ADAPTIVE FUZZY WAVELET NETWORK
CONTROL DESIGN FOR NONLINEAR SYSTEMS

DHIRAJ AHUJA and SUNIL KUMAR
M. Tech (EE) students, YMCA University of Science & Technology, Sector-6, Mathura Road, Faridabad Haryana-121 006, India

Abstract

This paper presents a new adaptive fuzzy wavelet network controller (A-FWNC) for control of nonlinear affine systems, inspired by the theory of multiresolution analysis (MRA) of wavelet transforms and fuzzy concepts. The proposed adaptive gain controller, which results from the direct adaptive approach, has the ability to tune the adaptation parameter in the THEN-part of each fuzzy rule during real-time operation. Each fuzzy rule corresponds to a sub-wavelet neural network (sub-WNN) and one adaptation parameter. Each sub-WNN consists of wavelets with a specified dilation value. The degree of contribution of each sub-WNN can be controlled flexibly. Orthogonal least square (OLS) method is used to determine the number of fuzzy rules and to purify the wavelets for each sub-WNN. Since the efficient procedure of selecting wavelets used in the OLS method is not very sensitive to the input dimension, the dimension of the approximated function does not cause the bottleneck for constructing FWN. FWN is constructed based on the training data set of the nominal system and the constructed fuzzy rules can be adjusted by learning the translation parameters of the selected wavelets and also determining the shape of membership functions. Then, the constructed adaptive FWN controller is employed, such that the feedback linearization control input can be best approximated and the closed-loop stability is guaranteed. The performance of the proposed A-FWNC is illustrated by applying a second-order nonlinear inverted pendulum system and compared with previously published methods. Simulation results indicate the remarkable capabilities of the proposed control algorithm. It is worth noting that the proposed controller significantly improves the transient response characteristics and the number of fuzzy rules and on-line adjustable parameters are reduced.

Keywords: Wavelet neural networks; Fuzzy wavelet networks; Feedback linearization control; Adaptive controller; Nonlinear systems

1. Introduction

In recent years, the wavelets have found various applications in many research areas [5 and 7]. In function approximation, the wavelet expansion presents a time–frequency localization of the signal. In wavelet transform domain most of the energy of the signal is well represented by linear combination of a finite number of wavelet basis functions. Recently, by utilizing soft computing and wavelet theory, a number of efficient techniques are represented, among which are wavelet networks [4, 9] and fuzzy wavelet networks [8, 11].

On the other hand, there are some short comings in using neural networks. The lack of theoretical interpretation of results leads to the lack of efficient constructive methods. There is no unified useful theoretical indication available for selecting the network structure (the number of layers and the number of neurons in each layer). Furthermore, the convergence of neural networks is not always guaranteed. The reason is that the output of such networks is highly nonlinear in its parameters, and the gradient decent method combined with random initializations may be stuck to a bad local minimal. To overcome the classic neural network short comings, the wavelet neural network (WNN) is proposed as an alternative to feed-forward neural networks, to approximate arbitrary nonlinear functions. Therefore, the main objectives of wavelet network can be specified as follows: (1) the “universal approximation” property is guaranteed. (2) Explicit link between the network coefficients and the wavelet transform is fulfilled and an initial guess for network parameters can be derived, by using the decomposition wavelet formula. (3) Potential achievement of the same extent of approximation

ISSN No: 2250-3536
Volume 3, Issue 1, Jan. 2013
148
with a network of reduced size. Furthermore, the wavelet networks are optimal approximators, since they require the smallest number of bits to obtain an arbitrary precision [10]. 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. The wavelet network can approximate any function to an arbitrary precision with a finite sum of wavelets. Also, the wavelet network provides an adaptive discretization of the wavelet transform by choosing influential wavelets based on a given data set most work done in the wavelet networks uses simple wavelets. Pati and Krishnaprasad [12] used in their networks are frame bases. Other wavelet networks structures use orthogonal bases. Daniel et al. [8] have proposed a fuzzy wavelet network based on multiresolution analysis (MRA) of wavelet transforms and fuzzy concepts to approximate arbitrary nonlinear functions.

2. NEURAL NETWORK

The term neural network was traditionally used to refer to a network or circuit of biological neurons.[1] The modern usage of the term often refers to artificial neural networks, which are composed of artificial neurons or nodes. Thus the term has two distinct usages:

1. Biological neural networks are made up of real biological neurons that are connected or functionally related in a nervous system (Figure 1). In the field of neuroscience, they are often identified as groups of neurons that perform a specific physiological function in laboratory analysis.

