partial \sigma_{ij}} = \begin{cases} \mu_j(x_i) \frac{2(x_i - c_{ij})^2}{\sigma_{ij}^3}, & \text{if } i \text{ node is connected} \\ 0, & \text{to rule node } j \\ & \text{otherwise.} \end{cases} \quad (15)$$

Here, $i = 1, \dots, m$, $j = 1, \dots, n$. Using (11)–(15), the derivatives in (9) and (10) are calculated and an update of the parameters of the FWNN is carried out.

One of the important problems in learning algorithms is the convergence. The convergence of the gradient descent method depends on the selection of the initial values of the learning rate and the momentum term. Usually, these values are selected in the interval [0–1]. A large value of the learning rate may lead to nonstable learning; a small value of learning rate leads to slow learning speed. In this paper, an adaptive approach is used for updating these parameters. That is, the learning of NN parameters is started with a small value of learning rate γ . During learning, the value of γ is increased if the value of change of error $\Delta E = E(t) - E(t+1)$ is positive, and the learning rate is decreased if the change in the error $\Delta E = E(t) - E(t+1)$ is negative. This strategy leads to stable learning of the FWNN, guarantees the convergence, and speeds up the learning of

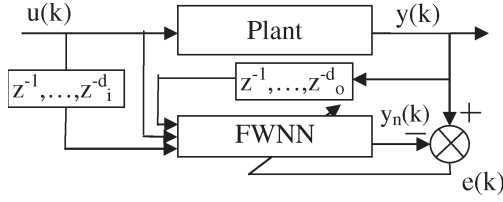


Fig. 3. Identification scheme.

FWNN. In addition, a momentum term is used to speedup learning processes.

IV. SIMULATION STUDIES

In order to evaluate the performance of the proposed structure, a number of simulation studies are carried out for both identification and control purposes; the plant models are taken from literature in order to be able to make a direct performance comparison.

A. Identification Performance Studies

The identification problem involves finding the relation between the input and the output of the system. In Fig. 3, the structure of the identification scheme is shown. The inputs to the FWNN identifier are the external input signals, its one-, ..., d_i —step delayed values and the one-, two-, ..., d_o —step delayed outputs of the plant. Here, the problem is to find such values of parameters of FWNN by using them in the system for all input values of $u(k)$ the difference between plant output $y(k)$ and network output $y_n(k)$ will be minimum. Here, $y(k)$ is plant output, $y_n(k)$ is output of FWNN system.

Example 1: As an example, identification of a nonlinear plant using FWNN is considered. The plant is described as

$$y(k) = 0.72y(k - 1) + 0.025y(k - 2)u(k - 1) + 0.01u^2(k - 2) + 0.2u(k - 3). \quad (16)$$

The mathematical model of dynamic plant is the same as that used in [26]. As can be seen, the current output of plant depends on previous input and output signals. However, only the current state of system and the control signal are fed into the FWNN as inputs. The identification is performed for two cases. In the first case, three fuzzy rules and, in the second case, five fuzzy rules are used in FWNN structure. The gradient descent algorithm is applied for learning of the parameter values of FWNN. The initial values of the parameters of the FWNN are generated in the interval $[-1, 1]$. For the identification of the plant, the following excitation signal is used:

$$u(k) = \begin{cases} \sin(\pi k/25), & k < 250 \\ 1.0, & 250 \leq k < 500 \\ -1.0, & 500 \leq k < 750 \\ 0.3 \sin(\pi k/25) + 0.1 \sin(\pi k/32) \\ \quad + 0.6 \sin(\pi k/10), & 750 \leq k < 1000. \end{cases} \quad (17)$$

Using the parameter update rules derived above, the learning of the parameter values of FWNN for the given input signals is

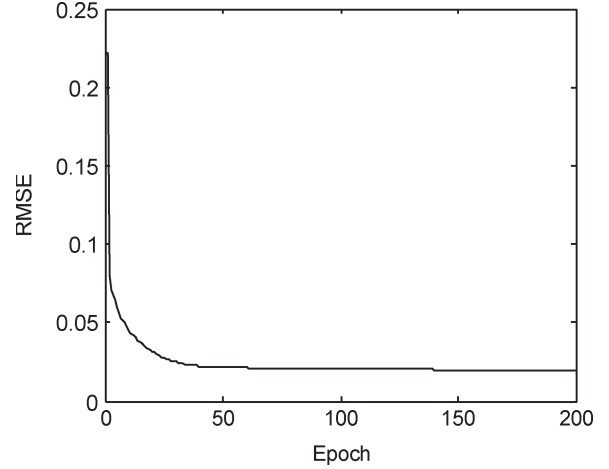


Fig. 4. RMSE values obtained during learning.

