MIMO Nonlinear Dynamic Systems Identification Using Fully Recurrent Wavelet Neural Network

Afsaneh Ghadirian, Maryam Zekri

Abstract—**In this paper, we expand a fully recurrent wavelet neural network (FRWNN) for the multi-input multi-output (MIMO) nonlinear dynamic systems identification. The presented identifier combines the properties of recurrent neural network (RNN) such as storage of past information of the network and the basic ability of wavelet neural network (WNN) such as the fast convergence and localization properties. Here, we use the MIMO FRWNN to identify the single-input single-output (SISO) and MIMO nonlinear dynamic systems. The real time recurrent learning (RTRL) algorithm is applied to adjust the shape of wavelet functions and the connection weights of the network. The simulation results verify that the FRWNN is capable of accurately identifying nonlinear dynamic systems and can rapidly get the dynamical performance. Also in this paper, the FRWNN is compared with a fully recurrent neural network (FRNN) that the structures of both are similar. Compared to the FRNN, the FRWNN has a less error and better performance.**

I. Introduction

NEURAL networks have been established as a general approximation tool for fitting nonlinear models for input-output data [1], [2]. Since the feed forward neural network is a static mapping, it can not represent a dynamic system without the help of tapped delay lines. Furthermore, the drawbacks associated with the use of feed forward neural networks for system identification are a long computation time, easily being affected by external noises and difficulty in obtaining an independent system simulator [3].

Recurrent neural networks are able to deal with time varying input or output through their own natural temporal operation. Thus the recurrent neural network (RNN) is a dynamic mapping and is better suited for dynamical system than the feed forward network. Recently, various recurrent neural networks have been offered to remedy the limitations of the feed forward networks [4]-[8].

On the other hand, the wavelet neural network (WNN) has been proposed as an alternative to feed forward NNs for approximating arbitrary nonlinear functions based on the wavelet transform theory [9]. Wavelet is a local function that has a bounded support. The WNNs keep the localization properties of wavelets and the learning abilities of NNs and have high accuracy and fast learning ability. Also, the concept of the WNN has become increasingly important and can be used for identification [10]-[12] and solving approximation, classification, prediction, and control problems [13]-[16].

The recurrent wavelet neural network (RWNN) combines the properties of the RNN and good performance of the WNN. Since RWNN is able to preserve past states of the network, it has the capability to deal with temporal problems. In RWNN structure, the mother wavelet layer is composed of internal feedback neurons to capture the dynamic response of a system. In [17], a RWNN structure was used for identification of a complex nonlinear dynamic system and prediction of a chaotic time series. In references [18], and [19] two RWNN controllers were used to solve nonlinear system control problems.

The objective of this paper is to expand the fully recurrent wavelet neural network (FRWNN) in [17], and [20] for MIMO nonlinear dynamic systems identification. The FRWNNs, in which every unit is connected to every other unit, are highly nonlinear dynamical systems that exhibit a rich and complex dynamical behavior. The real time recurrent learning (RTRL) algorithm is applied to adjust the shape of wavelet functions and the connection weights. Finally, the MIMO FRWNN is applied to identify the SISO and MIMO nonlinear dynamic systems. Also, the FRWNN is compared with a fully recurrent neural network (FRNN) that the structures of both are similar. In this paper, the parameters of Gaussian function of FRNN are considered to be adjustable because of the better comparison between FRWNN and FRNN identifiers.

This paper is organized as follows. In section II, we introduce the structure of proposed FRWNN. Section III discusses the problem statement of the identification scheme. The RTRL algorithm is discussed in section IV for training the network. Section V provides the computer simulations to validate the effectiveness of the proposed FRWNN. Finally, the conclusions are presented in section VI.

II. FRWNN Structure

A wavelet family is contained of dilated and translated version of a mother wavelet function $\psi(x)$ as follows:

Manuscript received September 28, 2011.

A. G. Author was with the Department of Electrical and Computer Engineering Isfahan University of Technology(IUT), Isfahan, IRAN, (e-mail: afsaneh_ghadirian@yahoo.com).

M. Z. Author is with the Department of Electrical and Computer Engineering Isfahan University of Technology(IUT), Isfahan, IRAN, (corresponding author to provide phone: +98-311-391-5397; fax: +98-311-391-2451; e-mail: mzekri@cc.iut.ac.ir).

$$\psi_{t,s}(x) = s^{-\frac{1}{2}}\psi\left(\frac{x-t}{s}\right), \quad t,s \in R \ , s > 0 \tag{1}$$

where s, t are dilation and translation parameters of a wavelet function, respectively. The multidimensional wavelet frames can be built by a tensor product of one dimension wavelet functions [21].

