2008 IEEE International Symposium on Intelligent Control
Part of 2008 IEEE Multi-conference on Systems and Control
San Antonio, Texas, USA, September 3-5, 2008

Identification and Control of Dynamic Plants Using Fuzzy Wavelet Neural Networks

Rahib H.Abiyev, *Member, IEEE*, Okyay Kaynak, *Fellow Member, IEEE*

***Abstract*— This paper presents a Fuzzy Wavelet Neural Network (FWNN) for identification and control of a dynamic plant. The FWNN is constructed on the basis of fuzzy rules that incorporate wavelet functions in their consequent parts. The architecture of the control system is presented and the parameter update rules of the system are derived. Learning rules are based on the gradient decent method and Genetic Algorithm (GA). The structure is tested for the identification and the control of the dynamic plants commonly used in the literature. It is shown that the proposed structure results in a better performance despite its smaller parameter space.**

I. Introduction

RECENTLY the soft computing methodologies such as fuzzy logic, neural networks, genetic algorithms are used to solve the control problems of dynamic systems 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. Fuzzy technology is an effective tool for dealing with complex, nonlinear processes that are characterized with ill-defined and uncertain factors. The rule base of such fuzzy systems is usually created using 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, neural networks and wavelet technology are used to solve identification and control of dynamic systems.

Numerous different neural and fuzzy structures are proposed for solving identification and control problems [2-10] and their parameter update algorithms are given. A well known structure is the Adaptive Neuro-Fuzzy Inference system (ANFIS) [2], another one is known as NEFCON (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 neural networks 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 neural networks. In [8] a recurrent fuzzy network is used for nonlinear modelling. In [9] a TSK-type recurrent neuro-fuzzy neural network (TRFN) is developed and in [10] the use of random set theory in a fuzzy scheme for identification of dynamic plants is considered.

Most NN structures seen in literature use the sigmoid activation function in neurons. However, the sigmoid function is not orthogonal and its energy is infinite, and this leads to a slow convergence speed. A relatively less common structure is the one that use wavelet functions. Wavelet is local function that has limited duration. A wavelet neural network (WNN) has a nonlinear regression structure that use basis functions in the hidden layer to achieve input-output mappings. 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 modelling [11,12] and some researchers [13-16] have used such structures for solving approximation, classification, prediction, and control problems.

A fuzzy wavelet neural network (FWNN) combines wavelet theory with fuzzy logic and neural networks. 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 [17-24] have used a combination of fuzzy technology and WNN for solving signal processing and control problems. In [17] a fuzzy system with a linear combination of basis function is proposed and in [18-20] the wavelet network model of a fuzzy inference system is proposed. In [18] 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 [21]. The use of such multilayer structures complicates the architecture of the system. A FWNN structure that is constructed on the base of a set of fuzzy rules is proposed in [22] and used for the approximation nonlinear functions. Other wavelet based approaches include FWNN structures developed for control

Manuscript received November 25, 2007.
Rahib H. Abiyev is with the the Department of Computer Engineering, Near East University, Lefkosa, North Cyprus (e-mail: rahib@neu.edu.tr).
Okyay Kaynak is with the Department of Electrical and Electronic Engineering, Bogazici University, Turkey. (e-mail: okyay.kaynak@boun.edu.tr).

of dynamic plants [23] and for time-series prediction [24].

Wavelet transform has the ability to analyze non-stationary signals to discover their local details. Fuzzy logic allows to reduce the complexity of the data and to deal with uncertainty. NNs have self-learning capability that increases the accuracy of the model. Their combination allows us to develop a system with fast learning capability that can describe nonlinear systems that are characterized with uncertainties. 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 7-layer structure is explained. In Sec. 3, the parameter update rules based on the gradient descent method are derived, the GA learning is described. In Sec. 4, the simulation studies are presented for both identification and control cases.

