A Fuzzy Wavelet Neural Network Model for System Identification

Sevcan Yılmaz[1], Yusuf Oysal[1]
[1]Anadolu University, Computer Engineering Department, Eskisehir, Turkey
sevcany@anadolu.edu.tr, yoysal@anadolu.edu.tr

Abstract

In this paper, a fuzzy wavelet neural network model is proposed for system identification problems. The proposed model is obtained from the traditional Takagi-Sugeno-Kang (TSK) fuzzy system by replacing the consequent part of fuzzy rules with wavelet basis functions that have time-frequency localization properties. We use a radial function of Mexican Hat wavelet in the consequent part of each rule. A fast gradient algorithm based on quasi-Newton methods is used to obtain the optimal values for unknown parameters of the model. Simulation results of some benchmark problems in the literature are also given to illustrate the effectiveness of the model.

1. Introduction

Models of real systems are of fundamental importance in virtually all disciplines. In engineering, models are required for the design of new processes and for the analysis of existing process [1]. System identification involves finding a relation between the input and output of the system [2, 3]. In the literature, several methods are applied to solve system identification problems such as polynomials [4], neural networks [3, 5], and neuro-fuzzy systems [6, 7].

In recent years, wavelets have become very popular and have been applied in many scientific and engineering research areas such as system identification, signal processing and function approximation. They have very important properties such as time-frequency localization property. With this property, wavelets can capture global (low frequency) and local (high frequency) behavior of any function easily [8].

Wavelet neural networks (WNN) which combine neural networks with wavelet functions are also used in function approximation and system identification problems [9]-[11]. In WNNs, wavelet functions are used in hidden layers of neural networks as activation functions instead of local functions in time such as Gaussian and sigmoid functions. WNNs can converge quickly, can be easily trained and give high accuracy.

Fuzzy wavelet neural networks (FWNN) combine neuro-fuzzy systems with wavelet functions. In the literature, several FWNN models are proposed for time series prediction, system identification and control problems [2], [8] and [12]-[14]. The FWNN proposed in [2] uses summation of dilated and translated versions of wavelet functions in consequent part of fuzzy rules for system identification and control purposes. In [8] and [12], each fuzzy rule is represented by a sub-WNN which consists of single-scaling wavelets that has same dilation parameters for all dimensions and orthogonal least-square algorithm is used to select important wavelets. The resulting network is used for function approximation in [8] and control of nonlinear systems in [12]. In [13], the proposed model consists of a set of IF–THEN rules and, THEN parts are series expansion in terms of wavelets functions and this model is applied to system modeling. In [14], the inputs enter into discrete wavelet transform block, then the output of this block is fuzzified and it forms the input to a single neural network.

A FWNN model which is called as radial FWNN (RFWNN) in this paper is proposed for system identification problems. In this model, a radial function of wavelets is used in consequent part of fuzzy rules whereas generally constant function or linear combination of inputs is used in TSK fuzzy systems. Gaussian type functions are used as membership functions in the premise part of the rules.

The rest of this paper is organized as follows. The structure of proposed RFWNN model is explained in Section 2. The training algorithm and parameter update rules are given in Section 3. To illustrate and compare the performance of the RFWNN, three simulation examples are provided in Section 4. Finally, a brief conclusion is drawn in Section 5.

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.96

2. Fuzzy wavelet neural network model structure

The proposed RFWNN combines TSK fuzzy system with wavelet functions. In the RFWNN, constant or linear functions in consequent part of the rules in TSK fuzzy system are substituted with wavelet functions in order to increase computational power of neuro-fuzzy system by using the fact that wavelets have time-frequency localization property. In this paper, translated and dilated version of Mexican Hat wavelet function is used which is given by the following equation:

$$\psi(\frac{x-b_i}{c_i}) = (1-(\frac{x-b_i}{c_i})^2)\exp(-\frac{1}{2}(\frac{x-b_i}{c_i})^2) \qquad (1)$$

Translation parameter (b_i) determines the center position of the wavelet, whereas dilation parameter (c_i) controls the spread of the wavelet.

