Adaptive fuzzy wavelet network control design for nonlinear systems

Maryam Zekri, Saeed Sadri, Farid Sheikholeslam*

Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran

Received 10 June 2007; received in revised form 25 November 2007; accepted 11 February 2008

Available online 5 March 2008

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.

© 2008 Elsevier B.V. All rights reserved.

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 [6,29,8]. 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 [39,38,4,13] and fuzzy wavelet networks [10,33,24,30].

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 [39]. Therefore, the main objectives of wavelet network can be specified as follows [39]: (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 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 [18]. 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 [38]. Most work done in the wavelet networks uses simple wavelets. The wavelets that Zhang and Benveniste [39] and Pati and Krishnaprasad [26] used in their networks are frame bases. Other wavelet networks structures use orthogonal bases [37].

Daniel et al. [10] have proposed a fuzzy wavelet network based on multiresolution analysis (MRA) of wavelet transforms and fuzzy concepts to approximate arbitrary nonlinear functions. The goal of the introduction of fuzzy model into WNN is to improve function approximation accuracy in terms of the dilation and translation parameters of wavelets, while not increasing the number of wavelet bases. In [10], the difficulties of selecting wavelets are reduced and orthogonal least-square (OLS) algorithm is used to select important wavelets. Since the efficient procedure of selecting wavelets used in the OLS method is not very sensitive to the input dimension [38], the dimension of the approximated function does not cause the bottleneck for constructing FWN. In [10], comparison with WNNs, the model accuracy and generalization capability of the FWN are improved by training the network to learn the translation parameters of the selected wavelets and adjusting the shape of fuzzy sets. Each fuzzy rule corresponding to a sub-WNN consists of single-scaling wavelets. Since the dilation parameter has explicit physical concept (i.e., resolution) the sub-WNNs with different resolution levels are used to capture different behaviours of (global or local) approximated function.

Considering these advantages, WNN and FWN can be applied to the problems of function approximation, system identification and control [12,1–3,36,32,21,22].

Feedback linearization techniques for controlling a special group of nonlinear systems have been developed in the last two decades [31,17]. However, these techniques can only be applied to certain nonlinear systems whose parameters are known accurately. If a nonlinear system contains unknown or uncertain parameters, then the feedback linearization is no longer applicable. The basic objective of adaptive fuzzy control is to maintain consistent performance of a system in the presence of uncertainty or unknown variation in plant parameters. Therefore, adaptive fuzzy strategies are suitable to solve the abovementioned problem. Wang [34] has presented a “direct adaptive fuzzy controller” based on fuzzy controller rules and an “indirect adaptive fuzzy controller” based on fuzzy modelling rules in [35]. Adaptive fuzzy control is a progressive area in control engineering and is expected to be more developed in the future [23,11,25,27].

Fuzzy controller model is based on constructing knowledge base created by experts’ knowledge. Since for some complicated process human experience is not available, construction of knowledge base is difficult. Therefore, the procedure of determining the fuzzy control rules, membership functions and structures of the fuzzy controller are crucial issues in fuzzy controller design. In adaptive-network-based fuzzy inference system (ANFIS) [15,16], Takagi–Sugeno (TS) model is the most common basis. In ANFIS, initial membership functions can be set up intuitively. Then, a set of fuzzy If–Then rules is generated by learning process to approximate a desired data set. In ANFIS model, the number of membership functions assigned to each input variable is chosen empirically, i.e., by examining the desired data set and/or by trial and error. Therefore, there are no simple ways to advance determining of the minimal number of hidden nodes, necessary to achieve a desired performance level [15]. On the other hand, the ANFIS generates a complete rule base and therefore, suffers from the curse of dimensionality [9].

This paper presents a new adaptive FWN controller to approximate the feedback linearization control input for SISO nonlinear affine systems. The method tries to get rid of the abovementioned difficulties. The proposed controller not only reserves the multiresolution capability, but also has the advantages of simple structure, high approximation accuracy and good generalization performance to nonlinear system. The network size, number of fuzzy rules, number of wavelets
in each sub-WNN and initial weights are determined in a reasonable number of iterations by OLS algorithm and the algorithm provides good initializations for the learning procedure. Translation parameters of the selected wavelets and shape of membership functions are trained by the training data set of the nominal system, such that the proposed FWN shows high approximation accuracy and fast convergence. Then, a direct adaptive law is developed based on the presented FWN, such that the adaptive FWN controller can be tuned on-line to approximate the control signal and also closed-loop system remains stable. The main features of the adaptive controller proposed in this work are:

(i) It is possible to handle problems of large dimension with such adaptive FWN controller. In general, in nonparametric estimation, the complexity of the estimators grows rapidly with input dimension which leads to curse of dimensionality. To solve this problem, some techniques are presented in [38]. According to [38], 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 modelling 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 behaviours (global or local) of approximated control signal. In defuzzification layer, FWN employs some sub-WNNs rather than using constants or linear equations as in traditional fuzzy models. Unlike the traditional fuzzy models with only one localized approximation of function, the FWN uses both globalized and localized approximation of function [10].

