

## APPROXIMATION OF FEEDBACK LINEARIZATION CONTROL INPUT BASED ON FUZZY WAVELET NETWORKS\*

M. ZEKRI\*\*, S. SADRI AND F. SHEIKHOESLAM

Dept. of Electrical and Computer Engineering, Isfahan University of Technology, I. R. of Iran  
Email: m\_zekri@ec.iut.ac.ir

**Abstract**– Based on the wavelet transform and fuzzy set theory, we present a fuzzy wavelet network (FWN) for approximating feedback linearization control input. Each fuzzy rule corresponds to a sub-wavelet neural network (sub-WNN) consisting of wavelets with a specified dilation value. The degree of contribution of each sub-WNN can be controlled flexibly. The constructed rules used to approximate the control signal in which the mathematical model of the system under control is unknown can be adjusted by learning the translation parameters of the selected wavelets and determining the shape of Gaussian membership functions of a fuzzy system. The proposed FWN shows good approximation accuracy and fast convergence. Finally a nonlinear inverted pendulum system is applied to verify the effectiveness and ability of the proposed network.

**Keywords**– Wavelet neural network, fuzzy wavelet network, feedback linearization control input

### 1. INTRODUCTION

In recent years wavelet networks have found many applications in system identification, signal processing and function approximation [1-3].

A wavelet network is a nonlinear regression structure that represents input-output mappings by dilated and translated versions of a single function (mother wavelet) which is localized both in the space and frequency domains. Wavelet networks can approximate complex functions to an arbitrary precision [1, 4, 5]. An appropriate initialization of wavelet neural networks allows fast convergence to be obtained. Orthogonal Least Square (OLS) algorithm can fulfill this aim [6]. The optimal dilations of wavelets increase training speed and ensure fast convergence.

Fuzzy wavelet networks (FWNs) based on multi-resolution analysis (MRA) theory and the Takagi-Sugeno-Kang (TSK) model [7, 8] not only reserve the multi-resolution capability of WNN, but also have the advantages of high approximation accuracy and good generalization performance [9]. According to these advantages, FWN can be applied to the problems of function approximation, system identification and control [9-11].

There is no general method to control nonlinear systems. Feedback linearization technique is a suitable and efficient method for controlling a special group of nonlinear systems [12, 13]. In the case that the mathematical model of the controlled system is unknown, the approximation of feedback linearization control input is an important issue.

In this research, we present a FWN for approximating the control input. The proposed network shows high approximation accuracy and fast convergence. This paper is organized as follows. Sections 2 and 3 discuss wavelet neural network and fuzzy wavelet network, respectively. Then a FWN is proposed for

---

\*Received by the editors October 22, 2006; final revised form September 26, 2007.

\*\*Corresponding author

approximating the control signal in Section 4. A simulation example is provided to verify the performance and capability of the network in Section 5. Finally a brief conclusion is drawn in Section 6.

### 2. WAVELET NEURAL NETWORK

Wavelet neural networks use a three-layer structure and wavelet activation functions. The structure of a wavelet neural network with one output  $y$ ,  $q$  inputs  $\{x_1, x_2, \dots, x_q\}$  and  $k$  nodes is given in Fig. 1 [1, 5].

The output signal of this network is calculated as

$$y = \sum_{i=1}^k \omega_i \psi_{a_i, \vec{b}_i}(\vec{x}) \tag{1}$$

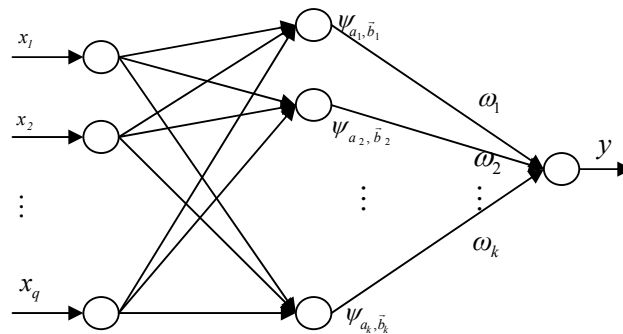


Fig. 1. Architecture of WNN

where  $\vec{x} = [x_1, x_2, \dots, x_q]^T$  is the vector of inputs,  $\omega_i$   $i = 1, 2, \dots, k$  are weight coefficients between hidden and output layers, and  $\psi_{a_i, \vec{b}_i}$  are dilated and translated versions of a mother wavelet function  $\psi : \mathfrak{R}^q \rightarrow \mathfrak{R}$  :

$$\psi_{a_i, \vec{b}_i}(\vec{x}) = a_i^{-q/2} \psi\left(\frac{\vec{x} - \vec{b}_i}{a_i}\right) \tag{2}$$

Naturally the mother wavelet  $\psi$  is required to have zero mean.

$$\int_{\mathfrak{R}^q} \psi(\vec{x}) d\vec{x} = \int_{\mathfrak{R}^q} \psi(x_1, x_2, \dots, x_q) dx_1 dx_2 \dots dx_q = 0 \tag{3}$$

and also  $|\psi(\vec{x})|$  and  $|\hat{\psi}(\vec{\omega})|$  rapidly decay to zero as  $\|\vec{x}\| \rightarrow \infty$  and  $\|\vec{\omega}\| \rightarrow \infty$ .

In Eq. (2), the dilation parameter  $a_i \in \mathfrak{R}_+$  controls the support of the wavelet and the translation parameter  $\vec{b}_i \in \mathfrak{R}^q$  determines its central position.