In this paper, we consider the FRWNN structure with multi-input and multi-output. A schematic diagram of the FRWNN structure consist of three layers is shown in Fig. 1, which has m external inputs and n units (p units in output layer and n-p units in hidden layer). In the recurrent network used here, any unit can be connected to any other and any unit can receive external input.

The layer 1 is an input layer. This layer accepts the input variables and transmits the accepted inputs to the hidden layer and output layer directly.

The layer 2 is a hidden layer. Each unit of this layer consists of a multidimensional wavelet.

The layer 3 is an output layer. The output units are the linear combination of consequences obtained from the output of the layer 2 and from itself. In addition, the output nodes accept directly input values from the input layer.

III. Problem Statement

Our interest in this paper is the identification using the proposed FRWNN for the nonlinear dynamic system expressed by:

$$\begin{aligned} y(k+1) = F\big[&y(k), y(k-1), \ldots, y(k-n+1), \\ &u(k), u(k-1), \ldots, u(k-m+1)\big], \end{aligned} \tag{2}$$

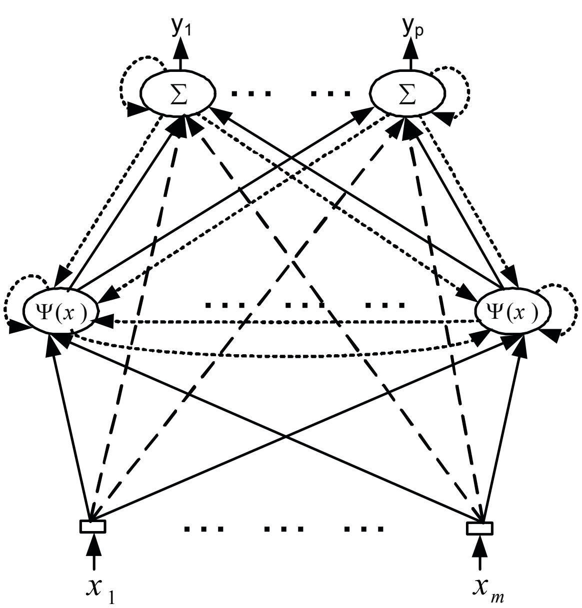

Fig. 1. Fully recurrent wavelet neural network.

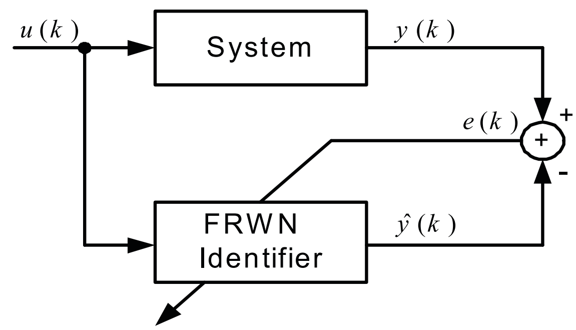

Fig. 2. Neural identifier scheme.

where $u(k) \in \Re^{p}$ and $y(k) \in \Re^{q}$ are the input and output, respectively. The mapping $F : \Re^{n_q + m_p} \rightarrow \Re^{q}$ is an unknown continuous function. 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. To model an unknown plant, the network only requires a sequence of input signals $u(k)$ and the corresponding plant output signals $y(k+1)$ at the discrete time k. The purpose of the weight learning of the FRWNN is to estimate the weight such that the output $\hat{y}(k)$ of the network convergences to the plant output $y(k)$ as $k \rightarrow \infty$. Here $y(k)$ is plant output, $\hat{y}(k)$ is output of FRWNN system.