According to [16], ANFIS controller can be applied for feedback linearization. To achieve good performance, total adjustable parameters of the ANFIS controller should be update on-line incrementally and a sliding control should be applied to guarantee the closed-loop stability. It is evident that on-line updating of a large number of parameters takes long time and needs much more computational requirements. Since ANFIS architecture is based on the TS fuzzy model and is constructed by data set training, we compare our A-FWN controller with ANFIS controller. Also, in [5], Chen and Wong have investigated a Takagi–Sugeno fuzzy controller (TSFC). In their work, the direct adaptive fuzzy controller is employed to directly adapt the gain parameter in the THEN-part, such that the control input can be approximated. THEN-part of TSF controller is a linear combination of the state variables which is essentially a global function. The coefficients of the linear combination are coefficients of the error equation which should be properly chosen such that the closed-loop control system is stabilized. In addition, in [5], a maximum control which is based on the system’s bounds is developed to guarantee system stability in the sense of Lyapunov. On the other hand, adaptive TSFC uses a complete rule base and therefore suffers from the curse of dimensionality. Since direct adaptive TSFC is based on the TS fuzzy model, we compare our A-FWN controller with adaptive TSFC.

This paper is organized as follows: A brief background of WNN is given in Section 2. Sections 3 and 4 discuss the structure of proposed adaptive FWN controller and the construction of FWN by the training data set, respectively. Section 5 describes the procedure by which the design of the proposed direct adaptive FWN controller is fulfilled. For a few desired outputs, the simulation examples are provided to verify the performance and capability of the proposed controller in Section 6. Finally, a brief conclusion is drawn in Section 7 and we compare the performance of the proposed adaptive controller with those of ANFIS controller and adaptive TSFC. The results of comparisons show that the number of fuzzy rules and on-line adjustable parameters are efficiently reduced, in comparison with ANFIS controller. Moreover the presented controller significantly improves the transient response characteristics, compared with TSFC [5].

2. Wavelet neural network

A wavelet network corresponds to a three-layer structure, using wavelets as activation functions [28]. The structure of wavelet network with one output $y$, $q$ inputs $(x_1, x_2, \ldots, x_q)$ and $k$ nodes in the hidden layer, is given in Fig. 1.

The output signal of network is calculated as

$$y = \sum_{i=1}^{k} o_i \psi_{d_i,t_j}(x)$$

(1)
where $\mathbf{x} = (x_1, x_2, \ldots, x_q)^T$ is the vector of inputs and $\omega_i$, $i = 1, 2, \ldots, k$ are weight coefficients between hidden and output layers, and $\psi_{a_i, t_i}$ are dilated and translated versions of a mother wavelet function $\psi : \mathbb{R}^q \to \mathbb{R}$:

$$
\psi_{a_i, t_i}(\mathbf{x}) = a_i^{-q/2} \psi \left( \frac{\mathbf{x} - t_i}{a_i} \right)
$$

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

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

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

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) behaviour and the wavelets with fine resolution capture the local (higher frequency) behaviour of the function to be approximated, using Eq. (1). 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 [19] 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 $\psi_s : \mathbb{R} \to \mathbb{R}$, multiscaling wavelet frame $\psi : \mathbb{R}^q \to \mathbb{R}$ is built by setting

$$
\psi(x) = \psi_s(x_1) \ldots \psi_s(x_q) \quad \text{for } x = (x_1, x_2, \ldots, x_q)^T
$$

3. 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:

\[
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\vdots \\
\dot{x}_q &= f(x_1, \ldots, x_q) + g(x_1, \ldots, x_q)u \\
y &= x_1
\end{aligned}
\]  

(5)

or, equivalently

\[
\begin{aligned}
x^{(q)} &= f(x, \dot{x}, \ldots, x^{(q-1)}) + g(x, \dot{x}, \ldots, x^{(q-1)})u \\
y &= x
\end{aligned}
\]  

(6)

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 $x = (x_1, x_2, \ldots, x_q)^T = (x, \dot{x}, \ldots, x^{(q-1)})^T \in \mathbb{R}^q$ is assumed to be available for measurement.