II. Fuzzy Wavelet Neural Network

Wavelets are defined in the following form

$$\Psi_j(\mathbf{x}) = |\mathbf{a_j}|^{-\frac{1}{2}} \psi\left(\frac{\mathbf{x}-\mathbf{b_j}}{\mathbf{a_j}}\right), \quad \mathbf{a_j} \neq 0 \tag{1}$$

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

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

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

Here $\Psi_j(\mathbf{x})$ is the wavelet function 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 wavelet function 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 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 a fuzzy wavelet neural network that integrates wavelet functions with Takagi-Sugeno-Kang (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 a linear function is used. In the case of modeling of complex non-linear processes a high number of rules may be required 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_{l1} and x_2 is A_{l2} and … and x_m is A_{lm}

$$\text{THEN } y_l \text{ is } \sum_{i=1}^{m} w_{il}(1 - z_{il}^2) e^{-\frac{z_{il}^2}{2}} \tag{3}$$

In above, $x_1, x_2, …, x_m$ are the input variables, $y_1, y_2, …, y_n$ are the output variables, A_{ij} is a membership function for i-th term of the j-th input defined as a Gaussian function. l is the number of rules. Conclusion parts of the rules contain Mexican Hat wavelet functions. The use of wavelets with different dilation and translation values allows us to capture different behaviours 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 is improved, and, FWNN can describe the nonlinear process with the desired accuracy.

The structure of fuzzy wavelet neural network proposed in this paper is depicted in Fig. 1. 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.

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

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 $\mu 1_j(x_i)$ 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 $R_1, R_2, \ldots, R_n$. Each node represents one fuzzy rule. Here to calculate the values of the output signals of the layer AND (min) operation is used. In below, Π represents the min operation.

$$\mu_j(x) = \prod_i \mu 1_j(x_i), \quad i=1,..,m,\ j=1,..,n \tag{5}$$

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

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

Here $z_{il} = \dfrac{x_i - b_{il}}{a_{il}}$ and a_{il} and b_{il} are the parameters of the wavelet function between the *i*-th (*i=1,..,n)* input and the *l*-th output of (*l=1,..,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.

$$u = \sum_{l=1}^{n} \mu_l(x) y_l \Big/ \sum_{l=1}^{n} \mu_l(x) \tag{7}$$

In above, y_l are the output signals of the wavelet neural networks.

The number of parameters N to be updated in the FWNN structure is determined by the number of parameters of the Gaussians, the number of parameters of the wavelets and the number of weights (w).

In [17-24], some fuzzy wavelet structures have been designed. In this paper, the consequent parts of fuzzy rules are computed using formula (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=(x-b)/a***, where ***a*** and ***b*** are the parameters of the wavelet function, and ***x*** is input signal of the network. That is to say the difference between the input signal ***x*** and the Mexican-hat wavelet centre is calculated. In the existing literature, e.g. [17,22], the input signal ***x*** is directly used as a parameter of the wavelet.

In the synthesis of a FWNN, there are two steps; one is the decision on its structure, the second is related to finding the optimal values of the parameters of FWNN. The former can be named as structure learning and the latter as parameter learning. The number of rules defines the structure of the FWNN. Here the main problem is the determination of the number of fuzzy rules. The relation between the input and output data set is described by these rules. In this work, a clustering approach is used, the details of which will not be given due to lack of space. Each rule corresponds to one cluster. Defining the cluster centers for the input data set leads to the definition of the rules for the control system. The input data with higher firing levels are located near to the cluster centers, the data with low firing level are located far from the cluster centre. After defining the structure, the learning of the FWNN parameters is performed.