IV. Learning Algorithm

The learning algorithm is similar to the real time recurrent learning algorithm (RTRL) [22] with random initial conditions. According to Fig.1, a FRWNN has n units (p units in output layer and n-p units in hidden layer) and m external input lines (including the bias input). Let $y(t)$ denote the n-tuple of outputs of the units in the network and let $x(t)$ denote the m-tuple of external inputs to the network at time t. It will be convenient in what follows to define $z(t)$ to be $(m+n)$-tuple obtained by concatenating $x(t)$ and $y(t)$. To distinguish the components of $z(t)$ representing the unit outputs from those representing external input values where necessary, let U denote the set of indices k such that z_k, the kth component of z, is the output of a unit in the network (which can receive target value), and let I denote the set of indices k for which z_k is an external input, i.e.

$$z_k(t) = \begin{cases} x_k(t) & if \ \ k \in I \\ y_k(t) & if \ \ k \in U \end{cases} \tag{3}$$

In the derivation below, we consider the mapping of the form $g:\Re^n \to \Re^p$. The derivation for the multiple input and single output (MISO) case is straight forward and the changes to be made will be indicated at appropriate places.

As the output nodes are linear, the output of the network in the case of a MIMO system is:

$$y_k(t+1) = \boldsymbol{W}\boldsymbol{z}, \ \textit{where} \quad k \in O, \ O \subset U \tag{4}$$

where O contains the indices of the nodes that receive targeted values and $\boldsymbol{W}$ is the weight matrix (of size $p \times (m+(n-p+1))$) between the output units and the remaining nodes in the network. But in the case of a MISO system, $\boldsymbol{W}$ is the weight vector (of length m+n) and (4) becomes:

$$y_n(t+1) = \boldsymbol{W}^T \boldsymbol{z}, \quad \boldsymbol{z} = [\boldsymbol{x}\ \boldsymbol{y}]^T \tag{5}$$

where in MISO $O = \{n\}$.

The output of the rest of the nodes is given by:

$$y_k(t+1) = \prod_{i=1}^{m+n} \psi\left(\frac{z_i - t_{ki}}{s_{ki}}\right), \tag{6}$$

where $k \in U - O$ in the MIMO case and $k \in \{1,2,\dots,n-1\}$ in the MISO case. t_{ki} and s_{ki} are the translation and dilation parameters which are independent of each other. $\boldsymbol{t}$ and **s** are matrices of size $S(U\text{-}O) \times (m+n)$, where $S(U\text{-}O)$ is the size of the set $U\text{-}O$ in the MIMO system and $(n\text{-}1) \times (m+n)$ in case of MISO. Here we consider a multidimensional wavelet as the tensor product of a one dimensional wavelet $\psi_s(x)$.

Now that we have specified the dynamics of the network, we need to consider how we might adapt the parameters in order to improve its performance over time. The fundamental approach to deriving such an adaptation scheme, just as with the more familiar back propagation algorithm, is to specify some measure of network performance and compute its gradient in the parameter space.

Let's define a time varying error measure e by:

$$e_k(t+1) = \begin{cases} y_k(t+1) - \hat{y}_k(t+1) & k \in O \ \textit{or} \ k = n \\ 0, & \textit{else} \end{cases} \tag{7}$$

where $y_k(t+1)$ and $\hat{y}_k(t+1)$ are the specified target value and actual value for *kth* node at time t+1, respectively. Now let $\Theta(t+1)$ denote the network error at time t+1 (where $k \in O$ in the MIMO case or $k = n$ for the MISO system) and define as follows:

$$\Theta(t+1) = \frac{1}{2p} \sum_{k=1}^{p} \left[e_k(t+1)\right]^2. \tag{8}$$

Now, we would like to minimize the above performance factor in the parameter space spanned by $\boldsymbol{w}$, $\boldsymbol{t}$ and $\boldsymbol{s}$. For this we compute the gradient of $\Theta(t+1)$ in the parameter space. In other words, for each parameter **w, *t,*** and **s** in the network, we get the incremental changes as:

$$\begin{aligned} \Delta w_{ij} &= -\alpha \frac{\partial \Theta(t+1)}{\partial w_{ij}} = \frac{\alpha}{p} \sum_{k \in O} e_k(t+1) \frac{\partial}{\partial w_{ij}} y_k(t+1) \\ \Delta t_{ij} &= -\beta \frac{\partial \Theta(t+1)}{\partial t_{ij}} = \frac{\beta}{p} \sum_{k \in O} e_k(t+1) \frac{\partial}{\partial t_{ij}} y_k(t+1) \\ \Delta s_{ij} &= -\gamma \frac{\partial \Theta(t+1)}{\partial s_{ij}} = \frac{\gamma}{p} \sum_{k \in O} e_k(t+1) \frac{\partial}{\partial s_{ij}} y_k(t+1), \end{aligned} \tag{9}$$