In Eq. (1), if pairs  $(a_i, \vec{b}_i)$  are taken from a grid  $\Lambda$  given by

$$\Lambda = \{(\alpha^n, \vec{m}\beta\alpha^n) : n \in \mathbb{Z}, \vec{m} \in \mathbb{Z}^q\} \tag{4}$$

where the scalar parameters  $\alpha$  and  $\beta$  define the step sizes of dilation and translation discretizations, (typically  $\alpha = 2$  ,  $\beta = 1$ ) respectively, according to the above definitions any function  $f \in L^2(\mathfrak{R}^q)$  (finite-energy and continuous or discontinuous) can be approximated to an arbitrary precision by the wavelet neural network [1, 5].

The wavelets that Zhang and Benveniste [5] and Pati and Krishnaprasad [14] used in their networks are wavelet frames. Orthogonal bases are used in other wavelet networks structures [15].

In [16], a number of methods are available to construct multidimensional wavelets in both single-scaling and multi-scaling forms, based on one-dimensional wavelet frames. In the single-scale multidimensional wavelet frames, a single dilation parameter is used in all the dimensions of each wavelet, and multidimensional wavelet frames can be built by using a single mother wavelet. To fulfill this purpose the radial functions are used.

The single-scaling radial wavelet frames (with simpler structure than multiscaling wavelet frames) in [6] are defined as

$$F = \left\{ \psi_{\vec{m},n}(\vec{x}) = \alpha^{-\frac{1}{2}qn} \psi(\alpha^{-n}\vec{x} - \beta\vec{m}) \quad : \quad n \in Z, \vec{m} \in Z^q \right\} \quad (5)$$

where  $\psi : \mathfrak{R}^q \rightarrow \mathfrak{R}$  is a radial wavelet function (Like Mexican Hat wavelet) and  $q$  is the input dimension.

In the multi-scaling multidimensional wavelet frames, an independent dilation parameter is used in each dimension, and the multidimensional wavelet frames can be built by a tensor product of one dimension (1-D) wavelet functions. Using a 1-D wavelet frame  $\psi_s : \mathfrak{R} \rightarrow \mathfrak{R}$ , a multi-scaling wavelet frame  $\psi : \mathfrak{R}^q \rightarrow \mathfrak{R}$  is built by setting

$$\psi(\vec{x}) = \psi_s(x_1) \dots \psi_s(x_q) \quad , \quad for \quad \vec{x} = (x_1, \dots, x_q) \quad (6)$$

According to the theory of MRA, the dilation parameter of a wavelet can be interpreted as the resolution parameter. In fact, based on the multiresolution property, it is possible to present a library of wavelets. The wavelets with coarse resolution can describe the global (low frequency) behavior and those with fine resolution can describe the local (higher frequency) behavior of the approximated function. Accordingly, compared to other functional approximators, WNNs have the advantages of fast convergence, easy training and high accuracy [9].

### 3. FWN STRUCTURE

In the Takagi-Sugeno-Kang (TSK) model, a set of fuzzy rules can be described by

$$R^i : IF \ x_1 \text{ is } A_1^i, \text{ and } x_2 \text{ is } A_2^i \text{ and } \dots \text{ and } x_q \text{ is } A_q^i \text{ THEN } \hat{y}_i = \sum_{j=1}^q k_j x_j \quad (7)$$

where  $A_j^i$  is membership function,  $\vec{x} = (x_1, x_2, \dots, x_q)^T$  is input vector and  $\hat{y}_i$  is output variable and a linear combination of inputs which is essentially a global function.

In the FWN structure,  $\hat{y}_i$  is the output of the local model for rule  $R^i$  which is equal to the linear combination of a finite set of wavelets with the same dilation parameter.

A FWN is described by a set of fuzzy rules which have the following form [9]

$$R^i : IF \ x_1 \text{ is } A_1^i, \text{ and } x_2 \text{ is } A_2^i \text{ and } \dots \text{ and } x_q \text{ is } A_q^i \text{ THEN } \hat{y}_i = \sum_{k=1}^{T_i} \omega_{M_i, \vec{t}_i^k} \psi_{M_i, \vec{t}_i^k}^{(k)}(\vec{x}) \quad (8)$$

$$M_i \in Z, \vec{t}_i^k \in \mathfrak{R}^q, \omega_{M_i, \vec{t}_i^k} \in \mathfrak{R} \text{ and } \vec{x} \in \mathfrak{R}^q$$

Assuming the number of fuzzy rules is  $c$ ;  $1 \leq i \leq c$ ;  $T_i$  is the total number of wavelets for the  $i$ th rule and  $\hat{y}_i$  is the output of the local model for rule  $R^i$ .  $\vec{t}_i^k = [t_1^k, t_2^k, \dots, t_q^k]$ , where  $t_j^k \in \mathfrak{R}$  is the translation value for corresponding wavelet  $k$ .  $A_j^i$  is Gaussian-type membership function defined as

$$A_j^i(x_j) = \exp[-(((x_j - p_{j1}^i) / p_{j2}^i)^2)^{\frac{p_{j3}^i}{2}}] \quad (9)$$

$$p_{j1}^i, p_{j2}^i \in \mathfrak{R}, 0 < p_{j3}^i \leq 5$$

In (9),  $p_{j1}^i$  and  $p_{j2}^i$  represent the center and the width of the Gaussian membership function and  $p_{j3}^i$  determines the shape of the membership function.