2. Artificial neural networks are composed of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons). Artificial neural networks may either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems without necessarily creating a model of a real biological system. The real, biological nervous system is highly complex: artificial neural network algorithms attempt to abstract this complexity and focus on what may hypothetically matter most from an information processing point of view. Good performance (e.g. as measured by good predictive ability, low generalization error), or performance mimicking animal or human error patterns, can then be used as one source of evidence towards supporting the hypothesis that the abstraction really captured something important from the point of view of information processing in the brain. Another incentive for these abstractions is to reduce the amount of computation required to simulate artificial neural networks, so as to allow one to experiment with larger networks and train them on larger data sets.

An Artificial Neural Network (ANN) is a highly parallel distributed network of connected processing units called neurons. It is motivated by the human cognitive process: the human brain is a highly complex, nonlinear and parallel computer. The network has a series of external inputs and outputs which take or supply information to the surrounding environment. Inter-neuron connections are called synapses which have associated synaptic weights. These weights are used to store knowledge which is acquired from the environment. Learning is achieved by adjusting these weights in accordance with a learning algorithm. It is also possible for neurons to evolve by modifying their own topology, which is motivated by the fact that neurons in the human brain can die and new synapses can grow.

![Figure 1. Mathematical model of non linear neuron](image)

Mathematically we describe the neuron by the following equations
WHERE $\Phi(V)$ is activation function

Activation Function. The activation function, denoted $\phi(v)$, defines the output of the neuron in terms of the local field $v$. Three basic types of activation functions are as follows:

ARCHITECTURES OF NEURAL NETWORKS

2.1 Single-Layer Feed-Forward Networks

The simplest form of a layered network, consisting of an input layer of source nodes that project onto an output layer of neurons. The network is strictly feed forward, no cycles of the information are allowed. Figure 2 shows an example of this type of network. The designation of single-layer refers to the output layer of neurons, the input layer is not counted since no computation is performed there.

2.2 Multi-layer feed forward

This class of feed-forward neural networks contains one or more hidden layers, whose computation nodes are correspondingly called hidden neurons. The hidden neurons intervene between the input and output layers, enabling the network to extract higher order statistics. Typically the neurons in each layer of the network have as their inputs the output signals of the neurons in the preceding layer only. Figure 3 shows an example with one hidden layer. It is referred to as a 3-3-2 network for simplicity, since it has 3 source nodes, 3 hidden neurons (in the first hidden layer) and 2 output neurons. This network is said to be fully connected since every node in a particular layer is forward connected to every node in the subsequent layer.

2.3 WAVELET NEURAL NETWORK

The structure of a wavelet neural network is very similar to that of a (1+ 1/2) layer neural network. That is, a feed-forward neural network, taking one or more inputs, with one hidden layer and whose output layer consists of one or more linear combiners or summers (see Figure 4). The hidden layer consists of neurons, whose activation functions are drawn from a wavelet basis. These wavelet neurons are usually referred to as wavelons.
There are two main approaches to creating wavelet neural networks.

- In the first the wavelet and the neural network processing are performed separately. The input signal is first decomposed using some wavelet basis by the neurons in the hidden layer. The wavelet coefficients are then output to one or more summers whose input weights are modified in accordance with some learning algorithm.
- The second type combines the two theories. In this case the translation and dilation of the wavelets along with the summer weights are modified in accordance with some learning algorithm.

### 2.3.1 One-Dimensional Wavelet Neural Network

The simplest form of wavelet neural network is one with a single input and a single output. The hidden layer of neurons consists of wavelons, whose input parameters (possibly fixed) include the wavelet dilation and translation coefficients. These wavelons produce a non-zero output when the input lies within a small area of the input domain. The output of a wavelet neural network is a linear weighted combination of the wavelet activation functions. Figure 5. shows the form of a wavelon. The output is defined as:

\[
\psi_{\lambda,t}(u) = \psi \left( \frac{u - t}{\lambda} \right)
\]

where \( \lambda \) and \( t \) are the dilation and translation parameters respectively.