where $i = \{1,2,\dots,n-p\}$, $j = \{1,2,\dots,m+n-p+1\}$ in the MIMO case and $i = \{1,2,\dots,n-1\}$, $j = \{1,2,\dots,m+n\}$ in the MISO case. Also, α, β, and γ are the learning rates. Since the value of $e_k(t+1)$ is known at time t+1**,** for each $k \in O$**,** all that remains is to find a way to compute the remaining factors:

$$\frac{\partial y_k(t+1)}{\partial w_{ij}}, \frac{\partial y_k(t+1)}{\partial t_{ij}} \ \textit{and} \ \frac{\partial y_k(t+1)}{\partial s_{ij}}. \tag{10}$$

For calculating (9) we should compute differentiating (4) and (6) with respect to w_{ij}, t_{ij} and s_{ij}. Then, we can create a dynamical system with variables $p_{ij}^k(t), q_{ij}^k(t)$ *and* $r_{ij}^k(t)$ for $k \in U$. Therefore, we can see this dynamical system in following equation:

$$\begin{aligned} \frac{\partial}{\partial w_{ij}} y_k(t+1) &= p_{ij}^k(t+1) \\ \frac{\partial}{\partial t_{ij}} y_k(t+1) &= q_{ij}^k(t+1) \\ \frac{\partial}{\partial s_{ij}} y_k(t+1) &= r_{ij}^k(t+1). \end{aligned} \tag{11}$$

Since we assume that the initial state of the network has no functional dependence on the parameters we have:

$$p_{ij}^{k}(t_0) = q_{ij}^{k}(t_0) = r_{ij}^{k}(t_0) = 0. \qquad (12)$$

Based on the algorithm, we should compute the variables p,q,r and e_k at each time step t+1 to compute the parameter changes given by (9).

V. SIMULATIONS

To validate the effectiveness of our fully recurrent wavelet network, we have conducted extensive computer simulations on dynamic system identification problems. Here, we present two examples of dynamic nonlinear system identification: a SISO system and a MIMO plant. Also in the first example, the simulation results are compared with a FRNN. Also, we consider the first derivative of Gaussian function as a mother wavelet function as $\psi(x) = -x\exp(-\frac{1}{2}x^2)$.

A. Identification of the SISO System

The nonlinear dynamical system considered in this section is described by the following difference equation [4]:

$$y_p(k+1) = f[y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)],$$

where the unknown function has the form

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

In the identification model the network had n = 3 and m = 2 (including the bias of -1). α, β, and γ were chosen to be equal to 0.05. The RTRL algorithm is applied for learning of the parameter values of FRWNN with random initial conditions. The total number of the network parameters is 25. A random input signal between [-1 +1] is applied as the training signal. Fig. 3 and Fig. 4 show the outputs of the FRWNN for two testing inputs $u_1(k)$ and $u_2(k)$, respectively, where the inputs are as follows:

$$u_1(k) = \begin{cases} \sin(2\pi k/250) & if \;\; k \le 500 \\ 0.8\sin(2\pi k/250) + 0.2\sin(2\pi k/25) & \\ \quad if \;\; 500 < k \le 1000 & \end{cases}$$

$$u_2(k) = \begin{cases} \sin(\pi k/25) & if \;\; k < 250 \\ 1.0 & if \;\; 250 \le k < 500 \\ -1.0 & if \;\; 500 \le k < 750 \\ 0.3\sin(\pi k/25) + 0.1\sin(\pi k/32) & \\ +\, 0.6\sin(\pi k/10) & if \;\; 750 \le k \le 1000 \end{cases}$$

In this example, the FRWNN identifier is compared with a FRNN identifier that the structures of both are similar. The activation function of FRNN is a Gaussian function as $f(x) = 1/(1+\exp(-a(x-b)))$ where b and a are crossover point and slope in crossover point, respectively. The parameters of Gaussian function of FRNN are considered to be adjustable because of the better comparison between FRWNN and FRNN identifiers. Fig. 5 and Fig. 6 depict the outputs of the FRNN for the testing signals $u_1(k)$ and $u_2(k)$, respectively. We can see that the FRWNN performs better than FRNN.

