PRAMANA — journal of physics
© Indian Academy of Sciences
Vol. 76, No. 6
June 2011
pp. 887–900

Robust adaptive fuzzy neural tracking control for a class of unknown chaotic systems

ABDURAHMAN KADIR[1,2,*], XING-YUAN WANG[1] and YU-ZHANG ZHAO[2]

[1]Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China
[2]School of Computer Science and Engineering, Xinjiang University of Finance and Economics, Urumchi 830012, China
*Corresponding author. E-mail: abdurahman.kadir@yahoo.com.cn

MS received 28 July 2010; revised 29 October 2010; accepted 9 November 2010

Abstract. In this paper, an adaptive fuzzy neural controller (AFNC) for a class of unknown chaotic systems is proposed. The proposed AFNC is comprised of a fuzzy neural controller and a robust controller. The fuzzy neural controller including a fuzzy neural network identifier (FNNI) is the principal controller. The FNNI is used for online estimation of the controlled system dynamics by tuning the parameters of fuzzy neural network (FNN). The Gaussian function, a specific example of radial basis function, is adopted here as a membership function. So, the tuning parameters include the weighting factors in the consequent part and the means and variances of the Gaussian membership functions in the antecedent part of fuzzy implications. To tune the parameters online, the back-propagation (BP) algorithm is developed. The robust controller is used to guarantee the stability and to control the performance of the closed-loop adaptive system, which is achieved always. Finally, simulation results show that the AFNC can achieve favourable tracking performances.

Keywords. Chaos control; adaptive control; adaptive identifier; fuzzy neural network; back-propagation algorithm.

PACS Nos 05.45.Gg; 05.45.Xt

1. Introduction

It is well known that chaos is useful and has great potential in many real-world engineering fields such as digital data encryption and secure communications, biomedical engineering, flow dynamics and liquid mixing, power-systems protection, and so on [1,2].

Chaos control, which is an important topic in the nonlinear science, in essence, is guiding a chaotic system to reach a desired goal dynamics through various controllers. Since chaos control was first considered by Ott *et al* [3], it has been investigated extensively by many others. Many linear and nonlinear control methods have been employed to control chaos [4–14]. Simple linear feedback control methods were proposed in [5,6]. The authors proposed time delay feedback control method to control chaotic system in [7,8]. To control chaos, sliding mode method was employed in [9–12], backstepping method in [13], and adaptive control method in [14–17]. These literatures can be classified into two groups. One is that which the precise mathematical model is needed; and the other is that which the part information of system mathematical model can be obtained. With the information completely known, the controllers are often simple, but they are complex if the information is partly known. As we know, in practical systems it is difficult to obtain a precise mathematical model. So in practical applications the investigators would like to employ simple and efficient controllers. Therefore, how to design a simple controller with the little information of chaotic systems obtained, is an open problem.

The current efforts are devoted to identifying, synchronizing and controlling chaos. The difficulties encountered in handling chaotic systems have posed a real need for using some intelligent approaches. The application of neural network and fuzzy logic controllers to chaotic systems was proposed [18–23], which appears to be quite promising. Recently, FNN incorporated the advantages of fuzzy inference and neuron learning [24–28]. The FNN possesses the merits of low-level learning and computational power of the neural network (NN), hence it requires high-level human knowledge representation and the thinking of fuzzy theory.

In this paper, we are looking into the possibility of developing a new FNN system that can be used to model chaotic systems and then to control it to a desired target. Finally, simulation results are presented to demonstrate the effectiveness of the proposed control scheme.

2. Fuzzy neural network

The configuration of the FNN is shown in figure 1, which consists of fuzzy logic and neural network. The fuzzy logic system can be divided into two parts: fuzzy IF–THEN rules and a fuzzy inference engine. The fuzzy inference engine uses the fuzzy IF–THEN rules to perform a mapping for an input linguistic vector $\mathbf{X}^{\mathrm{T}} = [x_1, x_2, \ldots, x_n] \in \mathbf{R}^n$ to an output linguistic variable $y \in \mathbf{R}$. The ith fuzzy IF–THEN rules are written as

$$\begin{array}{l} R^{(i)} : \text{if } x_1 \text{ is } A_1^i \quad \text{and} \quad x_n \text{ is } A_n^i \\ \qquad \text{then } y \text{ is } B^i, \end{array} , \tag{1}$$

where $A_1^i, A_2^i, \ldots, A_n^i$ and B^i are fuzzy sets [29,30]. Let h be the number of fuzzy IF–THEN rules. By using the product inference, centre-average and singleton fuzzifier, the output of the fuzzy logic system can be expressed as

$$y(\mathbf{X}) = \frac{\sum_{i=1}^{h} \bar{y}^i \left(\prod_{j=1}^{n} \mu A_j^i(x_j) \right)}{\sum_{i=1}^{h} \left(\prod_{j=1}^{n} \mu A_j^i(x_j) \right)} = \theta^{\mathrm{T}} \xi(\mathbf{X}), \tag{2}$$

where $\mu A^i_j(x_j)$ is the value of the membership function of the fuzzy variable, h is the total number of the IF–THEN rules, $\bar{y}^i$ is the point at which $\mu B^i(\bar{y}^i) = 1$, $\theta^{\mathrm{T}} = \left[\bar{y}^1, \bar{y}^2, \ldots, \bar{y}^h\right]$ is an adjustable parameter vector and $\xi^{\mathrm{T}} = \left[\xi^1, \xi^2, \ldots, \xi^h\right]$ is a fuzzy basis vector, where ξ^i is defined as

$$\xi^i(\mathbf{X}) = \frac{\prod_{j=1}^{n} \mu A^i_j(x_j)}{\sum_{i=1}^{h}\left(\prod_{j=1}^{n} \mu A^i_j(x_j)\right)}. \tag{3}$$