The structure of two input one output RFWNN model with two membership functions for each input is shown in Figure 1. In this model, the first rule is in the following form:

IF x_1 is A_{11} **AND** x_2 is A_{21} **THEN** Ψ_{11}

where x_1 and x_2 are input variables, A_{11} and A_{21} are Gaussian type membership functions and Ψ_{11} is a radial function of wavelets in consequent part of the rule.

The six layer structure of two input one output RFWNN model is explained layer by layer below for $i=1,2$ and $j=1,2$.

- ***Layer 1:*** This layer is the input layer. Each neuron in this layer transmits external crisp input signals (x_1 and x_2) directly to the next layer.
- ***Layer 2:*** This layer is fuzzification layer. Neurons in this layer represent fuzzy sets used in the antecedents of fuzzy rules. The outputs of this layer are the values of the membership functions. The *j*th Gaussian type membership function for the *i*th input is given by:

$$A_{ij}(x_i) = \exp(-\frac{1}{2}(\frac{x_i-\mu_{ij}}{\sigma_{ij}})^2) \qquad (2)$$

- ***Layer 3:*** This layer is the fuzzy rule layer. Each node in this layer represents a fuzzy rule. In order to calculate the firing strength of each rule, multiplication is used as AND (t-norm) operator.

$$\eta_{ij} = A_{1i}(x_1)\cdot A_{2j}(x_2) \qquad (3)$$

- ***Layer 4:*** This layer is normalization layer. Each neuron in this layer calculates the normalized activation strength of a given rule by:

$$\bar{\eta}_{ij} = \frac{\eta_{ij}}{\sum_{i=1}^{2}\sum_{j=1}^{2}\eta_{ij}} \qquad (4)$$

- ***Layer 5:*** This layer calculates the weighted consequent value of a given rule as follows:

$$f_{ij} = \bar{\eta}_{ij}\Psi_{ij} \qquad (5)$$

$$\Psi_{ij} = w_{ij}(1-\phi_{ij}^2)\exp(-\frac{1}{2}\phi_{ij}^2) + p_{ij} \qquad (6)$$

$$\phi_{ij} = \frac{\|\boldsymbol{x}-\boldsymbol{b_{ij}}\|}{c_{ij}} \qquad (7)$$

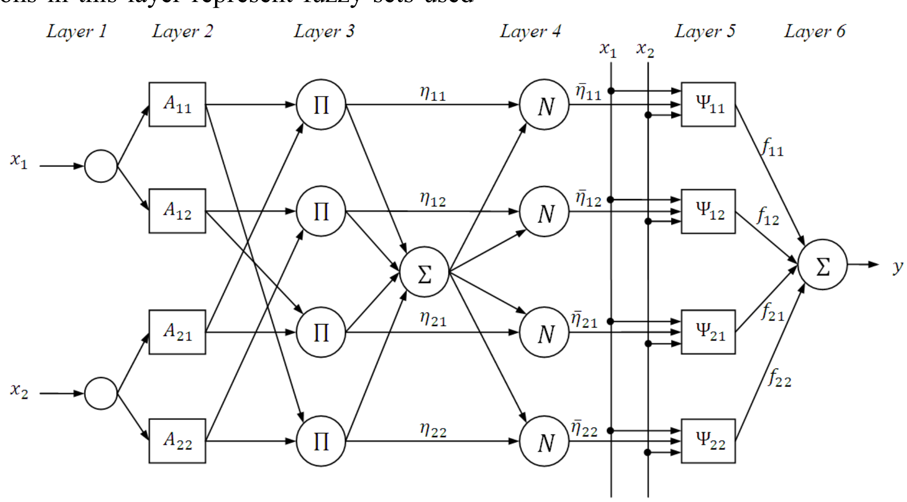

Figure 1. The fuzzy wavelet neural network model structure

Here, $\boldsymbol{x}$ is input vector, $\boldsymbol{b_{ij}}$ and c_{ij} are translation and dilation parameters of radial wavelet function respectively, and w_{ij} and p_{ij} are weight and bias parameters respectively. In each rule, single dilation parameter is used for all input dimensions whereas different translation parameters are used for each input.