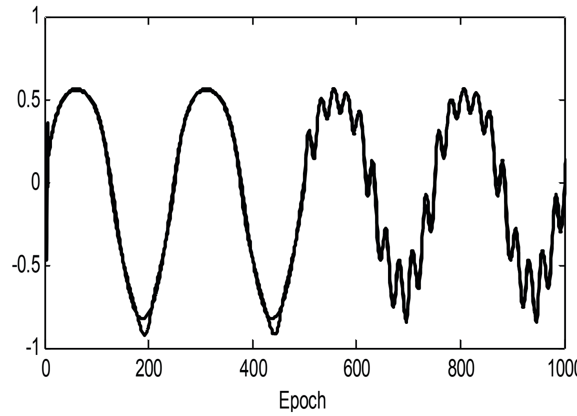

Fig. 3. The output response of SISO system using FRWNN identifier for u_1(k) (solid: Target data, dotted: Tested data).

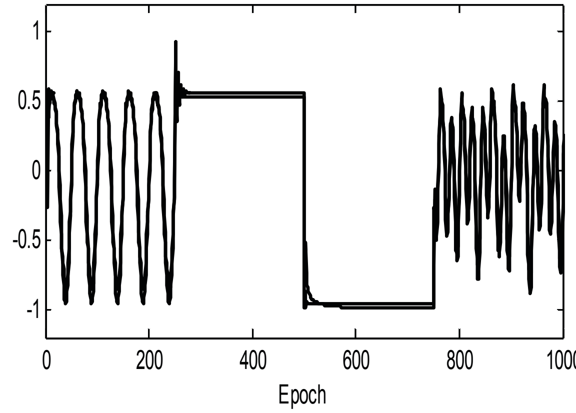

Fig. 4. The output response of SISO system using FRWNN identifier for u_2(k) (solid: Target data, dotted: Tested data).

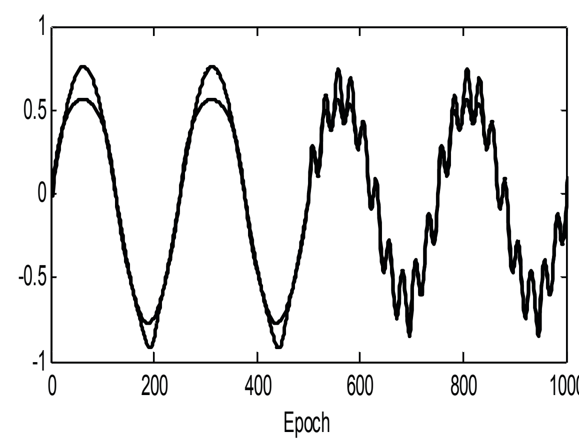

Fig. 5. The output response of SISO system using FRNN identifier for $u_1(k)$ (solid: Target data, dotted: Tested data).

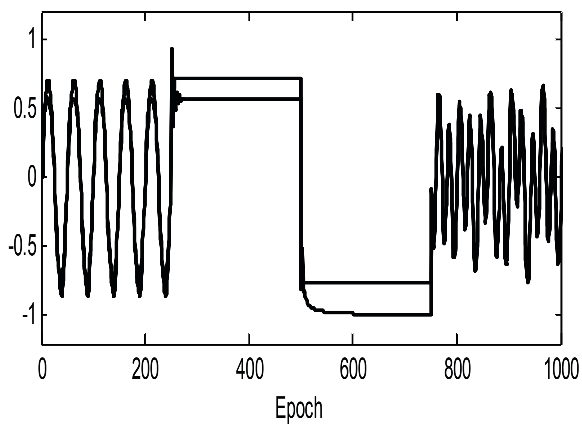

Fig. 6. The output response of SISO system using FRNN identifier for $u_2(k)$ (solid: Target data, dotted: Tested data).

B. Identification of the MIMO System

This is a MIMO plant with two inputs and two outputs. The plant is specified by [4]:

$$y_{p1}(k+1) = 0.5\left[\frac{y_{p1}(k)}{(1+y_{p2}^2(k))} + u_1(k)\right]$$
$$y_{p2}(k+1) = 0.5\left[\frac{y_{p1}(k)y_{p2}(k)}{(1+y_{p2}^2(k))} + u_2(k)\right]$$

This example is utilized to demonstrate the capability of the proposed network in identifying MIMO plants. In the identification model the network had $n = 4$ and $m = 2$. α, β, and γ were chosen to be equal to 0.1. The RTRL algorithm is applied for learning of the parameter values of FRWNN with random initial conditions. The total number of the network parameters is 34. First, we train the network for 1000 epochs with random inputs $u_1(k)$ and $u_2(k)$ uniformly distributed in the interval [-2 +2]. After the model is identified, the responses of the plant and the identification model for the following testing input signals (case1 and case2) are shown in Fig. 7 and Fig. 8.