Let define the tracking error $e$ as

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

(7)

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

\[
u^* = \frac{1}{g(x)} [-f(x) + y^{(q)}_d + \epsilon^T e]
\]  

(8)

where $\epsilon = (c_1, c_2, \ldots, c_q)^T$ is a positive constant vector and $\epsilon = (e, \dot{e}, \ldots, e^{(q-1)})^T$ is the error vector. Substituting (8) into (5), we can easily obtain error equation as

\[
e^{(q)} + c_1 e + \cdots + c_q e^{(q-1)} = 0
\]  

(9)

The elements in vector $c$ can be chosen appropriately, such that all roots of the polynomial $h(s) = s^q + c_q s^{q-1} + \cdots + c_1$ are in the open left half-plane. Thus the controlled system is stabilized if the control input (8) can be implemented.

In practice, both $f$ and $g$ are unknown functions and so it is a hard task to implement (8). To solve the problem, we propose an A-FWNC to approximate the feedback linearization control input.

Consider the following adaptive FWN controller structure can be described by a set of fuzzy rules:

\[
\text{IF } x_1 \text{ is } A^i_1, \text{ AND } x_2 \text{ is } A^i_2, \text{ AND } \ldots, x_q \text{ is } A^i_q \text{ THEN } u_{fi} = k_i u_{wi}
\]  

(10)

Suppose that $u_{wi}$ of the THEN-part is defined as

\[
u_{wi} = \sum_{k=1}^{T_i} \omega_{M_i, \ell^k} \psi_{M_i, \ell^k}(x)
\]  

(11)

Assuming $x_j (1 \leq j \leq q)$ is the $j$th state variable of the system; $R^i$ is the $i$th rule $(1 \leq i \leq c)$; $c$ is the total number of fuzzy rules; $T_i$ is the total number of wavelets for the $i$th rule; $k_i$ is a adaptation parameter to be adapted for $i$th rule and $u_{fi}$ is the output of the local model for rule $R^i$. $\ell^k \in [\ell^k_1, \ell^k_2, \ldots, \ell^k_q]$, where $\ell^k_j \in \mathbb{R}$ is the translation value for corresponding wavelet $k$. All the membership functions are Gaussian-type function defined as

\[
A^i_j(x_j) = \exp[-(((x_j - p^i_j_1)/p^i_j_2)^2)_{p^i_j_3/2}]
\]  

(12)

which contains three fitting parameters $p^i_j_1, p^i_j_2, p^i_j_3 \in \mathbb{R}$ and $0 < p^i_j_3 \leq 5$. Each of these parameters has a physical meaning: $p^i_j_1$ determines the centre of the corresponding membership function; $p^i_j_2$ is the half-width and $p^i_j_3$ controls the shape of membership function.
In each fuzzy rule or sub-WNN, wavelets $\psi_{M_i,j}^{(k)}(x)$ are expressed by tensor product of 1-D wavelet functions:
\[
\psi_{M_i,j}^{(k)}(x) = 2^{M_i/2}\psi_{M_i,j}^{(k)}(2^{M_i} x - t_j^k)
\]
\[
= \prod_{j=1}^{q} 2^{M_i/2}\psi_{M_i,j}^{(k)}(2^{M_i} x_j - t_j^k)
\]  

(13)

According to (13), in each fuzzy rule or sub-WNN the wavelets are single-scaling and there is the same dilation index in all the dimensions.

By applying fuzzy inference mechanism [14] output of whole network is calculated as
\[
u_{AFWN} = \sum_{i=1}^{c} \hat{\mu}_i(x)u_{fi} = \sum_{i=1}^{c} \hat{\mu}_i(x)k_i u_{wi}
\]

(14)

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

Eq. (14) can be written as
\[
u_{AFWN}(x|k) = k^T \eta_{ui}(x)
\]

(15)

where
\[
k = (k_1, k_2, \ldots, k_c)^T, \quad \eta_{ui}(x) = (\eta_{u1}(x), \eta_{u2}(x), \ldots, \eta_{uc}(x))^T, \quad \eta_{ui}(x) = \hat{\mu}_i(x)u_{wi}, \quad i = 1, 2, \ldots, c
\]

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 $\eta_{ui}(x)$ is the vector of $\eta_{ui}(x) = \hat{\mu}_i(x)u_{wi}, i = 1, 2, \ldots, c$, so that $\hat{\mu}_i$ determines the degree of contribution of each sub-WNN and $u_{wi}$ is a linear combinations 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.