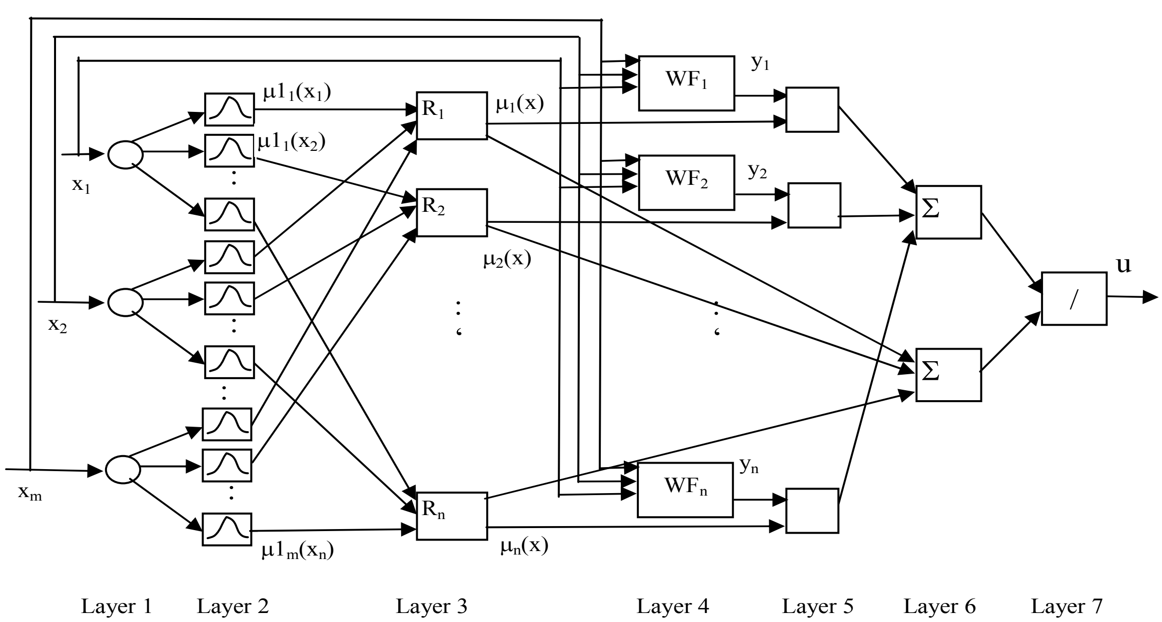

Fig. 1. Structure of FWNN

III. LEARNING

In this paper, two different approaches are used for updating the parameters of the FWNN, namely a gradient based approach and a genetic algorithm based approach.

A. Gradient Based Approach

In the case of the identification problem described below, a gradient based learning algorithm with adaptive learning rate is adopted for parameter updating. 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 parameters to be updated are the parameters of the membership functions in the second layer of the network ($c_{ij}(t)$ and $\sigma_{ij}(t)$; $i=1,..,n,\ \ j=1,..,m$) and the parameters of wavelets ($a_{il}(t)$, $b_{il}(t)$,$w_l(t)$; $i=1,..,m,\ \ l=1,..,n$) in the consequent part. The initial values are generated randomly.

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

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

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_l (i=1,..,m, l=1,..,n) of the wavelet neural network and the parameters of the membership functions c_{ij} and σ_{ij} (i=1,..,m, j=1,..,n) of the neuro-fuzzy structure are adjusted by using the following formulas.

$$w_l(t+1) = w_l(t) + \gamma\frac{\partial E}{\partial w_l} + \lambda(w_l(t) - w_l(t-1))$$
$$a_{il}(t+1) = a_{il}(t) + \gamma\frac{\partial E}{\partial a_{il}} + \lambda(a_{il}(t) - a_{il}(t-1)) \qquad (9)$$
$$b_{il}(t+1) = b_{il}(t) + \gamma\frac{\partial E}{\partial b_{il}} + \lambda(b_{il}(t) - b_{il}(t-1))$$

$$c_{ij}(t+1) = c_{ij}(t) + \gamma\frac{\partial E}{\partial c_{ij}} + \lambda(c_{ij}(t) - c_{ij}(t-1))$$
$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \gamma\frac{\partial E}{\partial \sigma_{ij}} + \lambda(\sigma_{ij}(t) - \sigma_{ij}(t-1)) \qquad (10)$$

Here 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).