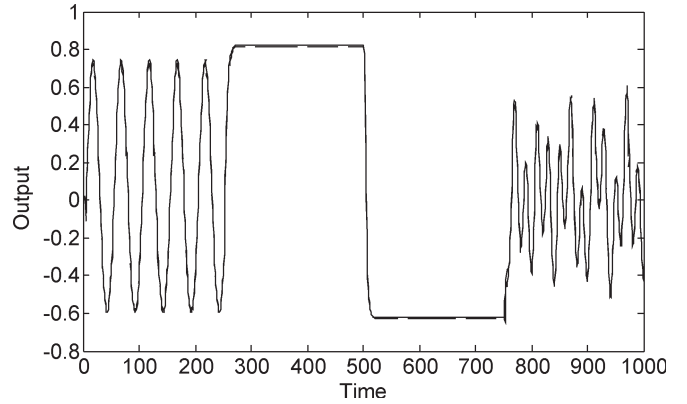


Fig. 5. Results of identification, where solid line denotes the output of the plant, dashed line denotes the FWNN output.

carried out. The training is continued for 200 epochs with 1000 time steps in each epoch. As a performance criterion, the root mean square error (RMSE) given in (18) is used with $K = 1000$

$$J = \sqrt{\frac{\sum_{i=1}^k (y(k + 1) - y_N(k + 1))^2}{K}}. \quad (18)$$

As a result of training, three fuzzy rules are generated, and the other parameters of the FWNN are determined. When the number of parameters is $N = 27$ and, the value of RMSE of identification obtained after training was 0.019736. RMSE value of FWNN for test data was 0.022609. When number of parameters is taken 43, the RMSE value of FWNN-based identification system for training and test data were 0.018713 and 0.020169, respectively. Fig. 4 shows the evolution of the RMSE values over 200 epochs. Fig. 5 compares the actual plant output with that of the FWNN identifier. In Table I, the aforementioned RMSE values are given together with the values reported in the literature for Elman’s recurrent NN (ERNN) [27], recurrent self-organizing neural fuzzy inference network (RSONFIN) [28] and TSK-type fuzzy network with supervised learning (TRFN-S) [9] models for the same plant and with the same excitation signal. As can be seen the RMSE value in FWNN model is less than the other models, despite the considerably smaller number of parameters to be learned.

TABLE I
SIMULATION RESULTS OF DIFFERENT MODELS
FOR DYNAMIC PLANT IDENTIFICATION

Models	Network Parameters	RMSE	
		train	test
ERNN[27]	54	0.036	0.078
RSONFIN [28]	49	0.03	0.06
TRFN-S [9]	33	0.0067	0.0313
FWNN	27	0.019736	0.022609
	43	0.018713	0.020169

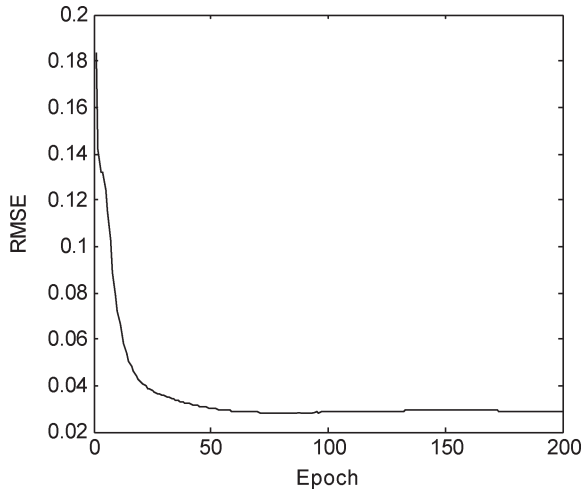


Fig. 6. RMSE values obtained during learning.