Wavelets  $\psi_{M_i, \vec{t}}^{(k)}(\vec{x})$  are expressed by the tensor product of 1-D wavelet functions:

$$\begin{aligned} \psi_{M_i, \vec{t}}^{(k)}(\vec{x}) &= 2^{M_i/2} \psi^{(k)}(2^{M_i} \vec{x} - \vec{t}^k) \\ &= \prod_{j=1}^q 2^{M_j/2} \psi^{(k)}(2^{M_j} x_j - t_j^k) \end{aligned} \quad (10)$$

According to Eq.(10), in each rule or sub-WNN, the wavelets are single-scaling and the same dilation index is in all the dimensions.

Output of the whole network is calculated as

$$\hat{y} = \sum_{i=1}^c \hat{\mu}_i(\vec{x}) \hat{y}_i \quad (11)$$

where  $\hat{\mu}_i(\vec{x}) = \frac{\mu_i(\vec{x})}{\sum_{i=1}^c \mu_i(\vec{x})}$ ,  $\mu_i(\vec{x}) = \prod_{j=1}^q A_j^i(x_j)$

$\hat{\mu}_i$  determines the degree of contribution of each sub-WNN in the output of the network and  $\hat{y}_i$  is the output of the local model for rule  $R^i$ .

Notice that, in THEN-parts of fuzzy rules, FWN employs  $c$  sub-WNNs at different resolution levels to capture different behaviors (global or local) of the approximated function rather than using constants or linear equations as in the traditional fuzzy models. Unlike the traditional FNNs with only one localized approximation of function, the FWN uses both globalized and localized approximation of function. For this reason, the FWN inspired by both fuzzy model and WNN has the advantages of improved local accuracy, nicer generalization capability and faster convergence [9].

#### 4. APPROXIMATING FEEDBACK LINEARIZATION CONTROL INPUT BY FWN

In this section, motivated by the reason stated in the previous section, we present a fuzzy wavelet network that can approximate the feedback linearization control input with high accuracy for a large group of nonlinear systems.

Consider the following  $q$ th order dynamical nonlinear system:

$$\begin{cases} \dot{x}^{(q)} = f(\vec{x}) + g(\vec{x})u \\ y = x \end{cases} \quad (12)$$

$$y = x \quad (13)$$

where  $f$  and  $g$  are unknown functions,  $u$  and  $y$  are the control input and the system output respectively, and  $\vec{x} = (x_1, x_2, \dots, x_q)^T = (x, \dot{x}, \dots, x^{(q-1)})^T$  is the state vector of the system which is assumed to be available for measurement.

Define the tracking error  $e$  as

$$e = y_r - y = y_r - x \quad (14)$$

where  $y_r$  is the desired output. According to Eqs. (12) and (13) the feedback linearization control input is

$$u^* = \frac{1}{g(\bar{x})} \left[ -f(\bar{x}) + y_r^{(q)} + \bar{c}^T \bar{e} \right] \quad (15)$$

where  $\bar{c} = (c_1, c_2, \dots, c_q)^T$  is a positive constant vector and  $\bar{e} = (e_1, e_2, \dots, e_q)^T = (e, \dot{e}, \dots, e^{(q-1)})^T$  is the error vector.

Substituting Eq. (15) into Eq. (12), the error equation can be obtained as

$$e^{(q)} + c_q e^{(q-1)} + \dots + c_1 e = 0 \quad (16)$$

The elements in vector  $\bar{c}$  can be chosen appropriately such that all the roots of error equation are in the open left-half complex plane. Thus the controlled system is stabilized if the control input, Eq. (15), can be implemented. In practice, both  $f$  and  $g$  are unknown functions and so it is hard to implement Eq. (15). To solve the problem, a fuzzy wavelet network is proposed to approximate the feedback linearization control input.

A typical fuzzy wavelet network for the approximation of (15) can be described by a set of fuzzy rules:

$R^i$  (ith rule):

$$\text{IF } x_1 \text{ is } A_1^i, \text{ and } x_2 \text{ is } A_2^i \text{ and } \dots \text{ and } x_q \text{ is } A_q^i \text{ THEN } \hat{u}_i = \sum_{k=1}^{T_i} \omega_{M_i, \bar{t}_i^k} \psi_{M_i, \bar{t}_i^k}^{(k)}(\bar{x}) \quad (17)$$

where  $x_j (1 \leq j \leq q)$  is the  $j$ th state variable of  $\bar{x}$  in the controlled system (12), and  $\hat{u}_i$  is the control signal (output) of the local model for  $i$ th rule. The structure of fuzzy wavelet network for approximating control input (15) is given in Fig. 2 and the structure of each WNN is given in Fig. 1.

According to Eqs.(9) and (10), the output of the network in Fig. 2 is calculated as

$$\hat{u} = \sum_{i=1}^c \hat{\mu}_i(\bar{x}) \hat{u}_i \quad (18)$$

where  $\hat{\mu}_i(\bar{x}) = \frac{\mu_i(\bar{x})}{\sum_{i=1}^c \mu_i(\bar{x})}$

$\hat{\mu}_i$  determines the degree of contribution of the control signal (output) of the wavelet based model with resolution level  $M_i$ .

After calculating the feedback linearization control input of FWN, the training of the network starts.