- ***Layer 6:*** This layer is output layer. It computes the overall output of system as follows:

$$y=\sum_{i=1}^{2}\sum_{j=1}^{2}f_{ij} \tag{8}$$

3. Training algorithm

The RFWNN training is to encapsulate a given function or input-output pairs by adjusting network parameters. Unknown parameters are center parameters (**$\boldsymbol{\mu}$**) and scaling parameters ($\boldsymbol{\sigma}$) of Gaussian membership functions in antecedent part of the rules, and translation (***b***), dilation (***c***) parameters of wavelet functions and weight (***w***) and bias (***p***) parameters in the consequent part of the rules.

The RFWNN training is done by minimizing a performance index. For this purpose, mean square error (MSE) is selected as performance index which is given by:

$$E=\frac{1}{N}\sum_{k=1}^{N}(y-y_d)^2 \tag{9}$$

where N is the total number of input-output pairs of the function to be approximated, y_d is the desired output, and y is the RFWNN output.

In this paper, unknown parameters of the RFWNN model are adjusted by using Broyden-Fletcher-Goldfarb-Shanno (BFGS) gradient method. BFGS method is derived from Newton's method in optimization which is a class of hill-climbing optimization techniques. It tries to seek the stationary point of a function, where the gradient is zero [15]. It is assumed that at the each iteration of the training algorithm, gradients of the performance index with respect to all unknown parameters (***p***) of RFWNN, $g=\frac{\partial E}{\partial p}$ is computed. Parameter update rules of BFGS algorithm is given by:

$$p^{k+1}=p^k+\tau^k d^k \tag{10}$$

$$\tau^k=\min_{\tau}\ \mathrm{J}(\mathrm{p}^{\mathrm{k}}+\tau d^k) \tag{11}$$

$$d^k=-H^k g_p^k \tag{12}$$

Here ***p*** is the parameter to be updated, ***d*** is the search direction, $\boldsymbol{\tau}$ is the optimal step size along the search direction, ***g*** is the cost gradient with respect to parameter ***p*** and $H\cong(\nabla_p J)^{-1}$ is the inverse of the approximate Hessian matrix given by:

$$H^{k+1}=\left[I-\frac{\Delta p^k(\Delta g_p^k)^T}{(\Delta p^k)^T\Delta g_p^k}\right]\mathrm{H}^{\mathrm{k}}\left[I-\frac{\Delta p^k(\Delta g_p^k)^T}{(\Delta p^k)^T\Delta g_p^k}\right]^{\mathrm{T}}+\frac{\Delta p^k(\Delta p^k)^T}{(\Delta p^k)^T\Delta g_p^k}\ ,\ \mathrm{H}^0=I \tag{13}$$

Δp and Δg are the backward differences of the parameter and gradient vectors, respectively. They provide the history of parameter and gradient changes yielding approximate second order information.

The gradients of the parameters in membership functions can be calculated by following formulas for *i=1,2* and *j=1,2*:

$$\frac{\partial E}{\partial\mu_{ij}}=\frac{\partial E}{\partial y}\frac{\partial y}{\partial A_{ij}}\frac{(x_i-\mu_{ij})}{\sigma_{ij}^2}\exp\left(-\frac{1}{2}\left(\frac{x_i-\mu_{ij}}{\sigma_{ij}}\right)^2\right) \tag{14}$$