Example 2: This example considers the second-order non-linear plant that has been used in [9], [26]. The process is described by the following difference equation:

$$y(k + 1) = f(y(k), y(k-1)y(k-2), u(k), u(k-1)). \quad (19)$$

From above

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_2^2} \quad (20)$$

and $y(k - 1)$, $y(k - 2)$, $y(k - 3)$ are one-, two- and three-step delayed outputs of the plant, respectively, $u(k)$ and $u(k - 1)$ are current and one-step delayed inputs of the plant, respectively. The identification of the same plant by using different models are considered in [9] and [28]. For the simulation studies, the current state of system and the control signal are used as the one used for the previous example.

As before, the values of the FWNN parameters are initialized in the interval $[-1, 1]$ and updated by the parameter update rules derived. After training three fuzzy rules are generated. Figs. 6 and 7 show the performance of the FWNN, and Table II compares the RMSE values with the other approaches reported in the literature, namely recurrent fuzzy NN, RSONFIN [28], feedforward neural fuzzy system, and TRFN-S [9].

B. Control Performance Studies

In this section, the performance of the proposed FWNN is investigated when used for control purposes. The structure of the FWNN-based control system is as shown in Fig. 8. Here,

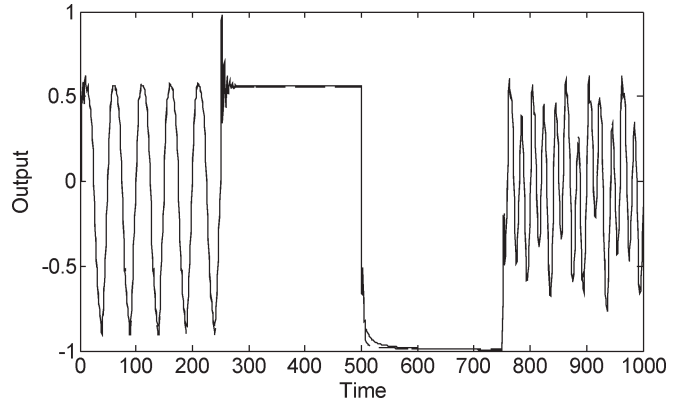


Fig. 7. Results of identification, where solid line denotes the output of the plant, dashed line denotes the FWNN output.

TABLE II
SIMULATION RESULTS OF DIFFERENT MODELS FOR
DYNAMIC PLANT IDENTIFICATION

Models	Network Parameters	RMSE	
		train	test
RFNN	112	0.0114	0.0575
RSONFIN [28]	36	0.0248	0.0780
Feedforward neural fuzzy system	48	0.0203	0.0521
TRFN-S [9]	33	0.0084	0.0346
FWNN	27	0.029179	0.031212
	43	0.028232	0.030125

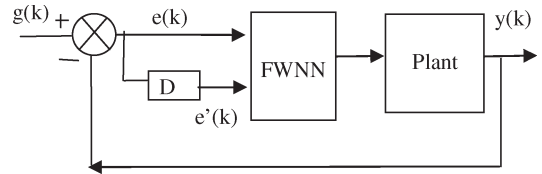


Fig. 8. Structure of FWNN-based control system.

$y(k)$ is the output signal of the plant, $g(k)$ is the set-point signal, $e(k)$ and $e'(k)$ are the error and the change in error, respectively. D represents the difference of error. Using these signals, the learning of the parameters of the FWNN structure is carried out in a closed-loop fashion, and thus, the IF-THEN rules of the controller is generated. The consequent parts of the rules result in the control signal to be applied to the plant.

Example 3: In this example, the proposed FWNN structure is used for the control of the dynamic described by the following difference equation:

$$y(k) = \frac{y(k - 1)y(k - 2) (y(k - 1) + 2.5)}{(1 + y(k - 1)^2 + y(k - 2)^2)} + u(k). \quad (21)$$

The FWNN structure has 27 parameters to be updated. The initial values of the parameters of FWNNs are generated randomly, in the interval $[-10, 10]$. The three fuzzy rules are used in FWNN structure. The parameters of the FWNN are updated using different set-point signals. The RMSE value, which is calculated by (18), is taken as the performance criterion. The number of data points used for training is 200. The training of FWNN system is performed for 200 data points. Fig. 9 shows the evolution of the RMSE values over 200 epochs.