4. FWN structure and construction

For constructing 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]:
\[
R^i: IF x_1 IS A_1^i, AND x_2 IS A_2^i, AND \ldots x_q IS A_q^i THEN \hat{u}_i = \sum_{k=1}^{T_i} \omega_M \psi_{M_i,j}^{(k)}(x)
\]

\[
M_i \in \mathbb{Z}, \quad l_j^k \in \mathbb{R}^q, \quad \omega_{M_i,j}^k \in \mathbb{R} \quad \text{and} \quad x \in \mathbb{R}^q
\]

(16)

where $x_j (1 \leq j \leq q)$ is the $j$th state variable of the system; and $\hat{u}_i$ 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 Fig. 2 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
\[
\hat{u} = \sum_{i=1}^{c} \hat{\mu}_i(x)\hat{u}_i
\]

(17)

where $\hat{\mu}_i(x) = \mu_i(x)/\sum_{i=1}^{c} \mu_i(x)$ and $\mu_i(x) = \prod_{j=1}^{q} A_j^i(x_j)$ 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:
\[
\{(\underline{\Delta}_l^d, u_l^d)\}, \quad 1 \leq l \leq L, \quad \underline{\Delta}_l^d \in \mathbb{R}^q, \quad u_l^d \in \mathbb{R}
\]

(18)
where $L$ is the total number of the training patterns, $X^d = [x^d_1, x^d_2, \ldots, x^d_L]^T \in \mathbb{R}^{L \times q}$ is the input matrix, and $u^d = [u^d_1, u^d_2, \ldots, u^d_L]^T$ 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 $u^d$ is minimized.

Most work done in the wavelet networks uses simple wavelets. Fuzzy wavelet networks in [10,36] use Mexican Hat wavelet and B-spline wavelets, respectively. We have selected Mexican Hat function as wavelet function and accordingly, based on data set, dilation value $M_i$ is chosen to be in the range from $-5$ to $4$.

After determining the dilation parameters, according to proposed method in [10], 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 [10]. 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 $\omega_{M_i,t_k}^{l_j}$ in a reasonable number of iterations [38]. Set of wavelet candidates $W = \{\psi_1, \psi_2, \ldots, \psi_{LW}\}$ 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 [7].

At the end stage of OLS algorithm, $S = \sum_{i=1}^c T_i$, the total number of selected wavelets, $T_i$, $i = 1, \ldots, c$, the number of selected wavelets at dilation ($M_i$) 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 $p_{jr}^l$, $t_k^j$ and then LS (least-squares) algorithm for updating all the weights $\omega_{M_i,t_k}^{l_j}$, 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 $J_i$ at $i$th iteration which is defined as

$$J_i = \frac{\sum_{l=1}^L (\tilde{u}_l - u^d_l)^2}{\sum_{l=1}^L (u^d_l - \bar{u})^2}, \quad \bar{u} = \frac{1}{L} \sum_{l=1}^L u^d_l$$

(19)
where $u^d_l$ is the desired control input and $\hat{u}_l$ 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 [10].

5. 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 $g$. To begin, adding and subtracting $gu^*$ on the right-hand side of (5), and substituting (8) into (5), we have

$$x(q) = g(x)(u_{AFWN} - u^*) + y_d(q) + \xi^T \xi$$  \hspace{1cm} (20)

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

$$e(q) = g(x)(u^* - u_{AFWN}) - \xi^T \xi$$  \hspace{1cm} (21)

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

$$\dot{e} = A e + b(u^* - u_{AFWN})$$  \hspace{1cm} (22)

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

$$A = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ -c_1 & -c_2 & \cdots & \cdots & \cdots & -c_q \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ g \end{bmatrix}$$  \hspace{1cm} (23)

Let define the optimal parameter vector:

$$k^* = \arg \min_{k \in \mathbb{R}} \left[ \sup_{x \in \mathbb{R}^q} |u_{AFWN}(x|k) - u^*| \right]$$  \hspace{1cm} (24)

and the minimum approximation error:

$$e = k^{*T} \eta_w(x) - u^*$$  \hspace{1cm} (25)

then, from (25), we have

$$u^* = k^{*T} \eta_w(x) - e$$  \hspace{1cm} (26)

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

$$\dot{e} = A e + b(k^{*T} - k^T) \eta_w(x) - be$$  \hspace{1cm} (27)

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

$$\dot{k} = \gamma e^T Pq \eta_w(x)$$  \hspace{1cm} (28)

where $\gamma$, adaptation gain, is a positive constant, $P_q$ is the last column of a symmetric positive definite matrix $P_{q \times q}$ which satisfies the Lyapunov equation, and $\eta_w(x)$ is according to (15).
Assumption 1. The approximation error $\varepsilon$ is bounded, i.e., $\varepsilon $ satisfies $|\varepsilon| \leq M_\varepsilon$.