$$\frac{\partial E}{\partial\sigma_{ij}}=\frac{\partial E}{\partial y}\frac{\partial y}{\partial A_{ij}}\frac{(x_i-\mu_{ij})^2}{\sigma_{ij}^3}\exp\left(-\frac{1}{2}\left(\frac{x_i-\mu_{ij}}{\sigma_{ij}}\right)^2\right) \tag{15}$$

$$\frac{\partial E}{\partial y}=\frac{2}{N}\sum_{k=1}^{N}(y-y_d) \tag{16}$$

For above calculations, partial derivative of the output y with respect to membership functions of each input variable is needed. For example, for the first membership function of the first input variable, this can be calculated as:

$$\frac{\partial y}{\partial A_{11}}=\frac{(\Psi_{11}A_{21}+\Psi_{12}A_{22})-y(A_{21}+A_{22})}{\sum_{i=1}^{2}\sum_{j=1}^{2}A_{1i}A_{2j}} \tag{17}$$

The gradients of the parameters of wavelet function in the consequent part of the rules are given by following formulas for *i=1,2, j=1,2* and *k=1,2*:

$$\frac{\partial E}{\partial w_{ij}}=\frac{\partial E}{\partial y}\bar{\eta}_{ij}\left(1-\phi_{ij}^2\right)\exp\left(-\frac{1}{2}\phi_{ij}^2\right) \tag{18}$$

$$\frac{\partial E}{\partial b_{ij}^k}=\frac{\partial E}{\partial y}\bar{\eta}_{ij}w_{ij}\frac{\left(x_i-b_{ij}^k\right)}{c_{ij}^2}\left(3-\phi_{ij}^2\right)\exp\left(-\frac{1}{2}\phi_{ij}^2\right) \tag{19}$$

$$\frac{\partial E}{\partial c_{ij}}=\frac{\partial E}{\partial y}\bar{\eta}_{ij}w_{ij}\frac{\phi_{ij}^2}{c_{ij}}\left(3-\phi_{ij}^2\right)\exp\left(-\frac{1}{2}\phi_{ij}^2\right) \tag{20}$$

$$\frac{\partial E}{\partial p_{ij}}=\frac{\partial E}{\partial y}\bar{\eta}_{ij} \tag{21}$$

where $\| \boldsymbol{x} - \boldsymbol{b}_{ij} \|^2 = \sum_{k=1}^{2} (x_i - b_{ij}^k)^2$ and ϕ_{ij} is (7).

4. Simulation examples

The proposed RFWNN model is applied to a function approximation, to a system identification problem and to a prediction problem in order to show the performance of the model.

4.1. Approximation of a piecewise function

A piecewise function studied by Zhang [16] and Chen [17] is used to compare the RFWNN with some other wavelet-based networks. This function is defined as

$$y_d = f(x) = \begin{cases} -2.186x - 12.864 & -10 \le x < -2 \\ 4.246x & -2 \le x < 0 \\ 10e^{-0.05x-0.5}\sin[(0.03x+0.7)x] & 0 \le x \le 10 \end{cases} \quad (22)$$

For the training process, N = 200 sample points are drawn from the data uniformly distributed over [-10, 10]. In order to compare the proposed model with other works, the measure in [8] is used:

$$J = \sqrt{\frac{\sum_{i=1}^{N} (y - y_d)^2}{\sum_{i=1}^{N} (y_d - y_{av})^2}} \quad (23)$$

where y_d is actual output, y is predicted output and $y_{av} = \frac{1}{N}\sum_{i=1}^{N} y_d$.

In Table 1, it is seen that the performance of the proposed model is superior to that of the other WNNs. Figure 2 illustrates the validity of the RFWNN model with smallest performance measure value among the simulations.