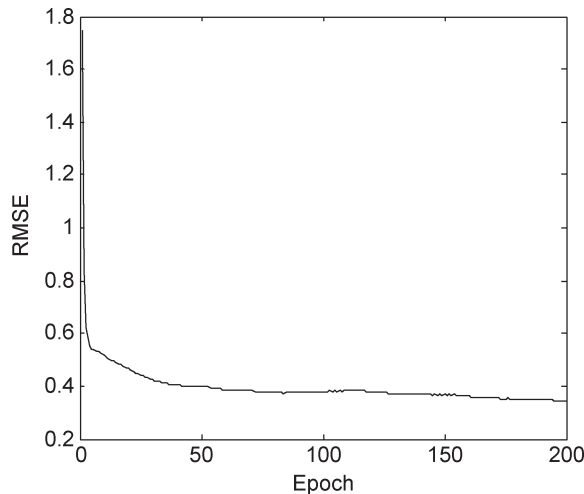


Fig. 9. RMSE values obtained during learning.

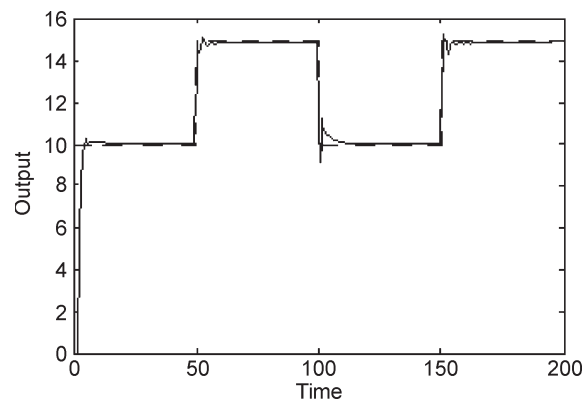


Fig. 10. Time response characteristic of control systems with FWNN for different values of set-point input signals. Dashed line is set-point signal, solid line is plant output.

TABLE III
SIMULATION RESULTS OF FWNN-BASED CONTROL OF DYNAMIC PLANT

Models	Network Parameters	RMSE	
		Mean	Best
ERNN+GA[27]	40	1.504	1.439
TRFN-G[9]	33	1.086	0.887
FWNN	27	0.351592	0.31812

In Fig. 10, the time response characteristics of the control system are shown. The best and the averaged RMSE values over 200 time steps are given in Table III. This performance is compared with the other methods proposed in the literature, namely ERNN with genetic algorithm (ERNN+GA) [27] and TRFN with genetic learning (TRFN-G) [9]. As can be seen, the RMSE value for the FWNN model is less than that of the other approaches.

Example 4: As the last example the FWNN-based control system is used for the control of the dynamic plant (16) given in Example 1. The parameters of the FWNN are updated using different set-point signals for 200 data points. The simulation studies carried out are similar to those in Example 3. Figs. 11 and 12 show the performance of the FWNN, and Table IV compares the best and averaged RMSE values with the other approaches. As can be seen, the performance of the FWNN

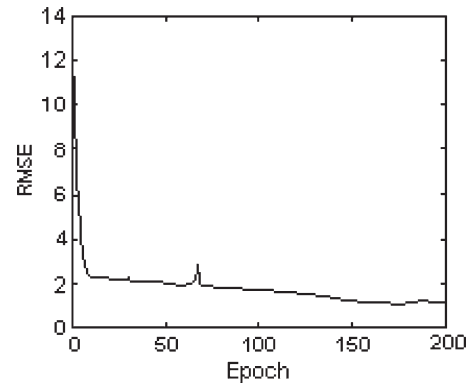


Fig. 11. RMSE values obtained during learning.