Consider the training data set of the nominal model system:

$$\left\{ (\bar{x}_l^d, u_l^d) \right\} \quad 1 \leq l \leq L \quad \bar{x}_l^d \in R^q, u_l^d \in \mathfrak{R} \quad (19)$$

where  $L$  is the total number of the training patterns. Our goal is to train the network in Fig. 2 based on data set in (19) so that the error between the output of FWN and  $\bar{u}^d$  is minimized.

Most work done in the wavelet networks uses simple wavelets. For fuzzy wavelet network in [9, 10], Mexican Hat wavelet and B-spline wavelets are used respectively. We have selected the Mexican Hat function  $\psi(x) = (1 - x^2) \exp(-\frac{x^2}{2})$  as the wavelet function and accordingly, based on data set, dilation values  $M_i$  is chosen to be in the range from -3 to 4.

After determining the dilation parameters, wavelet candidates are selected based on the input space training data. Wavelet candidates with various translation parameters, which are selected according to

input training data (desired state variables), are often redundant for constructing FWN [9]. Thus the OLS (Orthogonal Least Square) algorithm can be used for purifying wavelet candidates [6].

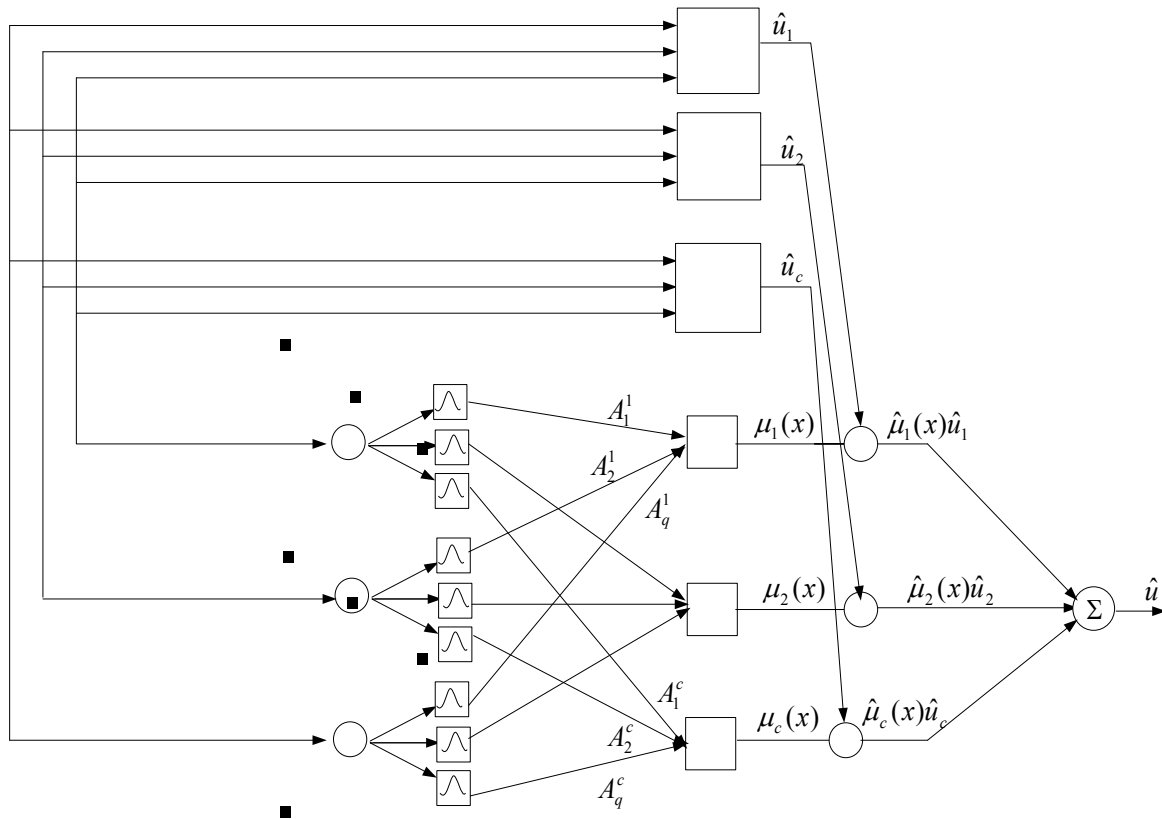


Fig. 2. Structure of fuzzy wavelet network for approximating control signal

The set of wavelet candidates  $W = \{\psi_1, \psi_2, \dots, \psi_{L_w}\}$  are a set of regressor vectors which construct the output of the network. In this step, the number of wavelet functions,  $S$ , from set  $W$  that best span the training data of the control signal are selected. Since these regressors are usually correlated, the degree of contribution of each regressor to the output energy is not clear. The OLS algorithm transforms the set of regressor vectors into a set of orthogonal basis vectors and it is possible to calculate the degree of contribution of each basis vector to the output energy [17].

From the initial set of wavelet candidates, the OLS algorithm selects the wavelet that best fits the training data, and then repeatedly selects one of the remaining wavelets that best fits the data when combined with all previously selected wavelets. For computational efficiency, later selected wavelets are orthonormalized to earlier selected ones.