The values of derivatives in (9) and (10) can be calculated by the following formulas.

$$\frac{\partial E}{\partial w_l} = (u(t) - u^d(t))\cdot\mu_l\cdot\psi(z_l)\Big/\sum_{l=1}^{n}\mu_l$$
$$\frac{\partial E}{\partial a_{il}} = \delta_l(3.5z_{il}^2 - z_{il}^4 - 0.5)e^{-\frac{z_{il}^2}{2}}\Big/(\sqrt{a_{il}^3}), \qquad (11)$$
$$\frac{\partial E}{\partial b_{il}} = \delta_l(3z_{il} - z_{il}^3)e^{-\frac{z_{ijl}^2}{2}}\Big/(\sqrt{a_{il}^3})$$

$$\frac{\partial E}{\partial c_{ij}} = \sum_j \frac{\partial \mathrm{E}}{\partial u}\frac{\partial u}{\partial \mu_l}\frac{\partial \mu_l}{\partial c_{ij}},\ \frac{\partial E}{\partial \sigma_{ij}} = \sum_j \frac{\partial \mathrm{E}}{\partial u}\frac{\partial u}{\partial \mu_l}\frac{\partial \mu_l}{\partial \sigma_{ij}}$$
$$\frac{\partial E}{\partial u} = u(t) - u^d(t),\ \ \frac{\partial u}{\partial \mu_l} = (y_l - u)\Big/\sum_{l=1}^{L}\mu_l \qquad (12)$$

$$\frac{\partial \mu_l(x_j)}{\partial c_{ji}} = \begin{cases} \mu_l(x_j)\dfrac{2(x_j - c_{ji})}{\sigma_{ji}^2} & if\ \ j \text{ node is connected to rule node } l \\ 0, & \text{otherwise} \end{cases} \qquad (13)$$

$$\frac{\partial \mu_l(x_j)}{\partial \sigma_{ji}} = \begin{cases} \mu_l(x_j)\dfrac{2(x_j - c_{ji})^2}{\sigma_{ji}^3} & if\ \ j \text{ node is connected to rule node } l \\ 0, & \text{otherwise} \end{cases} \qquad (14)$$

Here $\delta_l = (u(t) - u^d(t))\cdot\mu_l\cdot w_l\Big/\sum_{l=1}^{n}\mu_l$, $i=1,..,m$, $l=1,..,n$, $j=1,..,n$ Using equations (11-14) the derivatives in (9) and (10) are calculated and an update of the parameters of the FWNN is carried out.

B. Learning using GA

When no training data is available or supervised training is computationally very demanding GA approach is used. Dynamic system control is one such problem. At the same time, sometimes the network learning using the gradient method for nonlinear processes has "local minima" problem and could not find a global optimal solution. Genetic Algorithm (GA) is effective optimization technique that can be used to improve training of the FWNN and avoid "local minima" problem. In the dynamic system control problem discussed below, a real coded GA is used for learning, the number of chromosomes, which is the population size, being generated randomly. These chromosomes consist of genes that represent the network parameters, i.e. the parameters of the membership functions and the parameters of the wavelets of FWNN.

GA operators are applied for training the parameters of the FWNN. The main operations in GA are selection, crossover and mutation. The aim of the selection is to give more reproductive chances to population members (or solutions) that have higher fitness values. Crossover and mutation are two main components in the reproduction process in which selected pairs mate to produce the next generation. The purpose of crossover and mutation is to give the next generation of solutions chances to differ from their parental solutions. It gives children chances to differ from their parents, and hope that some of the children can be closer to the optimal destination than their parents. The tournament selection is applied for obtaining of new generation. In this method two members of the population are selected and their fitness values are compared. The member with the higher fitness is selected for the next generation.

The real coded multipoint crossover operation is used for correction of individuals. According to the crossover rate the individuals are selected for the crossover operation in order to generate the new solution. A high value of crossover rate leads to a quick generation of the new solution. The typical value of crossover rate is in the interval 0.5-1. After a crossover operation two parent members $X=(x_1\ x_2 \dots x_n)$ and $Y=(y_1\ y_2\ \dots\ y_n)$ will have the following form. $X'=(x'_1, x'_2, ..., x'_n)$ and $Y'=(y'_1, y'_2, ..., y'_n)$. The crossover operation is performed using the following formula.

$$x_1^n = x_1^o + \delta(y_1^o - x_1^o); \quad y_1^n = y_1^o + \delta(x_1^o - y_1^o) \quad (15)$$

when $F(x_1)>F(y_1)$. The σ is changed between 0 and 0.5.