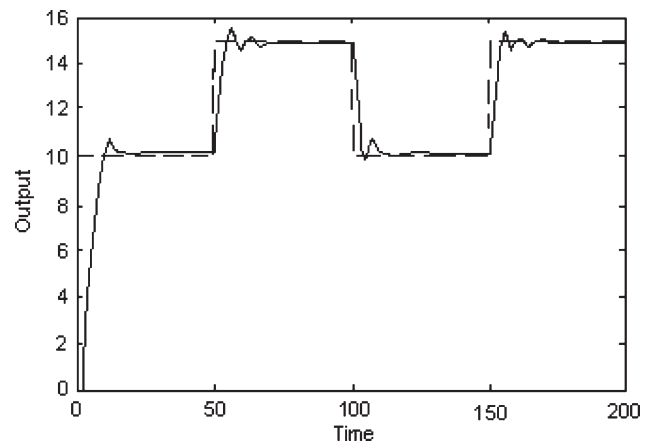


Fig. 12. Time response characteristic of control systems with FWNN for different values of set-point input signals. Dashed line is set-point signal, solid line is plant output.

TABLE IV
SIMULATION RESULTS OF FWNN-BASED CONTROL OF DYNAMIC PLANT

Models	Network Parameters	RMSE	
		Mean	Best
ERNN+GA[27]	40	1.5046	1.4394
TRFN-G[9]	33	1.2353	1.1374
FWNN	27	1.2043	1.13402

is significantly better than those that can be obtained with ERNN+GA [27] and TRFN-G [9].

V. CONCLUSION

In this paper, a novel adaptive structure is proposed for identification and control of dynamic plants that combines the advantages of WF, NNs and fuzzy logic. The parameter update rules of the structure are derived based on the gradient descent algorithm. Several simulation studies are carried out for both identification and control purposes. The plant models are taken from the literature to enable a direct performance comparison. It is seen that FWNN can converge faster and is more adaptive to new data. In both the identification and the control cases, the performance is much better, resulting in smaller RMSE values, despite the smaller number of parameters. Future works on the topic includes the use of genetic algorithms for the optimization

of the structure of the FWNN and the extending of the research for the control of uncertain plants.