Table 1. Comparison of RFWNN model with other models for the piecewise function

Models	Network Param.	J
RFWNN	42	0.0031
RFWNN	36	0.0041
RFWNN	30	0.0116
FWN [8]	28	0.021
WNN[16]	22	0.05057
WNN[17]	23	0.0480

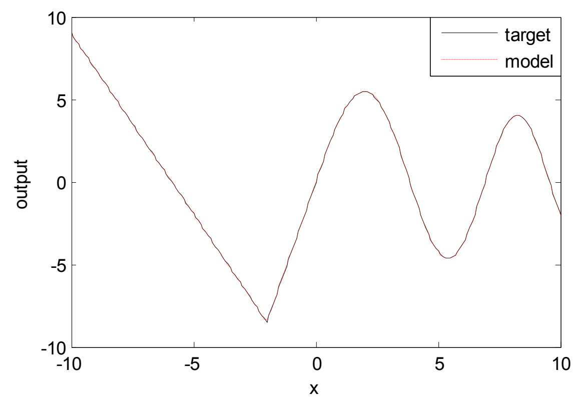

Figure 2. Actual and predicted values with RFWNN for the piecewise function

4.2. System identification example

System identification involves finding the relation between the input and output of the system [2,3]. In this example, the plant to be identified is given by following equation:

$$y(k) = 0.72y(k-1) + 0.025y(k-2)u(k-2) + 0.01u^2(k-3) + 0.02u(k-4) \quad (24)$$

The output of the system depends on two previous output values and three previous input values. However, only u(k-1) and y(k) are used as inputs to the FWNN models to predict y(k+1). Two membership functions are selected for each input of the RFWNN model. In order to train the RFWNN, 900 inputs are used similar to the inputs used in [18] and [19]. The half of the inputs is independent and identically distributed (i.i.d.) uniform sequence over [-2, 2] and the remaining is a sinusoid given by 1.05sin(πk/45). The RFWNN is trained for 200 epochs (Figure 3). After training, the following input signal which is same test signal with other compared models is used for testing the performance of the RFWNN.

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

Root mean square error (RMSE) is taken as measure for system identification example. Figure 4 shows the actual and predicted output of the plant for test signal with RFWNN model. From Table 2, it can be seen that the proposed RFWNN model illustrates much better performance than the models in [18]-[20] with less parameters. The proposed RFWNN gives better training results than FWNN[2] and almost same test result with FWNN[2].

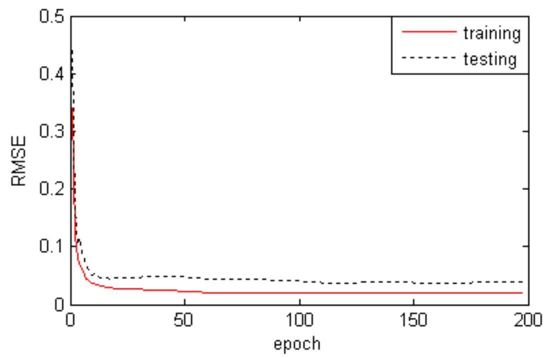

Figure 3. RMSE values during training and testing for system identification example

Table 2. Comparison of RFWNN model with other models for system identification example

Models	Network Param.	RMSE Training	RMSE Testing
ERNN[20]	54	0.036	0.078
RSONFIN[18]	49	0.03	0.06
TRFN-S[19]	33	0.0067	0.0313
FWNN[2]	27	0.01973	0.02260
FWNN[2]	43	0.01871	0.02016
RFWNN	28	0.00968	0.02220

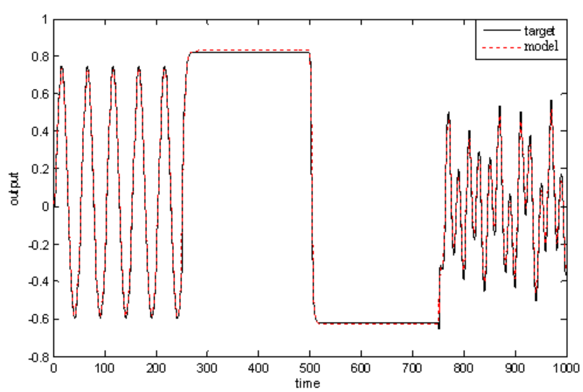

Figure 4. Actual and predicted values with RFWNN for system identification example