Using  $S$  selected wavelets, the approximated control signal is expressed as

$$\hat{u}_s = \sum_{i=1}^S \omega_i \psi_i(\bar{x}) \tag{20}$$

where  $\{l_1, l_2, \dots, l_S\}$  is a subset of  $\{1, 2, \dots, L_w\}$ . According to a set of training data as defined in (19), we can write

$$\vec{u} = \Psi \vec{\omega} + \vec{e}_u \tag{21}$$

where

⋮

$$\Psi = \begin{bmatrix} \psi_{l_1}(\bar{x}_1) \cdots \psi_{l_s}(\bar{x}_1) \\ \vdots \quad \quad \quad \vdots \\ \psi_{l_1}(\bar{x}_L) \cdots \psi_{l_s}(\bar{x}_L) \end{bmatrix}$$

$$\bar{\omega} = (\omega_{l_1}, \omega_{l_2}, \dots, \omega_{l_s})^T$$

$$\bar{u} = (u_1, u_2, \dots, u_L)^T$$

$$\bar{e}_u = (e_{u_1}, e_{u_2}, \dots, e_{u_L})^T$$

$\bar{e}_u$  is the error vector of the approximation problem defined by the training data (19). Our goal is to minimize the sum of square errors  $\bar{e}_u^T \bar{e}_u = \sum_{i=1}^L e_{u_i}^2$ . The computation of  $\bar{\omega}$  can be performed by the least squares method. Let us define the following notations:

$$\bar{p}_i = [\psi_{l_1}(\bar{x}_1), \dots, \psi_{l_i}(\bar{x}_L)]^T \quad \text{for } i = 1, 2, \dots, L_w$$

Then  $\Psi$  is expressed as

$$\Psi = [\bar{p}_1, \bar{p}_2, \dots, \bar{p}_{l_s}] \tag{22}$$

Since generally the vectors  $\bar{p}_1, \bar{p}_2, \dots, \bar{p}_{L_w}$  are not mutually orthogonal, the classical or modified Gram-Schmidt algorithm can be applied to decompose  $\Psi$  as

$$\Psi = Q A \tag{23}$$

where  $Q = [\bar{q}_1, \bar{q}_2, \dots, \bar{q}_{l_s}]$  is a  $L \times S$  matrix with orthogonal columns and  $A$  is an  $S \times S$  upper triangular matrix with 1's on the diagonal, that is

$$A = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1S} \\ 0 & 1 & \alpha_{23} & \cdots & \alpha_{2S} \\ 0 & 0 & 1 & \cdots & \alpha_{3S} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & \alpha_{S-1S} \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

Then (21) can be rewritten as

$$\bar{u} = Q \bar{v} + \bar{e}_u \tag{24}$$

And the vector  $\bar{v} = A \bar{\omega}$  can be determined by the least squares solution

$$\bar{v} = (Q^T Q)^{-1} Q^T \bar{u} \tag{25}$$

Since the columns of  $Q$  are orthogonal, (25) becomes

$$\bar{v} = [v_{l_1}, v_{l_2}, \dots, v_{l_s}]$$

$$v_{l_i} = \frac{\bar{q}_{l_i}^T \bar{u}}{\bar{q}_{l_i}^T \bar{q}_{l_i}} \quad i = 1, 2, \dots, S$$

and the variance of  $\bar{u}$  is calculated as

$$\bar{u}^T \bar{u} = \sum_{i=1}^S v_{l_i}^2 \bar{q}_{l_i}^T \bar{q}_{l_i} + \bar{e}_u^T \bar{e}_u \tag{26}$$

It can be seen from (26) that the term  $v_{l_i}^2 \bar{q}_{l_i}^T \bar{q}_{l_i}$  is the part of the variance of  $u$  explained by regressor  $\bar{q}_{l_i}$ . A regressor is significant if this amount is large. Therefore, error reduction ratio ( $err$ ) can be defined in the following form:

$$err_i = \frac{v_{l_i}^2 \bar{q}_{l_i}^T \bar{q}_{l_i}}{\bar{u}^T \bar{u}} \quad (27)$$

The OLS method uses  $err$  to select the important wavelets which have significant contribution to the output of the network. Then  $T_i, i = 1, 2, \dots, c$  the number of selected wavelets at dilation  $M_i, \sum_{i=1}^c T_i = S$  the total number of selected wavelets and initial weights  $\omega_{M_i, \bar{i}^k}$  are determined. Details of the network initialization can be found in [9].

The EKF (Extended Kalman Filter) Method is used for training parameters  $p_{jr}^i, t_j^k$  and then LS (Least-Squares) algorithm for updating  $\omega_{M_i, \bar{i}^k}$  where  $j = 1, 2, \dots, q, r = 1, 2, 3, k = 1, 2, \dots, S$ . In [18], EKF and BP training algorithms are compared and it is shown that the EKF method has the advantages of better convergence and faster training speed.

The learning procedure will be repeated according to a performance index  $J_i$  at  $i$ th iteration which is defined as

$$J_i = \sqrt{\frac{\sum_{l=1}^L (\hat{u}_l - u_l^d)^2}{\sum_{l=1}^L (u_l^d - \bar{u})^2}}, \quad \bar{u} = \frac{1}{L} \sum_{l=1}^L u_l^d \quad (28)$$

where  $u_l^d$  is the desired control input and  $\hat{u}_l$  is the estimated output from the FWN in Fig. 2.