REFERENCES

- [1] R. R. Yager and L. A. Zadeh, Eds., *Fuzzy Sets, Neural Networks and Soft Computing*. New York: Van Nostrand Reinhold, 1994.
- [2] J.-S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Englewood Cliffs, NJ: Prentice-Hall, 1997, ch. 17.
- [3] N. Dettlef and K. Rudolf, "Designing neuro-fuzzy systems through backpropagation," in *Fuzzy Modelling: Paradigms and Practice*, W. Pedryz, Ed. Boston, MA: Kluwer, 1996, pp. 203–228.
- [4] M. O. Efe and O. Kaynak, "On stabilization of gradient-based training strategies for computationally intelligent systems," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 564–575, Oct. 2000.
- [5] V. Topalov, G. L. Cascella, V. Giordano, F. Cupertino, and O. Kaynak, "Sliding mode neuro-adaptive control of electrical drives," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 671–679, Feb. 2007.
- [6] F. Da and W. Song, "Fuzzy neural networks for direct adaptive control," *IEEE Trans. Ind. Electron.*, vol. 50, no. 3, pp. 507–513, Jun. 2003.
- [7] M. J. Er and Y. Gao, "Robust adaptive control of robot manipulators using generalized fuzzy neural networks," *IEEE Trans. Ind. Electron.*, vol. 50, no. 3, pp. 620–628, Jun. 2003.
- [8] C.-H. Lee and C.-C. Cheng, "Identification and control of dynamic systems using recurrent fuzzy neural network," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 4, pp. 349–366, Aug. 2000.
- [9] C.-F. Juang, "A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 155–170, Apr. 2002.
- [10] J. Nunez-Garcia and O. Wolkenhauer, "Random set system identification," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 3, pp. 287–296, Oct. 2002.
- [11] T. Kugarajah and Q. Zhang, "Multidimensional wavelet frames," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1552–1556, Nov. 1995.
- [12] Q. Zhang and A. Benviste, "Wavelet networks," *IEEE Trans. Neural Netw.*, vol. 3, no. 6, pp. 889–898, Nov. 1992.
- [13] H. Szu, B. Telfer, and J. Garcia, "Wavelet transforms and neural networks for compression and recognition," *Neural Netw.*, vol. 9, no. 4, pp. 695–708, Jun. 1996.
- [14] J. Zhang, G. G. Walter, Y. Miao, and W. N. W. Lee, "Wavelet neural networks for function learning," *IEEE Trans. Signal Process.*, vol. 43, no. 6, pp. 1485–1497, Jun. 1995.
- [15] T. Q. D. Khao, L. M. Phuong, P. T. T. Binh, and N. T. H. Lien, "Application of wavelet and neural network to long-term load forecasting," in *Proc. POWERCON*, Singapore, 2004, pp. 840–844.
- [16] R.-J. Wai, R.-Y. Duan, J.-D. Lee, and H.-H. Chang, "Wavelet neural network control for induction motor drive using sliding-mode design technique," *IEEE Trans. Ind. Electron.*, vol. 50, no. 4, pp. 733–748, Aug. 2003.
- [17] R.-J. Wai and J.-M. Chang, "Implementation of robust wavelet-neural-network sliding-mode control for induction servo motor drive," *IEEE Trans. Ind. Electron.*, vol. 50, no. 6, pp. 1317–1334, Dec. 2003.
- [18] C. K. Lin and S. D. Wang, "Fuzzy modelling using wavelet transform," *Electron. Lett.*, vol. 32, no. 24, pp. 2255–2256, Nov. 1996.
- [19] M. Thuillard, "Fuzzy logic in the wavelet framework," in *Proc. Toolmet*, Oulu, Finland, Apr. 13–14, 2000.
- [20] M. Thuillard, *Wavelets in Softcomputing*. Singapore: World Scientific, 2001.
- [21] Q.-J. Guo, H.-B. Yu, and A.-D. Xu, "Wavelet fuzzy network for fault diagnosis," in *Proc. Int. Conf. Commun. Circuits Syst.*, 2005, pp. 993–998.
- [22] Y. Lin and F.-Y. Wang, "Predicting chaotic time-series using adaptive wavelet-fuzzy inference system," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 888–893.
- [23] D. W. C. Ho, P.-A. Zhang, and J. Xu, "Fuzzy wavelet networks for function learning," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 1, pp. 200–211, Feb. 2001.
- [24] R. H. Abiyev, "Controller based of fuzzy wavelet neural network for control of technological processes," in *Proc. IEEE Int. CISMA*, Giardini Naxos, Italy, 2005, pp. 215–219.
- [25] R. H. Abiyev, "Time series prediction using fuzzy wavelet neural network model," in *Lecture Notes in Computer Sciences*, vol. 4132. Berlin, Germany: Springer-Verlag, 2006, pp. 191–200.
- [26] K. S. Narendra and K. Parthasarathy, "Identification and control dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.
- [27] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, 1990.
- [28] C. F. Juang and C. T. Lin, "A recurrent self-organizing neural fuzzy inference network," *IEEE Trans. Neural Netw.*, vol. 10, no. 4, pp. 828–845, Jul. 1999.



Rahib Hidayat Abiyev (M'02) received the Ph.D. degree in electrical and electronic engineering from Azerbaijan State Oil Academy, Baku, Azerbaijan, in 1997.

He worked as a Research Assistant with the research laboratory "Industrial intellectual control systems." Since 1999, he has been working as an Associate Professor with the Department of Computer Engineering, Near East University, Lefkosa, North Cyprus. He is Vice Chairman of the Computer Engineering department. His research interests are

softcomputing, control systems, pattern recognition, and signal processing. He has published more than 100 research publications in the related subjects.



Okyay Kaynak (M'80–SM'90–F'03) received the B.Sc. (with first class honors) and Ph.D. degrees in electronic and electrical engineering from the University of Birmingham, Birmingham, U.K., in 1969 and 1972, respectively.

From 1972 to 1979, he held various positions within the industry. Since 1979, he was with the Department of Electrical and Electronics Engineering, Bogazici University, Istanbul, Turkey, where he is currently a Full Professor, holding the UNESCO Chair on Mechatronics. He has held long-term (near

to or more than a year) Visiting Professor/Scholar positions at various institutions in Japan, Germany, U.S., and Singapore. His current research interests are in the fields of intelligent control and mechatronics. He has authored three books and edited five, and authored or coauthored more than 200 papers that have appeared in various journals and conference proceedings.

Dr. Kaynak is active in international organizations, has served on many committees of IEEE, and was the President of IEEE Industrial Electronics Society from 2002 to 2003.