According to Refs. [39,37], the wavelet network has universal and $L^2$ approximation properties. In other words, any function $f \in L^2(\mathbb{R}^q)$ (finite-energy and continuous or discontinuous) can be approximated to an arbitrary precision by the wavelet network. On the other hand, the wavelet networks are optimal approximators, since they require the smallest number of bits for an arbitrary precision [18]. Also the OLS method uses the modified Gram–Schmidt method and ERR (error reduction ratio) to select the important wavelets which have significant contribution to the constructed function [38,7]. Therefore, the number of fuzzy rules and the number of the selected wavelets in each sub-WNN represented by OLS method are adequate to describe the system dynamics and it is reasonable to assume that the approximation error $\varepsilon$ (25) is bounded and accordingly the limit $M_\varepsilon$ can be fulfilled.

Theorem 1. Consider the closed-loop system of Fig. 3 with the nonlinear system (5) or (6). The adaptive FWN controller is chosen as (10)–(15) and the parameter vector adaptive law is (28). If Assumption 1 is satisfied, then the proposed adaptive FWN controller can guarantee the following properties:

1. The closed-loop control system is stable.
2. $e \to 0$ as $k \to k^*$.

The proof of this theorem is given in Appendix B.

All these results are reflected in our simulations. The block diagram of the adaptive FWN controller is given in Fig. 3.

6. 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 [16] and adaptive TSFC [5]. 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.
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

$$f = \frac{g_r \sin x_1 - \frac{mLx_1^2 \sin x_1 \cos x_1}{m + M}}{L \left( \frac{4}{3} - \frac{m \cos^2 x_1}{m + M} \right)}$$

$$g = \frac{\cos x_1}{L \left( \frac{4}{3} - \frac{m \cos^2 x_1}{m + M} \right)}$$

For nominal model, $g_r$ (acceleration due to gravity) is $9.81 \text{ m/s}^2$; $L$ (half-length of the pole) is $0.5 \text{ m}$; $M$ (mass of the cart) is $1.0 \text{ kg}$; and $m$ (mass of the pole) is $0.1 \text{ kg}$. For this system, according to Fig. 2, our FWN has two inputs $(x_1, x_2)$ and one output $u$. We choose $\zeta = [1, 2]^T$ (so that $h(s) = s^2 + c_2s + c_1$ 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.”

**Example A.** Balancing of the pole in the vertical position.

In this example, we sampled 240 pairs $\{x^d, u^d\}$ of the nominal model as training data set where $x_0 = [\pi/4; 0], t = [0, 12]$ and $y_d = 0$. By using OLS algorithm [38], five fuzzy rules or sub-WNN are represented to approximate the control input for balancing of the pole in vertical position. The A-FWN controller used here contains five fuzzy rules, with five membership functions being assigned to each input state variable and five adaptation parameters that tune the parameters of the controller in the real-time operation system. According to (10) each fuzzy rule has the following form:

$$IF \ x_1 \ is \ A^i_1, \ AND \ x_2 \ is \ A^i_2 \ THEN \ u_{fi} = k_i \sum_{k=1}^{T_i} \omega_{M_i,t_i} \psi_{M_i,t_i}^{(k)}(x), \ \ i = 1, 2, 3, 4, 5$$

The number of the selected wavelets, $T_i$, and dilation value, $M_i$, presented by OLS method, in each fuzzy rule are given in Table 1. The output of each sub-WNN in Fig. 2 is depicted in Fig. 5.

To show that five fuzzy rules are enough for constructing of the FWN, we simulate three cases. In case A.1, we consider sub-WNN$_1$ and sub-WNN$_5$, which have the most contribution in construction of control signal, and construct...
Table 1

<table>
<thead>
<tr>
<th>$M_i$ (dilation parameter value)</th>
<th>1</th>
<th>0</th>
<th>$-1$</th>
<th>$-2$</th>
<th>$-3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_i$ (number of wavelets in each sub-WNN)</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>7</td>
</tr>
</tbody>
</table>

Fig. 5. Output of each sub-WNN.

the FWN with two fuzzy rules. Then, in the case A.2, sub-WNNs $1, 2, 5$ are used to construct the FWN. Finally, in case A.3, we construct the FWN with all of the wavelets represented by OLS algorithm. In each case, the membership functions of the IF-part which are obtained at the end stage of learning algorithm are depicted in Fig. 6 and the parameters in case A.3 after learning step are given in Table 2.