### 2.3.2 Multidimensional Wavelet Neural Network

The input in this case is a multidimensional vector and the wavelons consist of multidimensional wavelet activation functions. They will produce a non-zero output when the input vector lies within a small area of the multidimensional input space. The output of the wavelet neural network is one or more linear combinations of these multidimensional wavelets. Figure 6. shows the form of a wavelon. The output is defined as:

\[
\Psi(u_1, \ldots, u_N) = \prod_{n=1}^{N} \psi_{\lambda_n,t_n}(u_n)
\]

This wavelon is in effect equivalent to a multidimensional wavelet.

### Figure 4. Wavelet Neural Network

### Figure 5. One-Dimensional Wavelet Neural Network

### Figure 6. Wavelet Neuron with a Multidimensional Wavelet Activation Function
nodes in the hidden layer, is given in Fig. 1. The output signal of network is calculated as:

\[
\psi_{a_i, t_i}(x) = \alpha_i^{-d/2} \psi \left( \frac{x - L_i}{\alpha_i} \right)
\]

where \(x = (x_1, x_2, \ldots, x_q)\) is the vector of inputs and \(_i \), \(i = 1, 2, \ldots, k\) are weight coefficients between hidden and output layers, and \(_ai, t I\) are dilated and translated versions of a mother wavelet function \(_i q \rightarrow \_i\):

\[
\psi_{a_i, t_i}(x) = \alpha_i^{-d/2} \psi \left( \frac{x - L_i}{\alpha_i} \right)
\]

The mother wavelet \(\psi\) is a waveform that has limited duration and zero mean value. Also \(|\psi(x)|\) and \(|\psi(\omega)|\) rapidly decay to zero as \(\frac{\omega}{\pi} \rightarrow \infty\) and \(\frac{\omega}{\pi} \rightarrow \infty\).

In Eq. (2), the dilation or scaling parameter \(a_i \in \mathbb{R}^+\) controls the support of the wavelet and the translation parameter \(t_i \in \mathbb{R}^q\) determines its central position. In Eq. (1), pairs \((a_i, t_i)\) are taken from a grid \(A\) given by

\[A = \{(\alpha^n,m^n) : n \in \mathbb{Z}, m \in \mathbb{Z}^q\}\]

where the scalar parameters \(\alpha\) and \(\beta\) define the step sizes of dilation and translation discretizations, respectively (typically \(\alpha = 2\), \(\beta = 1\)). According to above definitions, any function \(f \in L_2(\mathbb{R}^q)\) (finite-energy and continuous or discontinuous) can be approximated by an arbitrary precision using the wavelet network given in Fig. 1.

According MRA theory, the dilation parameter of a wavelet can be interpreted as resolution parameter. Therefore, Eq. (1) illustrates that any function can be described by a linear combination of wavelets with different resolution levels. In fact, based on the multiresolution property it is possible to present a library of wavelets. The wavelets with coarse resolution capture the global (low frequency) behavior and the wavelets with fine resolution capture the local (higher frequency) behavior of the function to be approximated, using Eq. Accordingly, WNNs as function approximators have the advantages of fast convergence, easy training and high accuracy.

On the other hand, a number of methods are available in to construct multidimensional wavelets, in both single-scaling and multiscaling forms, based on one-dimensional mother wavelet functions. In the single-scale multidimensional wavelet frames, a single dilation parameter is used in all the dimensions of each wavelet, and the multidimensional wavelet frames can be built by using a single mother wavelet. To fulfill this aim, the radial functions are used that can generate single-scaling wavelet frames. In the multiscaling 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 \(\Psi : \mathbb{R} \rightarrow \mathbb{R}\), multiscaling wavelet frame \(\Psi : \mathbb{R}^q \rightarrow \mathbb{R}\) is built by setting

\[
\Psi(x) = \Psi(s(x_1) \ldots \Psi(x_q)\)
\]

for \(x = (x_1, x_2, \ldots, x_q)\).

### 2.4 STRUCTURE OF ADAPTIVE FWN CONTROLLER

In this section, the structure of the new proposed adaptive fuzzy wavelet network controller (A-FWNC) is described. Then, the adaptation parameters in the controller are introduced in Section 5, such that the adaptive FWN controller can be changed during real-time operation.