After crossover operation the mutation operation is applied. In this operation, for each gene, a random number is generated. If this random number is less than the mutation rate, then the corresponding gene is selected for mutation. During mutation a small random number, taken from the interval [0,1], is added to the selected gene in order to determine its new value.

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 being taken from literature in order to be able to make a direct performance comparison.

A. *Identification performance studies*

The identification problem involves the finding of the relation between the input and the output of the system. In Fig. 2 the structure of the identification scheme is shown. The inputs to the FWNN identifier are the external input signals, its one- two-,…, d_i- step delayed values and the one-, two-,…,d_o- step delayed outputs of the plant. Here the

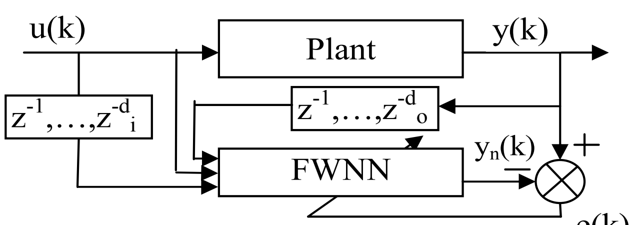

Fig.2. Identification scheme

problem is to find such values of parameters of FWNN that 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 the second order nonlinear plant that has been used in [25,9] is considered.

$$y(k)=f(y(k-1),y(k-2)y(k-3),u(k),u(k-1)) \quad (16)$$

In above

$$f(x_1,x_2,x_3,x_4,x_5)=\frac{x_1x_2x_3x_5(x_3-1)+x_4}{1+x_3^2+x_2^2} \quad (17)$$

and y(k-1), y(k-2), y(k-3) are one-, two- and three- step delayed outputs of the plant, u(k) and u(k-1) are current and one step delayed inputs of the plant. As can be seen, the current output of plant depends on previous input and output signals. To generate less number of parameters of the FWNN, only the current state of system and the control signal are fed into the FWNN as inputs. Increasing the number of input signals allow to increase the number of parameters of network. The FWNN with three fuzzy rules is applied for identification of dynamic plant (16). For the identification of the plant, the following excitation signal is used.

$$u(k)=\begin{cases}\sin(\pi k/25), & k<250\\ 1.0, & 250\le k<500\\ -1.0, & 500\le k<750\\ 0.3\sin(\pi k/25)+0.1\sin(\pi k/32) & \\ \quad +0.6\sin(\pi k/10), & 750\le k<1000\end{cases} \quad (18)$$

The gradient decent algorithm is applied for learning of the parameter values of FWNN. The initial values of the parameters are generated in the interval [–1, 1]. The training is continued for 200 epochs with 1000 time steps in each epoch. As a performance criterion the following root-mean-square-error (RMSE) is used (where K=1000).

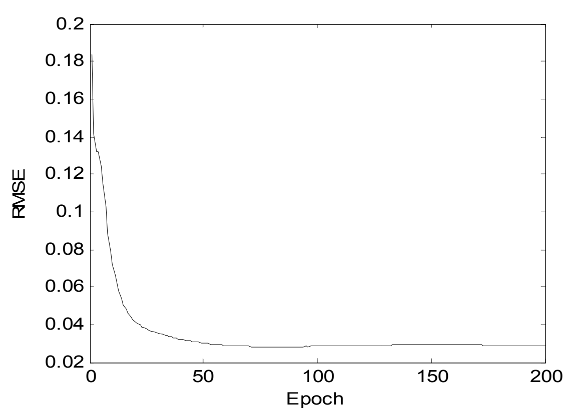

Fig.3. RMSE values obtained during learning

TABLE I
SIMULATION RESULTS OF DIFFERENT MODELS

Models	Network Parameters	RMSE	
		train	test
RFNN	112	0.0114	0.0575
RSONFIN [27]	36	0.0248	0.0780
Feedforward NFS	48	0.0203	0.0521
TRFN-S [9]	33	0.0084	0.0346
FWNN	27	0.0282	0.0301