In each case, the simulations include two stages: (a) the initial conditions of state variables $x_0$ are changed and then the adaptation law is applied to tune the rules which can balance the inverted pendulum, (b) the mass of the cart and the length of the pole in the actual system are time-variant, for instance $M = 1 + 0.5 \cdot \text{rand}(|\Delta t|)$ and $L = 0.5 + 0.3 \sin(t)$, where $\text{rand}(|\Delta t|) \in [0, 1]$ and random value updates each $\Delta t = 0.05$ s. Under these conditions, our goal is to show the robustness of the proposed controller to time-variant control problems. Additionally, we examine the variation of adaptation gain $\gamma$ in each case. The simulation results of the three cases are depicted in Figs. 7–12. From the simulated results it can be seen that

(i) In case A.1, by using two fuzzy rules, the output response decays quickly to $y_d = 0$, but oscillates around it with a small amplitude. In cases A.2 and A.3, the proposed adaptive FWN controller is able to balance the inverted pendulum in the vertical position for various initial conditions $x_0$. Comparison of case A.2 and A.3 shows that the amplitude of control input is reduced when the FWN is constructed by five fuzzy rules. Comparison of Fig. 8 and Fig. 13a, shows that in case A.2, by using less fuzzy rules, the output response decays faster than that with using adaptive TSFC [34] without maximum control. In [5], with a maximum control, the output response characteristics are similar to our case A.2. In case A.3, Figs. 10–12 illustrate that the controller provides a faster
(ii) In whole of the cases, the proposed adaptive controller is robust to the time-variant mass of the cart and length of the pole. In case B.2, the complexity of the ANFIS controller, compared to the controller with four rules in the case B.1 and with four and nine rules in case B.2, but obviously they are too simple to describe the control signal in two cases. In case B.2, the number of membership functions assigned to each input of the ANFIS was set to two, so the number of rules is four. Fig. 14 shows the results after applying off-line training step.

Example B. Tracking a desired output.

In this example, we apply the A-FWN controller, and the ANFIS controller to regulate the inverted pendulum system for tracking a desired output. In this simulation, 300 pairs \( \{x, u\} \) of nominal system as training data set are used where \( x_0 = [\pi/6; 0.1], t = [0, 15] \). For constructing the FWN, five fuzzy rules and 12 selected wavelets represented by OLS algorithm [38] generate the feedback linearization control signal for tracking desired output \( y_d(t) = 0.5 \sin(t) \) (case B.1). In case B.2, for tracking the desired output \( y_d(t) = 0.2(\sin t + \sin 2t) \), by using OLS algorithm [38], the feedback linearization control signal is approximated by eight rules and the number of the selected wavelets is 28. The simulation results of the two cases are depicted in Figs. 15–17. Also Fig. 18 depicts the closed-loop trajectories for cases B.1 and B.2 for initial condition \( x_0 = [\pi/6; 0.1] \) and time period \( t_0 = 0 \) to \( t_f = 20 \) s.

From the simulated results, it can be seen that (i) when the initial conditions of state variables \( x_0 \) are changed, the proposed adaptation law tunes the controller and good tracking performance is achieved, (ii) the proposed adaptive controller is robust to the time-variant mass and length of the pole, in which \( m = 0.1 + 0.05 \sin(30t) \) and \( L = 0.5 + 0.3 \sin(30t) \).

Table 4 shows the performance comparison of the proposed controller with the controller based on ANFIS model. The ANFIS controllers used here for cases B.1 and B.2 contain nine and 16 rules, respectively. (We also tried ANFIS controller with four rules in the case B.1 and with four and nine rules in case B.2, but obviously they are too simple to describe the control signal in two cases.) In case B.2, the complexity of the ANFIS controller, compared to the A-FWN controller, can be seen from Table 4. When complexity of the approximated signal is increased, the simulation results show that the structure of ANFIS controller begins to become more and more complex. The reason is that in ANFIS controller, firstly, THEN-part of TSF model is a linear combination of the state variables which is essentially a global function and secondly, number of statements in this linear combination is the same for all fuzzy rules.
note that the A-FWN controller, by using less fuzzy rules, utilizes the sub-WNNs with different resolutions levels and various number of wavelets for each sub-WNN, captures different behaviours (global or local) of approximated control signal.
Output Responses with Several Different Initial Conditions

Time-Invariant and Time-Variant Mass of the Cart and Length of the Pole

Parameter Trajectories of $K$

Control Signal by A-FWNC

Fig. 8. Case A.2: (three fuzzy rules) $k = [1, 1, 1]^T, \gamma = 600.$

Figs. 19 and 20 depict the output of the inverted pendulum system after applying off-line ANFIS controller for cases B.1 and B.2. In each case, the ANFIS controller cannot regulate the system to track a desired output and the output of the system is unstable for various initial conditions and also is not robust to time-variant
Output Responses with Several Different Initial Conditions