Consider the following \(q\)-th order nonlinear affine system:

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \ldots \\
\dot{x}_q &= f(x_1, x_2, \ldots, x_q) + g(x_1, \ldots, x_q)uy = x_1 \\
\end{align*}
\]

or, equivalently

\[
\begin{align*}
x(q) &= f(x, \dot{x}, \ldots, x(q-1)) + g(x, \dot{x}, \ldots, x(q-1))uy = x \\
\end{align*}
\]

where \(f\) and \(g\) are unknown real continuous functions, \(u \in \mathbb{R}\) and \(y \in \mathbb{R}\) are the control input and output of the system, respectively. The state vector of the system \(x = (x_1, x_2, \ldots, x_q)\) is assumed to be available for measurement.

Let define the tracking error \(e\) as

\[
e = yd - y = yd - x
\]

where \(yd\) is the desired output. According to (5) and (6), if the functions \(f, g\) are known, then the feedback linearization control input is

\[
u^* = 1/g(x)[-f(x) + y(q)k + cTe]
\]
By applying fuzzy inference mechanism [14] output of whole network is calculated as
\[ u_{AFWN} = \sum_{i} c_i \_i(x)u_i = \sum_{i} c_i \_i(x)k_iu_i \] (14)
where \( \_i(x) = A_{i_0}(x) = A_{i_1}(x) = A_{i_2}(x) = \ldots \)
\[ \hat{u}_i = \sum_{i} A_{i_j}(x_j)u_i, i = 1, 2, \ldots, c, \] (15)
In (15), \( k \) is the adaptation vector such that each fuzzy rule corresponds to only one adaptation parameter which needs to be adapted on-line, and \( w(x) \) is the vector of \( w_i(x) = \_i(x)u_i, i = 1, 2, \ldots, c \), so that \( \_i \) determines the degree of contribution of each sub-WNN and \( u_i \) is a linear combination of single-scaling wavelets (11).

Our next task, in this constructive route, is to construct an FWN based on the training data set and to develop an adaptive law to adjust the adaptation parameters, so that adaptive FWN controller can approximate the feedback linearization control input and the closed-loop control system is stable.

### 2.5 FWN Structure and Construction

For constructing an A-FWN controller, the first task is to construct an FWN based on the training data set. A typical fuzzy wavelet network for approximating the control input can be described by a set of fuzzy rules [10]:

\[ R_i : IF x_1 \ is \ A_{i_1}, \ AND \ x_2 \ is \ A_{i_2}, \ AND \ \ldots \ \ AND \ x_q \ is \ A_{i_q} \ THEN \ u_i = \sum_{i} A_{i_j}(x_j)u_i, i = 1, 2, \ldots, c, \] (12)
where \( x_j (1 \leq j \leq q) \) is the \( j \) th state variable of the system; and \( u_i \) is the output (control signal) 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.

The structure of fuzzy wavelet network for approximating control input is given in Figure 7 and the structure of each sub-WNN is given in Fig. 1. According to (12) and (13), the output of the network in Fig. 2 is calculated as

\[ u_{AFWN} = \sum_{i} c_i \_i(x)u_i = \sum_{i} c_i \_i(x)k_iu_i \] (14)

where \( \_i(x) = A_{i_0}(x) = A_{i_1}(x) = A_{i_2}(x) = \ldots \)
\[ \hat{u}_i = \sum_{i} A_{i_j}(x_j)u_i, i = 1, 2, \ldots, c, \] (15)
where
\[ \hat{u}_i(x) = \sum_{i=1}^{c} \mu_i(x) \hat{u}_i \]

for \( i = 1, 2, \ldots, c, j = 1, 2, \ldots, q, c \) is the total number of fuzzy rules and \( q \) is the number of state variables of the system. After calculating the output of FWN, the training of network starts.

Consider the training data set of the nominal system:

\[ \{(x_1^t, u_d^t)\}, \ 1 \leq t \leq L, \ x_1^t \in \Re^q, \ u_1^t \in \Re \]

**Figure 7. Fwn Structure and Construction**

where \( L \) is the total number of the training patterns, \( Xd = [xd1, xd2, \ldots, xdl] \) is the input matrix, and \( ud = [ud1, ud2, \ldots, udl] \) is the desired control signal vector. Our goal is to train the network in Fig. 2 based on data set in (18) so that the error between the output of FWN and \( ud \) is minimized.