## 5. SIMULATION RESULTS

To demonstrate the capability of the proposed FWN we apply it to approximate the control signal for a second-order inverted pendulum system. In this simulation example, the objective is to approximate an appropriate control signal  $u$  to control the motion of the cart, such that the pole can track a desired output. Assume that  $x_1 = \theta$  is the angle of the pole with respect to the vertical axis, and  $x_2 = \dot{\theta}$  is the angular velocity of the pole. The inverted pendulum system is shown in Fig. 3.

The state equations can be expressed by

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f + gu \end{cases}$$

$$\text{where } f = \frac{g_r \sin x_1 - \frac{mLx_2^2 \sin x_1 \cos x_1}{m+M}}{L\left(\frac{4}{3} - \frac{m \cos^2 x_1}{m+M}\right)} \quad g = \frac{\cos x_1}{L\left(\frac{4}{3} - \frac{m \cos^2 x_1}{m+M}\right)}$$

$g_r$ (acceleration due to gravity) = 9.81 m/s<sup>2</sup>

$L$  (half length of the pole) = 0.5 m

$M$  (mass of the cart) = 1.0 kg

$m$  (mass of the pole) = 0.1 kg



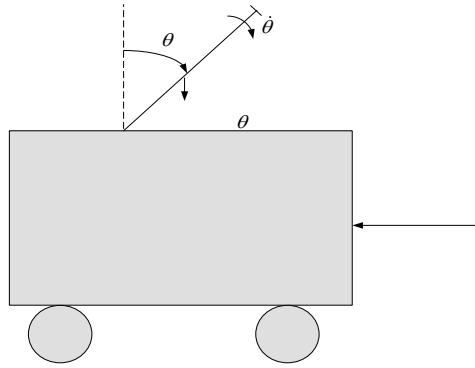


Fig. 3. Inverted pendulum system

For this system our FWN has two inputs ( $x_1, x_2$ ) and one output  $u$ . In the simulation, we use 300 pairs  $\{\bar{x}^d, u^d\}$  as training data sets of the nominal model system. By using the OLS algorithm [6], five fuzzy rules or sub-WNN are represented to approximate the control input for tracking the desired output  $y_r(t) = 0.5 \sin t$  (case 1). The number of the selected wavelets and dilation parameter value presented by the OLS method in each sub-WNN are given in Table 1. The total number of the selected wavelets is 17, and 74 unknown parameters are used for training.

In case 2, for tracking the desired output  $y_r(t) = 0.2(\sin t + \sin 2t)$ , the feedback linearization control signal is approximated by 8 rules and the total number of the selected wavelets is 28. The number of the selected wavelets and dilation parameter value presented by the OLS method in each sub-WNN are given in Table 2. The performance of FWN is shown in Table 3.

Table 1. The number of the selected wavelets and dilation value in each sub-WNN for Case 1

$M_i$ (dilation parameter value)	-4	-3	-2	-1	0	1	2
$T_i$ (number of wavelets in each sub-WNN)	3	2	2	2	2	4	1

Table 2. The number of the selected wavelets and dilation value in each sub-WNN for Case 2

$M_i$ (dilation parameter value)	-4	-3	-2	-1	0	1	2	3
$T_i$ (number of wavelets in each sub-WNN)	5	5	2	4	3	3	3	3

Table 3. The performance of proposed FWN

	$y_r(t)$	Number of rules	Number of unknown parameters used for training	Performance index ( $J_i$ )	Number of iterations
Case 1	$0.5 \sin t$	7	74	0.028	351
Case 2	$0.2(\sin t + \sin 2t)$	8	104	0.070	198

In case 1, Figs. 4 and 5 show the desired input data and approximated control signal constructed by FWN, respectively. In Fig. 4, the dot curve shows the desired output and solid curves show the input training data. In Fig. 5, the thin and thick curves are desired control signal and approximated control signal constructed by FWN, respectively. The same results are presented for case 2 in Figs. 6 and 7. In both cases, our FWN has fast convergence and appropriate accuracy.