Time-Invariant and Time-Variant Mass of the Cart and Length of the Pole

Parameter Trajectories of K

Control Signal by A-FWNC

Fig. 9. Case A.2: (three fuzzy rules) $\delta = [1, 1, 1]^T$, $\gamma = 2000$. 
Output Responses of Inverted Pendulum

Parameter Trajectories of $k$

Control Force Trajectories of proposed Controller with Time-Invariant Time-Variant Mass of the Cart and Length of the Pole

Fig. 10. Case A.3: (five fuzzy rules) $\hat{k} = [1, 1, 1, 1]^T$, $\gamma = 50$. 
Fig. 11. Case A.3: (five fuzzy rules) $k = [1, 1, 1, 1]^T$, $\gamma = 200$. 

Output Responses of Inverted Pendulum

Parameter Trajectories of $k$

Control Force Trajectories of Proposed Controller with Time-Invariant Time-Variant Mass of the Cart and Length of the Pole
parameters of the system. According to [16], we should update 39 adjustable parameters for case B.1 and 64 adjustable parameters of the ANFIS controller on-line to regulate the inverted pendulum system and apply a sliding control to guarantee the closed-loop stability. It is evident that on-line updating a large number of parameters takes much more time and needs much computational requirements. For cases B.1 and B.2, on the other hand, five and eight adaptation parameters are used to tune the A-FWN controller during real-time operation system, respectively.

7. Conclusions

In this paper, 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 modelling 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 behaviours (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.
Output Responses with Several Different Initial Conditions

![Graph showing output responses with different initial conditions](image)

Output Response with Time-variant and Time-invariant Mass of the Cart and Length of the Pole

![Graph showing output response with time-variant and time-invariant mass](image)

Fig. 14. The performance of ANFIS controller for Case A with four fuzzy rules.

Table 3

<table>
<thead>
<tr>
<th></th>
<th>Number of rules</th>
<th>Number of nonlinear parameters</th>
<th>Number of linear parameters</th>
<th>Training cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case A.1</td>
<td>2</td>
<td>25</td>
<td>13</td>
<td>60</td>
</tr>
<tr>
<td>Case A.2</td>
<td>3</td>
<td>36</td>
<td>18</td>
<td>60</td>
</tr>
<tr>
<td>Case A.3</td>
<td>5</td>
<td>64</td>
<td>34</td>
<td>260</td>
</tr>
<tr>
<td>ANFIS</td>
<td>4</td>
<td>8</td>
<td>12</td>
<td>100</td>
</tr>
</tbody>
</table>

Appendix A. FWN Initialization

According to Section 4, the free parameters to be trained in the FWN are $p_{jr}^i$, $t_j^k$, $\omega_{Mrk}t_k^i$, where $r = 1, 2, 3$, $j = 1, 2, \ldots, q$, $i = 1, 2, \ldots, c$ and $k = 1, 2, \ldots, S$, respectively. At the end stage of OLS algorithm $t_j^k$ and initial
Fig. 15. Case B.1: (five fuzzy rules) $k = [1, 1, 1, 1]^T$, $\gamma = 1e6$.

weights $\omega_{M_i,t}^k$ are obtained. Also, all $p_{ji}^j$ which are free parameters of the membership functions are initialized as follows. These membership functions determine the degree of contribution $\hat{\mu}_i$ of each sub-WNN with a special resolution value. According to (16) and the training data set (18), the approximation degree of contribution $\bar{\mu}_{it}$ can be
Output Responses of Inverted Pendulum

\[ m = 0.1, \ L = 0.5 \]
\[ m = 0.1 + 0.05 \sin(30 \cdot t), \ L = 0.5 + 0.3 \sin(30 \cdot t); \]
\[ Y_d = 0.2 \left[ \sin(t) + \sin(2t) \right] \]

Parameter Trajectories of \( K \)

Control Force Trajectories of Proposed Controller
with Time-variant Time-invariant Mass and Length of the Pole

\[ \kappa = [1, 1, 1, 1, 1, 1, 1, 1]^T, \ \gamma = 5e4. \]
Fig. 17. Output responses with several different initial conditions. Case B.1: (five fuzzy rules) $\mathbf{K} = [1, 1, 1, 1, 1]^T$, $\gamma = 1e6$. Case B.2: (eight fuzzy rules) $\mathbf{K} = [1, 1, 1, 1, 1, 1, 1]^T$, $\gamma = 5e4$.