Most work done in the wavelet networks uses simple wavelets. Fuzzy wavelet networks in [8] use Mexican Hat function \( \hat{\mu}(x) = (1 - x^2) \exp(-x^2/2) \) as wavelet function and accordingly, based on data set, dilation value \( Mi \) is chosen to be in the range from \(-5\) to \(4\). After determining the dilation parameters, according to proposed method in [8], wavelet candidates are selected. Some wavelet candidates with various translation parameters, which are selected according to input training data and not according to the output data, are often redundant for constructing FWN. So OLS algorithm can be used for purifying wavelet candidates [8]. This algorithm automatically determines the network size (the number of fuzzy rules, total number of wavelets and number of wavelets in each sub-WNN) and estimates initial weights \( _Mi ,tk \) in a reasonable number of iterations. Set of wavelet candidates \( W = \{1, 2, \ldots, LW\} \) are a set of regressor vectors which construct the output of the network in Fig. 2. Since these regressors are usually correlated, the degree of the 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, so that it is possible to calculate the contribution degree of each basis vector to the output energy [6].

At the end stage of OLS algorithm, \( S = \sum_{i=1}^{c} Ti \), the total number of selected wavelets, \( Ti, i = 1, \ldots, c \), the number of selected wavelets at dilation \( (Mi) \) and \( c \), the number of fuzzy rules or sub-WNNs, are determined. Details of the FWN initialization parameters can be found in Appendix A.

Learning algorithm uses a two-stage efficient learning scheme. The EKF (extended Kalman filter) method is used for training the nonlinear parametric parameters \( pijr , t kj \) and then LS (least-squares) algorithm for updating all the weights \( _Mi ,tk \) where \( j = 1, 2, \ldots, q, r = 1, 2, 3, k = 1, 2, \ldots, S \) and \( i = 1, 2, \ldots, c \). In [20], EKF and BP training algorithms are compared and it is shown that, EKF method has the advantages of better convergence and faster training speed.

The learning procedure will be repeated according to a performance index \( Ji \) at \( i \)th iteration which is defined as

\[ Ji = \frac{\sqrt{\sum_{t=1}^{L} (\hat{u}_t - u_d^t)^2}}{\sqrt{\sum_{t=1}^{L} (u_d^t - \bar{u})^2}}, \ 1 = \frac{1}{L} \sum_{t=1}^{L} u_d^t \]

where \( u_d^t \) is the desired control input and \( \bar{u} \) is the estimated output from the FWN in Fig. 2. At the end of learning algorithm, parameter values of membership functions and sub-WNNs are found which then are used in the adaptive FWN controller.
Notice that, in THEN-parts of fuzzy rules, FWN employs $c$ sub-WNNs (linear combination of wavelets) rather than using constants or linear equations as in the traditional fuzzy models. FWN uses both globalized and localized approximation of the 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.

### 2.6 DESIGN OF ADAPTIVE FWN CONTROLLER

The main objective in this section is to derive the proper adaptation rule of adaptive FWN controller, so that we can yield the feedback linearization control input (8) in the situation of unknown functions $f$ and $g$ Figure 8. To begin, adding and subtracting $gu^*$ on the right-hand side of (5), and substituting (8) into (5), we have

$$x(q) = g(x)(u_{AFWN} - u^*) + y(q)d + cTe$$

Using definition of the tracking error, we obtain the error equation as

$$e(q) = g(x)(u^* - u_{AFWN}) - cTe$$

or equivalently, the error equation (21) can be rewritten in the vector form

$$e = Ae + b(u^* - u_{AFWN})$$

where $A_{q\times q}$ and $b_{q\times 1}$ are in the following form:

$$A = 
\begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
-c_1 & -c_2 & \cdots & \cdots & -c_q
\end{bmatrix},
\quad b = 
\begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
g
\end{bmatrix}$$

Let define the optimal parameter vector:

$$k^* = \arg \min_{k \in \mathbb{R}^q} \left[ \sup_{x \in \mathbb{R}^q} |u_{AFWN}(x|k) - u^*| \right]$$

and the minimum approximation error:

$$e = k^*T \eta_{W}(x) - u^*$$

then, from (25), we have

Using (15), (26), the error equation (22) can be rewritten as

$$\dot{e} = Ae + b(k^*T - kT)\eta_{W}(x) - Be$$

Now, the design task is to derive $k$, so that the closed-loop control system is stable and $e \to 0$ as $k \to k^*$. By considering Assumption 1 and Theorem 1 in the following, the adaptive law is chosen as

$$\dot{k} = eTPq\eta_{W}(x)$$

where $Pq$ is the last column of a symmetric positive definite matrix $P_{q\times q}$ which satisfies the Lyapunov equation, and $w(x)$ is according to (15).