4.3. Sunspot number prediction

In this section, annually recorded sunspot time series for the years 1700-1979 is considered to show the performance of RFWNN model. These numbers show the yearly average relative number of sunspots observed. To make meaningful comparisons, the dataset is divided into three parts. The data points between years 1700-1920 are used for training the models. The data points for years 1921-1955 and 1956-1979 form the first and the second test sets respectively. The y(t-4), y(t-3), y(t-2) and y(t-1) are used as inputs to our models in order to predict the output y(t). Two membership functions are selected for each input and the model trained for 200 epochs. Normalized mean square error (NMSE) is used to compare the RFWNN with other models.

$$NMSE = \frac{\sum_{k=1}^{N}(y-y_d)^2}{\sum_{k=1}^{N}(y_d-\bar{y}_d)^2} \quad \text{where } \bar{y}_d = \frac{1}{n}\sum_{k=1}^{N} y_d$$

Training and testing error values are given in Table 3 with comparison of other models in the literature. In Figure 5, actual output of time series and prediction results of RFWNN are shown.

Table 3. Comparison of RFWNN model with other models for sunspot number prediction

Model	Network Param.	NMSE training	NMSE testing 1	NMSE testing 2
Tong [21]	16	0.097	0.097	0.28
Weigend [22]	43	0.082	0.086	0.35
Svarer [23]	12-16	0.090	0.082	0.35
Transversal Net[24]	14	0.0987	0.0971	0.3724
Recurrent net[24]	22	0.1006	0.0972	0.4361
RFNN[25]	-	-	0.074	0.21
RFWNN	128	0.0796	0.1099	0.2549

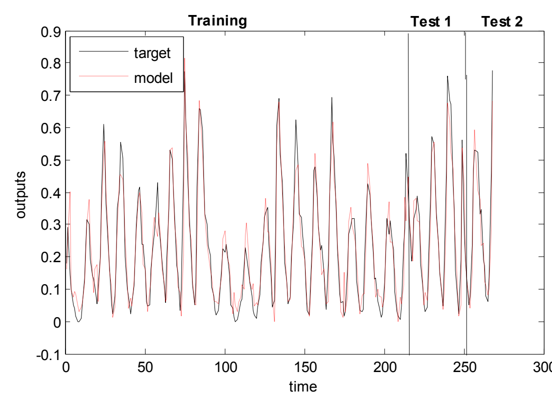

Figure 5. Actual and predicted values with RFWNN for sunspot number prediction

5. Conclusion

In this paper, a RFWNN model is introduced. This model combines neuro-fuzzy model with wavelet basis functions which have time frequency localization properties. The presented RFWNN model has advantages of high approximation accuracy and good generalization performance for system identification problems.

This study is one of the parts of our ongoing research on FWNN models [26] and [27]. It is believed that the RFWNN model can also be applied to a wider range of real-world problems such as speech and image processing, financial data analysis and prediction and other system identification and control applications. Other optimization techniques such as particle swarm optimization can also be used for training unknown parameters of the RFWNN.

6. References

[1] Oliver Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models,* Springer, December 2000.

[2] R. H. Abiyev, O. Kaynak, "Fuzzy Wavelet Neural Networks for Identification and Control of Dynamic Plants-A Novel Structure and Comparative Study," *IEEE Trans. Ind. Electron.*, vol. 55, no. 8, pp. 3133 – 3140, Aug. 2008.

[3] 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, 1990.

[4] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to nonlinear system identification," *Int. J. Control*, vol. 50, pp. 1873–1896, Nov. 1989.

[5] P. S. Sastry,G. Santharam, and K. P. Unnikrishnan, "Memory Neuron Networks for Identification and Control of Dynamical Systems," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 306-319, March 1994.