Fig. 18. Closed-loop system trajectories $(x_1(t), x_2(t))$ using A-FWNC for Cases B.1, B.2.
Table 4
Comparison of the A-FWNC with the ANFIS controller in training step

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of rules</th>
<th>Number of nonlinear parameters</th>
<th>Number of linear parameters</th>
<th>Training cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case B.1</td>
<td>AFWN 5</td>
<td>42</td>
<td>12</td>
<td>351</td>
</tr>
<tr>
<td>Case B.1</td>
<td>ANFIS 9</td>
<td>12</td>
<td>27</td>
<td>100</td>
</tr>
<tr>
<td>Case B.2</td>
<td>AFWN 8</td>
<td>76</td>
<td>28</td>
<td>198</td>
</tr>
<tr>
<td>Case B.2</td>
<td>ANFIS 16</td>
<td>16</td>
<td>48</td>
<td>100</td>
</tr>
</tbody>
</table>

![Graph showing output responses with several different initial conditions](image1)

![Graph showing output response with time-variant mass and length of the pole](image2)

Fig. 19. The performance of ANFIS controller for Case B.1 with nine fuzzy rules.

computed as

\[ \bar{u}_{il} = \frac{|\hat{u}_{il}(x^d_l)|}{\sum_{i=1}^{c} |\hat{u}_{il}(x^d_l)|}, \quad i = 1, 2, \ldots, c, \quad l = 1, 2, \ldots, L \]  

(A.1)
Then, the initial parameters $p_{ji}^j$ can be set according to formulas [10]

$$p_{j1}^j = \frac{\sum_{l=1}^{L} \bar{P}_{il} \bar{X}_{il}^d}{\sum_{l=1}^{L} \bar{P}_{il}}, \quad p_{j2}^j = \frac{2 \sum_{l=1}^{L} \bar{P}_{il} (x_{il}^d - p_{j1}^j)^2}{\sum_{l=1}^{L} \bar{P}_{il}}, \quad 0 < p_{j3}^j \leq 5
$$

$$i = 1, 2, \ldots, c, \quad l = 1, 2, \ldots, L$$

(A.2)

Appendix B. Proof of the theorem

Proof. Consider the Lyapunov function candidate as

$$V = \frac{1}{2} e^T P e + \frac{g}{2 \gamma} \phi^T \phi$$

(B.1)
where $\gamma$, adaptation gain, is a positive constant, $\phi = k^* - k$, and $P_{q \times q}$ is a positive definite matrix satisfying the following Lyapunov equation:

$$A^TP + PA = -Q$$ (B.2)

where $Q_{q \times q}$ is a positive definite matrix. According to the Lyapunov theorem, if $\dot{V} < 0$, then $V$ will decrease gradually. Therefore our control goal of $e = 0$ and $\tilde{k} = k^*$ will be achieved. The time derivative of $V$ is

$$\dot{V} = \frac{1}{2} \dot{e}^T P e + \frac{1}{2} \dot{e}^T P \dot{e} + \frac{g}{2 \gamma} \dot{\phi}^T \dot{\phi}$$

$$\dot{V} = -\frac{1}{2} \dot{e}^T Q e - \dot{e}^T P \dot{e} + \frac{g}{2 \gamma} \dot{\phi}^T \dot{\phi}$$ (B.3)

Using $\dot{\phi} = k^* - \tilde{k}$ and the form of $b_{q \times 1}$, the third term at the right-hand side of (B.3) becomes

$$\dot{V} = -\frac{1}{2} \dot{e}^T Q e - \dot{e}^T P \dot{e} + \frac{g}{2 \gamma} \dot{\phi}^T (\gamma \dot{e}^T P q \eta_w(x) + \dot{\phi})$$ (B.4)

where $P_q$ is the last column of $P$. Then (A.3) can be written as

$$\dot{V} = -\frac{1}{2} \dot{e}^T Q e - \dot{e}^T P \dot{e}$$ (B.5)

If the adaptation law is selected as

$$\dot{k} = -\dot{\phi} = \gamma e^T P q \eta_w(x)$$ (B.6)

then (B.3) can be formulated as

$$\dot{V} = -\frac{1}{2} \dot{e}^T Q e - \dot{e}^T P \dot{e}$$ (B.7)

According to Assumption 1, the approximation error $e$ is bounded and the limit $M_e$ can be fulfilled, such that $|e^T P b| \leq M_e |e^T P b| < \frac{1}{2} \dot{e}^T Q e$. In other words, $\dot{V}$ is negative and the closed-loop control system is stabilized and $e \to 0$ as $\tilde{k} \to k^*$. The proof of the theorem is thus established.

References