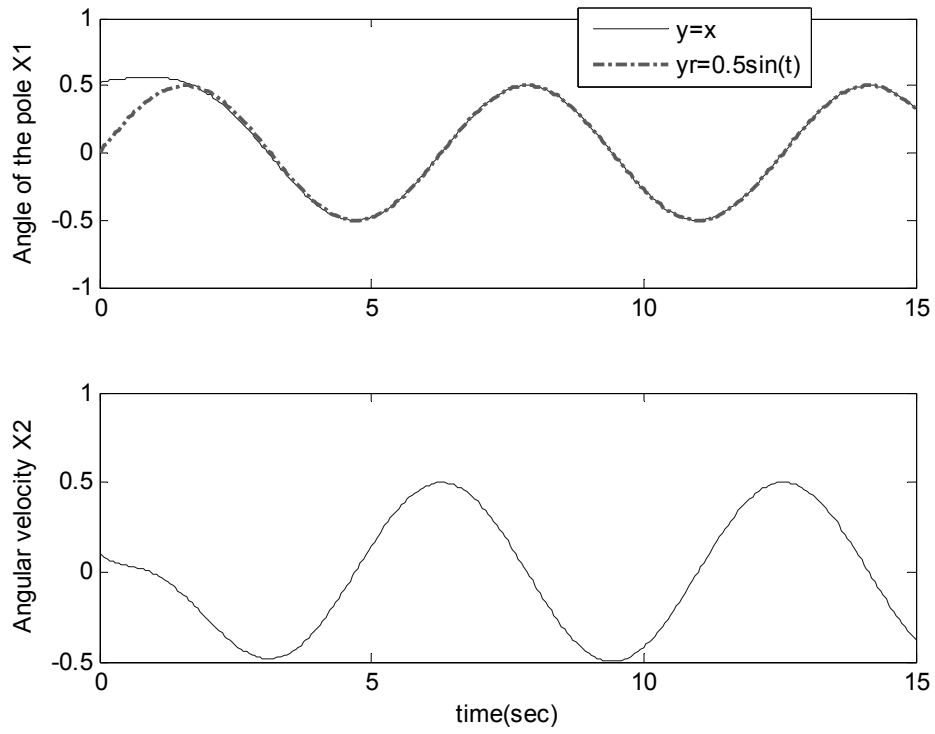


Fig. 4. Input training data  $\vec{x}_1^d, \vec{x}_0 = \left[ \frac{\pi}{6}, 0.1 \right]^T$

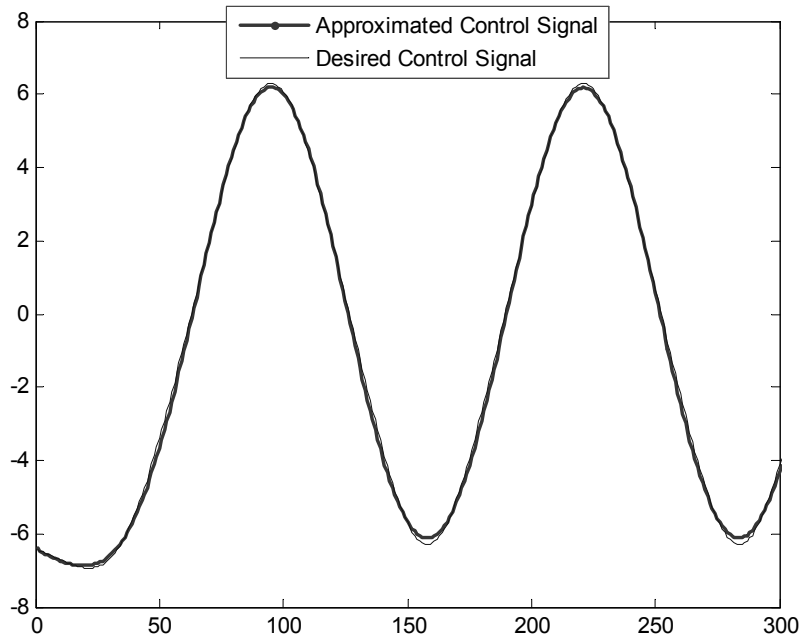


Fig. 5. Approximated control signal

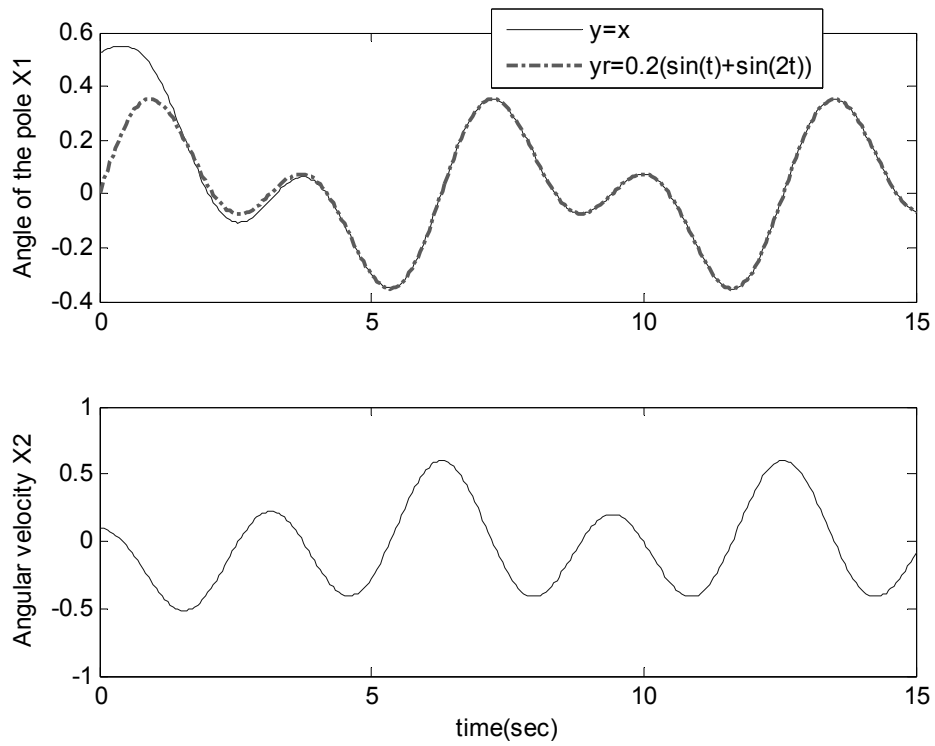


Fig. 6. Input training data  $\vec{x}_t^d, \vec{x}_0 = \left[ \frac{\pi}{6}, 0.1 \right]^T$