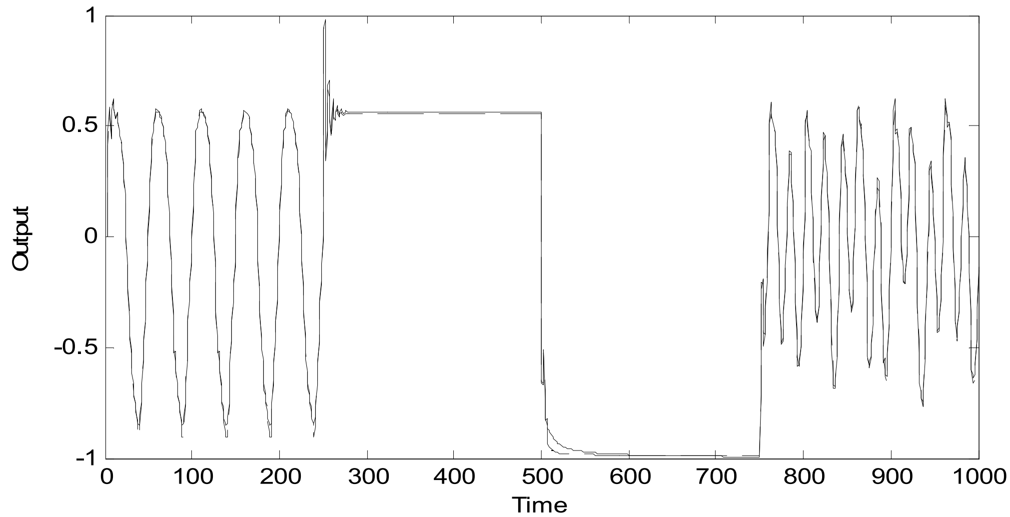

Fig. 4. Results of identification, where solid line denotes the output of the plant, dashed line denotes the FWNN output

$$J=\sqrt{\frac{\sum_{i=1}^{K}(y(k+1)-y_N(k+1))^2}{K}} \quad (19)$$

As a result of training, three fuzzy rules are generated and the parameters of the FWNN are determined. When the number of parameters is N=27 and, the value of root mean square error (RMSE) of identification obtained after training was 0.0282. RMSE value of FWNN for test data was 0.0301. Fig.3 depicts the evolvement of the RMSE values over 200 epochs. Fig.4 compares the actual plant output with that of the FWNN identifier. Table 1 compares the RMSE values with the other approaches reported in the literature, namely Recurrent Fuzzy Neural Network (RFNN), RSONFIN [27], feedforward neural fuzzy system (NFS) and TRFN-S [9].

B. *Control performance studies*

Gradient algorithm can be used for FWNN design when input-output training patterns are available then. When input-output training patterns aren't available or expensive to collect, another learning algorithm is required [9]. In this section, the GA is applied for design of FWNN for control purposes. The structure of the FWNN based control system is as shown in Fig. 5. Here 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 differentiation. Using these signals the learning of the

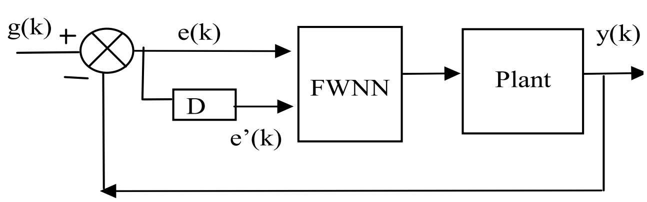

Fig. 5. Structure of FWNN based control system.

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 results in the control signal to be applied to the plant.