$$case1: \begin{cases} u_1(k) = \sin(2\pi k/25) \\ u_2(k) = \cos(2\pi k/25) \end{cases}$$

$$case2: \begin{cases} u_1(k) = 0.5\sin(\pi k/25) \\ u_2(k) = \begin{cases} 0.5\sin(\pi k/25), & k < 250 \\ 0.3\sin(\pi k/25) + 0.1\sin(\pi k/32) \\ +0.6\sin(\pi k/10), & else \end{cases} \end{cases}$$

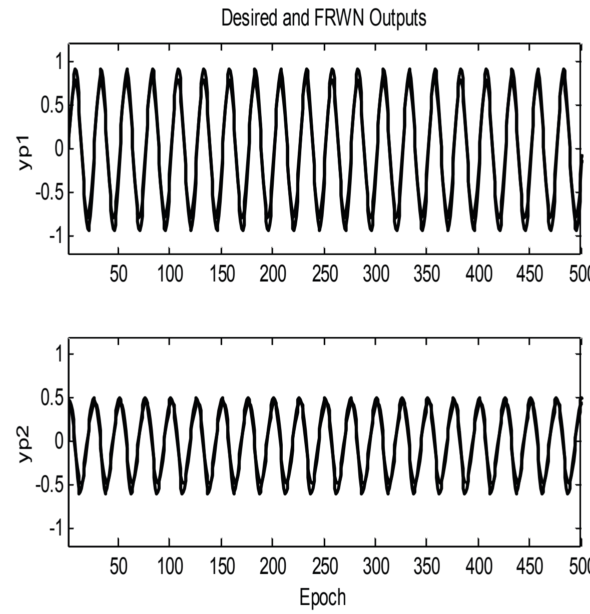

Fig. 7. The output responses of the MIMO system using FRWNN identifier for case1 (solid: Target data, dotted: Tested data).

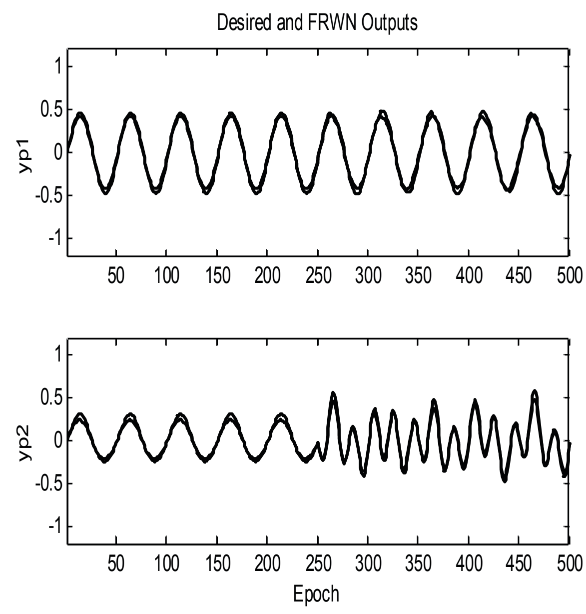

Fig. 8. The output responses of a MIMO system using FRWNN identifier for case2 (solid: Target data, dotted: Tested data).

Finally, compared to [17], and [20], we obtain the better simulation results for SISO nonlinear dynamic system identification and presented FRWNN is capable of accurately identifying MIMO nonlinear dynamic systems. Also, compared to the FRNN, the FRWNN has a less error and better performance.

VI. CONCLUTION

In this paper, a MIMO fully recurrent wavelet neural network is used to identify the SISO and MIMO nonlinear dynamic systems. The presented identifier combines the properties of the RNN and the basic ability of the WNN such as fast convergence and localization properties. The RTRL algorithm is applied to adjust the shape of wavelet functions and the connection weights. The obtained results of simulations demonstrate that the identifier is effective in identification of nonlinear dynamic systems. Also, compared to a fully recurrent neural network, the FRWNN has a less error and better performance.

REFERENCES

[1] F. Abdollahi, H. A. Talebi, and R. V. Patel, "Stable identification of nonlinear systems using neural network: theory and experiments," *IEEE/ASME Transactions on mechatronics*, vol. 11, no. 4, pp. 488-495, 2006.