[6] J. Kim, and N. Kasabov, "HyFIS: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems," *Neural Networks*, vol. 12, pp. 1301-1319, 1999.

[7] Mahdi Jalili-Kharaajoo, "Nonlinear System Identification Using ANFIS Based on Emotional Learning," *Lecture Notes in Computer Science*, vol. 3315, pp. 697-707, 2004

[8] D.W.C. Ho, Ping-Au Zhang, Jinhua Xu, "Fuzzy wavelet networks for function learning," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 1, pp. 200-211, Feb. 2001.

[9] J. Zhang, G.G. Walter, and W.N.W. Lee, "Wavelet neural networks for function learning," *IEEE Trans. Signal Process*, vol. 43, no.6, pp. 1485-1497, June 1995.

[10] A. Billings Stephen, Hua-Liang Wei, "A New Class of Wavelet Networks for Nonlinear System Identification," *IEEE Trans. Neural Netw.,* vol. 16, no. 4, July 2005.

[11] Y. Chen, B. Yang, and J. Dong, "Time-series prediction using a local linear wavelet neural network," *Neurocomputing*, vol. 69, pp. 449-465, 2006.

[12] Maryam Zekri, Saeed Sadri, Farid Sheikholeslam, "Adaptive fuzzy wavelet network control design for nonlinear systems," *Fuzzy Sets and Systems*, vol. 159, pp. 2668 – 2695, Oct. 2008.

[13] Engin Karatepe, Musa Alçı, "A new approach to fuzzy wavelet system modeling," *International Journal of Approximate Reasoning*, vol. 40, no.3, pp. 302-322, Nov. 2005.

[14] S. Srivastavaa, M. Singha, M. Hanmandlub, A.N. Jha, "New fuzzy wavelet neural networks for system identification and control," *Applied Soft Computing*, vol. 6, pp. 1–17, 2005.

[15] P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization*, Academic Press Ltd, 1993.

[16] Q. Zhang, and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Netw.*, 3, 889-898, 1992.

[17] J. Chen, and D.D. Bruns, "WaveARX neural network development for system identification using a systematic design synthesis," *Ind. Eng. Chem. Res.*, vol. 34, pp. 4420-4435, 1995.

[18] 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, 1999.

[19] 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, 2002.

[20] J. L. Elman, "Finding structure in time," *Cognit. Sci.,* vol. 14, pp. 179–211, 1990.

[21] H. Tong and K. S. Lim, "Threshold autoregression, limit cycles and cyclical data," *Journal Royal Statistical Society B*, vol. 42, pp. 245–292, 1980.

[22] A. S. Weigned, D. E. Rumelhart, and B. A. Huberman, "Predicting the future: A connectionist approach," Techn. Rep. Stanford-PDP-90- 01 or PARC-SSL-90-20, 1990.

[23] C. Svarer, L. K. Hansen, and J. Larsen, "On design and evaluation of tapped-delay neural network architectures," in *Proc. IEEE Int. Conf. Neural Netw.*, 1992, San Francisco.

[24] J. R. McDonnell, D. Waagen, "Evolving Recurrent Perceptrons for Time-Series Modeling," *IEEE Trans. Neural Netw.*, vol. 5, no. 1, 1994.

[25] R.A. Alieva, B.G. Guirimov, Bijan Fazlollahi, R.R. Aliev, "Evolutionary algorithm-based learning of fuzzy neural networks. Part2: Recurrent fuzzy neural networks," *Fuzzy Sets and Systems*, vol. 160, no. 17, pp. 2553–2566, 2009.

[26] Sevcan Yılmaz, "A New Fuzzy Wavelet Neural Network Design for Time Series Prediction," *Master's Thesis, Anadolu University*, Eskişehir, Turkey, June 2009.

[27] Sevcan Yılmaz, Yusuf Oysal, "Fuzzy Wavelet Neural Network Models for Prediction and Identification of Dynamical Systems," *IEEE Trans. Neural Netw,* submitted, 2009.