Example 2. The proposed FWNN structure is used for the control of the dynamic described by the following difference equation.

$$y(k)=\frac{y(k-1)y(k-2)(y(k-1)+2.5)}{(1+y(k-1)^2y(k-2)^2)}+u(k) \quad (20)$$

Three fuzzy rules are used in FWNN structure and consequently 27 parameters have to be updated. The initial values of the parameters of FWNNs are generated randomly in the interval [–10, 10] and a GA based approach is used to

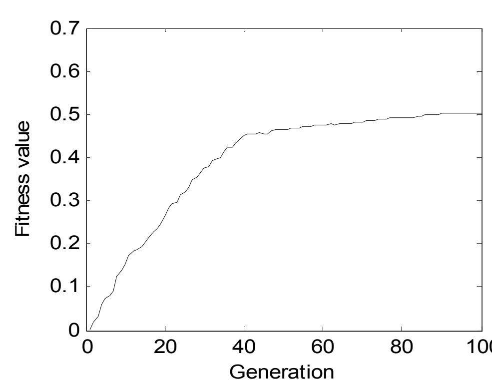

Fig. 6. The fitness values on each generation

reach the optimal values. The 50 chromosomes are

TABLE II
SIMULATION RESULTS OF FWNN BASED CONTROL OF DYNAMIC PLANT

Models	Network Parameters	RMSE	
		Mean	Best
ERNN+GA[26]	40	1.504	1.439
TRFN-G[9]	33	1.086	0.887
FWNN	27	0.6859	0.5876

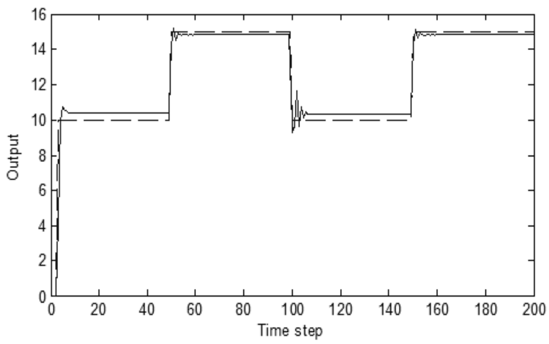

Fig. 7. Time response characteristic of the control system with. Dashed line is set-point signal, solid line is plant output.

generated as initial solution. The tournament selection, multipoint crossover and mutation operations are applied. Crossover rate is taken as 0.8, the probability of mutation is taken as 0.05. The training of FWNN system is performed for 200 data points. The fitness value is calculated as: J=1/($\sum_{i=1}^{ps} \sum_{k=1}^{K} (g - y(k))^2$). Here k=1,..,200 data points, ps is population size (ps=1,..,50). is Fig. 6 depicts the fitness values for each generation and in Fig. 7 the time response characteristics of the control system is shown. The best and the averaged RMSE values over 200 time steps are given in Table 2. This performance is compared with the other methods proposed in the literature, namely ERNN with Genetic Algorithm (ERNN+GA) [26] 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.

V. Conclusion

In this paper a fuzzy neural structure that uses wavelet functions is proposed for identification and control of dynamic plants. The proposed structure incorporates the advantages of wavelet function, neural networks and fuzzy logic. Using the gradient decent algorithm and GA the parameter update rules are derived. 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. Simulation results demonstrate that Fuzzy Wavelet Neural Network 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.

References

[1] R.R. Yager, L.A. Zadeh (Eds).. "Fuzzy sets, neural networks and soft computing", Van Nostrand Reinhold, New York. 1994.

[2] J-SR. Jang, C-T Sun, E. Mizutani. Neuro-fuzzy and soft computing.Ch 17, Prentice-Hall New Jersey. 1997.

[3] D. Nauck and R. Kruse. "Designing neuro-fuzzy systems through backpropagation", In Witold Pedryz, editor, Fuzzy Modelling: Paradigms and Practice, pages 203-228, Kluwer Academic Publisher, Boston, 1996.

[4] M.Onder Efe, and O. Kaynak, "On stabilization of Gradient-Based Training Strategies for Computationally Intelligent Systems". IEEE Transactions on Fuzzy Systems, Vol.8, No.5,pp.564-575, October, 2000.

[5] V. Topalov, G. L. Cascella, V. Giordano, F. Cupertino and O. Kaynak, "Sliding Mode Neuro-Adaptive Control of Electrical Drives", IEEE Trans. Indust. Electron.,v.54, no:1, pp. 671-679, 2007.

[6] F. Da, W. Song, "Fuzzy neural networks for direct adaptive control," Trans. on Industrial Electronics, vol. 50, no. 3, pp. 507- 513, Jun 2003.