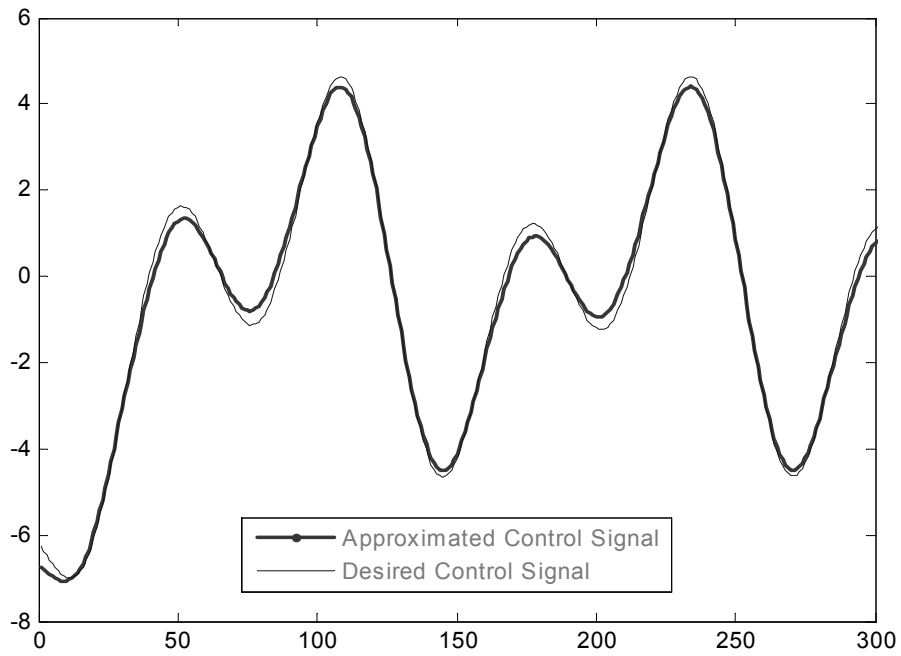


Fig. 7. Approximated control signal

## 6. CONCLUSIONS

In this paper, we presented a fuzzy wavelet network for approximating feedback linearization control input in which the mathematical model of the controlled system is unknown. The proposed FWN with the combination of WNN and fuzzy logic has advantages of the approximation accuracy and good generalization performance. Fuzzy rules correspond to sub-WNNs with different resolution levels such that the degree of contribution of various sub-WNNs can be controlled flexibly. The obtained results of simulation examples demonstrated the efficiency of the presented approach.

## REFERENCES

1. Iyengar, S. S., Cho, E. C. & Phoha, V. V. (2002). *Foundations of wavelet networks and applications*. CRC Press.
2. Billings, S. A. & Wei, H. L. (2005). A new class of wavelet networks for nonlinear system identification. *IEEE Trans. Neural Net.* Vol. 16, pp. 826-874.
3. Lin, F. J., Shieh, H. J. & Huang, P. K. (2006). Adaptive wavelet neural network control with hysteresis estimation for piezo-positioning mechanism. *IEEE Trans, Neural Net.*, Vol. 17, No. 2, pp. 432-444.
4. Chui, C. K. (1992). *An introduction to wavelets*. Academic Press, NY, U.S.A.
5. Zhang, Q. & Benveniste, A. (1992). Wavelet networks. *IEEE Trans. Neural Net.*, Vol. 3, No. 6, pp. 889-898.
6. Zhang, Q. (1997). Using wavelet network in non Parametric estimation. *IEEE Trans. Neural Networks*, Vol. 8, No. 2, pp. 227-236.
7. Jane, J. S. R., Sun, C. T. & Mizutani, E. (1997). *Neuro-fuzzy and soft computing*. Engle Wood Cliffs, NJ: Prentice-Hall.
8. Tahani, V. & Sheikholeslam, F. (1999). Total stability of fuzzy feedback control systems. *Iranian Journal of Science & Technology*, Vol. 23, No. 1, pp. 27-34.
9. Ho, D. W. C., Zhang, P. A. & Xu, J. (2001). Fuzzy wavelet networks for function learning. *IEEE Trans. Fuzzy Systems*, Vol. 9, No. 1, pp. 200-211.
10. Wang, J., Xiao, J. & Hu, D. (2005). Fuzzy wavelet network modelling with B-spline wavelet. *Proceedings of International Conference on Machine Learning and Cybernetics*, Vol. 7, pp. 4144-4148.
11. Abiyev, R. H. (2005). Controller based of fuzzy wavelet neural network for control of technological processes. *IEEE International Conference on CIMS*, pp. 215-219.
12. Slotin, J. E. & Li, W. (1991). *Applied nonlinear control*. Englewood Cliffs, NJ: Prentice Hall International.
13. Khalil, H. K. (2005). *Nonlinear systems*. Third Edition, Prentice Hall.
14. Pati, Y. & Krishnaprasad, P. (1993). Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformation. *IEEE Trans. Neural works*, Vol. 4, No. 1, pp. 73-85.
15. Zhang, J., Walter, G. G. & Lee, W. N. W. (1995). Wavelet neural networks for function learning. *IEEE Trans. Signal Processing*, Vol. 43, No. 6, pp. 1485-1497.
16. Kugarajah, Th. & Zhang, Q. (1995). Multidimensional wavelet frames. *IEEE Trans. Neural Networks*, Vol. 6, No. 6, pp. 1552-1556.
17. Chen, S., Cowan, C. F. N. & Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. Neural Networks*, Vol. 2, No. 2, 302-309.
18. Li, S. (2001). Comparative analysis of back propagation and extended kalman filter in pattern and batch forms of training neural networks. *IEEE Trans. NeuralNetworks*, Vol. 1, pp. 144-149.