[2] M. Wlas, Z. Krzeminski, and H. A. Tolivat, "Neural network based parameter estimations of induction motors," *IEEE Transactions on industrial electronics*, vol. 55, no. 4, pp. 1783-1794, 2008.

[3] D. T. Pham, and X. Liu, "Dynamic system identification using partially recurrent neural networks," *Journal of Systems Engineering*, 2, (2), pp. 90-97, 1992.

[4] K. S. Narendra, and K. Parthasarathy, "Identification and control of dynamical system using neural networks," *IEEE Trans. Neural Network*, vol. 1, no. 1, pp. 4-27, Mar. 1990.

[5] H. A. Talebi, K. Khorasani, and S. Tafazoli, "A recurrent neural network based sensor and actuator fault detection and isolation for nonlinear systems with application to the satellite's attitude control subsystem," *IEEE Transactions on neural networks*, vol. 20, no. 1, pp. 45-60, 2009.

[6] A. Kalinli, and S. Sagiroglu, "Elman network with embedded memory for system identification," *Journal of Information Science and Engineering*, vol. 22, pp. 1555-1568, 2006.

[7] Y. P. Chen, and J. S. Wang, "An automated Hammerstein recurrent neural network for dynamic applications," *in Proceeding of IASTED International Conference on Computational Intelligence*, pp. 193-198, 2005.

[8] J. S. Wang, and Y. P. Chen, "A fully automated recurrent neural network for unknown dynamic system identification and control," *IEEE Transactions on Circuits and Systems-I*, vol. 53, pp. 1363-1372, 2006.

[9] Q. Zhang, and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Network*, vol. 3, no. 6, pp. 889-898, Nov. 1992.

[10] S. H. Ling, H. H. C. Iu, F. H. F. Leung, and K. Y. Chan, "Improved hybrid particle swarm optimized wavelet neural network for modeling the development of fluid dispensing for electronic packaging," *IEEE Trans. Ind. Electron*., vol. 55, no. 9, pp. 3447-3460, Sep. 2008.

[11] T. Wang, and Y. Sugai, "A wavelet neural network with OLS algorithm for the approximation of nonlinear multivariable functions," *Proceedings of the IEICE General Conference (Institute of Electronics, Information and Communication Engineers)*, vol. 2000, page. 22, 2000.

[12] C. Pan, W. Chen, and Y. Yun, "Fault diagnostic method of power transformers based on hybrid genetic algorithm evolving wavelet neural network," *IEEE Transactions on electronic power application*, vol. 2, no. 1, pp. 71-76, 2008.

[13] A. Karami, H. R. Karami, P. Jabehdar Maralani, and B. Moshiri, "Intelligent optimal control of robotic manipulators using wavelets," *International journal of wavelets, multi-resolution and information processing*, vol. 6, pp. 575-592, 2008.

[14] M. Yarahmadi, S. M. Karbassi, and A. Mirzaei, "Robust wavelet sliding-sode control via time-variant sliding function," *IEICE transaction on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E93-A, no. 6, pp. 1181-1189, 2010.

[15] J. X. Xu, and Y. Tan, "Nonlinear adaptive wavelet control using constructive wavelet networks," *IEEE Transactions on neural networks*, vol. 18, no. 1, pp. 115-127, 2007.

[16] M.O. 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.

[17] S. S. Rao, and B. Kumthekar, "Recurrent wavelet networks," *IEEE world congress on computational intelligence*, vol. 5, pp. 3143-3147, 1994.

[18] C. H. Lu, "Design and application of stable predictive controller using recurrent wavelet neural networks," *IEEE Transactions on industrial electronics*, vol. 56, no. 9, pp. 3733-3742, 2009.

[19] A. Ghadirian, and M. Zekri, "Control of nonlinear dynamic systems using recurrent wavelet neural network," *Accepted in 4th International Conference on Intelligent Information Technology Application (IEEE-IITA2010)*, to be published.

[20] S. He, and Z. He, "Blind equalization of nonlinear communication channels using recurrent wavelet neural networks," *IEEE,* pp. 3305-3308, 1997.

[21] T. Kugarajah, and Q. Zhang, "Multidimensional wavelet frame," *IEEE Trans. Neural Networks*, vol. 6, no. 6, pp. 1552-1556, 1995.

[22] R. J. Williams, and David Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation,* vol. 1, MIT Press, 1989.