[7] M.J. Er, Y. Gao, "Robust adaptive control of robot manipulators using generalized fuzzy neural networks," Trans. on Industrial Electronics, vol. 50, no. 3, pp. 620- 628, 2003.

[8] C.H.Lee, and C.C.Theng, "Identification and control of dynamic systems using recurrent fuzzy neural network", IEEE Trans. Fuzzy Systems, vol. 8, pp.349-366, 2000.

[9] C-F. Juang, "A TSK –type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithm", IEEE Trans. Fuzzy Systems, vol.10, pp.155-170, 2002.

[10] J. Nunez-Garcia and O. Wolkenhauer, "Random Set System Identification", IEEE Transactions on Fuzzy Systems, Vol.10, No.3, October, pp.287-296. 2002.

[11] T. Kugarajah and O.Zhang. Multidimensional wavelet frames. IEEE Transaction on Neural Networks 6, pp.1552-1556. 1995.

[12] Q.Zhang and A.Benviste. Wavelet networks. IEEE Transaction on Neural Networks, vol. 3, pp.889-898. 1995.

[13] H. Szu, B. Telfer and J. Garcia. Wavelet Transforms and Neural Networks for Compression and recognition. Neural Networks 9, pp.695-708. 1996.

[14] J.Zhang, G.G.Walter, W.N.Wayne Lee. Wavelet Neural Networks for Function Learning. IEEE Transaction on Signal Processing, Vol.43, No.6. pp.1485-1497. 1995.

[15] T.Q.D.Khao,L.M.Phuong, P.T.T.Binh, N.T.H.Lien. "Application of Wavelet and Neural network to Long-Term Load Forecasting. International Conference on Power System technology, POWERCON 2004, Singapore, pp.840-844. 2004.

[16] R-J. Wai, R-Y. Duan, J-D. Lee, H-H. Chang, "Wavelet neural network control for induction motor drive using sliding-mode design technique," Trans. on Industrial Electronics, vol. 50, no. 4, pp. 733-748, Aug 2003.

[17] C.K.Lin, S.D.Wang. Fuzzy Modelling Using Wavelet Transform". Electronics letters, Vol.32, pp.2255-2256. 1996.

[18] M. Thuillard. Fuzzy logic in the wavelet framework. Proc. Toolmet'2000, April 13-14,Oulu. 2000.

[19] M. Thuillard. Wavelets in Softcomputing. World Scientific Press. 2001.

[20] O-J. Guo, H-B. Yu, A-D. Xu. Wavelet fuzzy network for fault diagnosis. Proceedings of International Conference on Communications, Circuits and Systems, pp.993-998. 2005.

[21] Y. Lin, F-Y. Wang. Predicting Chaotic Time-series Using Adaptive Wavelet-Fuzzy Inference System. Proceeding of IEEE Intelligent Vehicles Symposium, pp.888-893. 2005.

[22] W.C. Daniel Ho, P-A. Zhang, and J. Xu. Fuzzy Wavelet Networks for Function Learning. IEEE Transactions on Fuzzy Systems, Vol.9, No.1, pp.200-211. 2001.

[23] R. H. Abiyev. Controller Based of Fuzzy Wavelet Neural Network for Control of Technological Processes CIMSA 2005 – IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, Giardini Naxos, Italy. pp.215-219. 2005.

[24] R. H. Abiyev. Time Series Prediction Using Fuzzy Wavelet Neural Network Model. Lecture Notes in Computer Sciences, Springer-Verlag, Berlin Heidelberg, pp.191-200. 2006.

[25] K.S.Narendra, K.Parthasarathy. "Identification and control dynamical systems using neural networks". IEEE Transaction on Neural Networks, vol.1, pp.4-27, 1990.

[26] J.L. Elman. Finding structure in time. Cognit Sci 14:179–211, 1990.

[27] C-F. Juang, C-T. Lin. A recurrent self-organizing neural fuzzyinference network. IEEE Trans. Neural Networks, 10(4):828–845, 1999.