![Figure 8. Design of Adaptive Fwn Controller](image)

### 3. SIMULATION RESULTS

In this section, two examples are given to demonstrate the validity of the proposed A-FWN controller. We apply A-FWNC to control a second-order nonlinear inverted pendulum system in order to compare the capabilities of the adaptive FWN controller with ANFIS controller [14] and adaptive TSFC [15]. The low dimension of proposed example does not alter the generality of the algorithm in higher dimension. In the first example, the objective is to generate an appropriate actuator force $u$ to control the motion of the cart, such that the pole can be balanced in the vertical position. In this case, we show that the number of constructed fuzzy rules represented by OLS method is enough to approximate control signal $u$, such that the closed-loop control system is stable and the transient response characteristics is proper. In second example, an appropriate actuator force $u$ is represented by the proposed controller, such that the pole can track a desired output. In each
example, we show that the proposed controller is robust to time-variant control problems. Figure 9.

![Figure 9. A-FWN controller](image)

Assume that $x_1 = \theta$ is the angle of 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. 4. The state equations can be expressed by

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= f + gu
\end{align*}
\]

Where

\[
f = g r \sin x_1 - \frac{m L x_1^2 \sin x_1 \cos x_1}{m + M}, \quad g = \frac{\cos x_1}{L} \left( \frac{4}{3} \frac{m \cos^2 x_1}{m + M} \right)
\]

For nominal model, $gr$ (acceleration due to gravity) is 9.81m/s²; $L$ (half-length of the pole) is 0.5 m; $M$ (mass of the cart) is 1.0 kg; and $m$ (mass of the pole) is 0.1 kg. For this system, according to Fig. 2, our FWN has two inputs $(x_1, x_2)$ and one output $u$. We choose $c = [1, 2]^T$ (so that $h(s) = s^2 + c^2s + c1$ is stable) and $Q = \text{diag}(10, 10)$. Then, by solving the Lyapunov equation (B.2), we obtain

\[
P = \begin{bmatrix} 15 & 5 \\ 5 & 5 \end{bmatrix}
\]

In all simulations, we integrated the closed-loop system differential equations using the MATLAB command “ode45”.

4. CONCLUSIONS AND FUTURE SCOPE

In this our project a new adaptive fuzzy wavelet network controller is developed for nonlinear affine systems. The presented controller combines fuzzy models with the theory of multiresolution analysis (MRA) of wavelet transforms. The proposed adaptive gain controller, which results from the direct adaptive approach, is applied to tune the adaptation parameter in the THEN-part of each fuzzy rule during real-time operation. Fuzzy rules are corresponding to sub-WNNs with different resolution levels such that the degree of contribution of various sub-WNNs can be controlled flexibly. Orthogonal least square method represents the number of fuzzy rules and selected wavelets. FWN is constructed based on the training data set of the nominal system and the constructed fuzzy rules can be adjusted by learning the translation parameters of the selected wavelets and also determining the shape of membership functions. Then, the constructed adaptive FWN controller is employed, such that the feedback linearization control input can be best approximated and the closed-loop stability is guaranteed. Referring to our discussion about the adaptive FWN controller, we state the objectives of the paper at these points: (i) It is possible to handle problems of large dimension with such adaptive FWN controller. The reason is that the efficient procedure of selecting wavelets used in the OLS method is not very sensitive to the input dimension. (ii) Each fuzzy rule corresponds to only one adaptation parameter that needs to be adapted on-line, no matter how many state variables are used in the modeling of the system and how many membership functions correspond to each state variable in each fuzzy rule. (iii) Wavelets with different dilation values under fuzzy rules are fully utilized to capture various behaviors (global or local) of approximated control signal. Unlike the traditional fuzzy models with only one localized approximation of function, the FWN uses both globalized and localized approximation of function. The obtained results of simulation examples demonstrate that the proposed controller is quite effective in control nonlinear systems. The results of comparisons show that the number of fuzzy rules and on-line adjustable parameters are efficiently reduced in comparison with ANFIS controller and also the presented controller significantly improves the transient response characteristics, compared with adaptive TSFC.

REFERENCES