When the inputs are given into the FNN, the truth value ξ^i (layer 3) of the antecedent part of the ith implication is calculated by eq. (3). Among the commonly used defuzzification strategies, the outputs (layer 4) of the fuzzy neural system are expressed as eq. (2). The fuzzy logic approximator based on NN can be established as in [31,32]. Figure 1 shows the configuration of the fuzzy neural function approximator. The approximator has four layers. At layer 1, nodes which are inputs stand for input linguistic vector $\mathbf{X}^{\mathrm{T}} = [x_1, x_2, \ldots, x_n]$. At layer 2, nodes represent the values of the membership function of total linguistic variables. Each node of layer 2 performs a membership function value. At layer 3, nodes are the values of the fuzzy basis vector ξ. Each node of layer 3 performs a

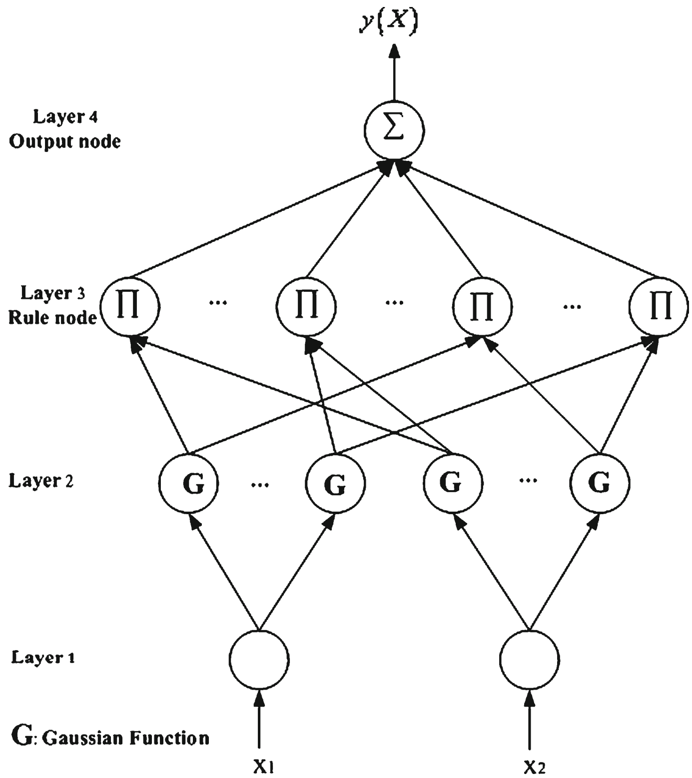

Figure 1. Schematic diagram of a fuzzy neural network.

fuzzy rule. The links between layers 3 and 4 are fully connected by the weighting factors, $\theta^{\mathrm{T}} = \left[\bar{y}^1, \bar{y}^2, \ldots, \bar{y}^h\right]$, i.e., the adjusted parameters. At layer 4, the outputs stand for the values of the output $y(\mathbf{X})$.

3. Adaptive fuzzy neural controller design

Consider the nth-order chaotic system of the controllability canonical form [33]

$$\begin{aligned}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= x_3, \\
&\vdots \\
\dot{x}_n &= f(x_1, x_2, \ldots, x_n) + \Delta f(x_1, x_2, \ldots, x_n) + d(t) + u, \\
y &= x_1,
\end{aligned} \tag{4}$$

or, equivalently

$$\begin{aligned}
x^{(n)} &= f(\mathbf{X}) + \Delta f(\mathbf{X}) + d(t) + u, \\
y &= x,
\end{aligned} \tag{5}$$

where $\Delta f(\mathbf{X})$ is the parameters uncertainty, $d(t)$ is the external noise disturbance, $f(\mathbf{X})$ are the unknown real continuous functions, $u \in \mathbf{R}$ and $y \in \mathbf{R}$ are the control input and output of the system, respectively. $\mathbf{X} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}} = [x, \dot{x}, \ldots, x^{(n-1)}]^{\mathrm{T}} \in \mathbf{R}^n$ is the state vector of the system which is assumed to be available for measurement. Based on the well-known fact that chaotic attractors are bounded in the phase space, the function $f(\mathbf{X})$ is bounded naturally. The objective of the control is to force the system output y to follow a given bounded reference signal y_m. First, define the reference vector $\mathbf{Y}_m$, and the tracking error vector $\mathbf{E}$ as follows:

$$\begin{aligned}
\mathbf{Y}_m^{\mathrm{T}} &= \left[y_m \ \dot{y}_m \ \ldots \ y_m^{(n-1)}\right], \\
\mathbf{E} &= \mathbf{Y}_m - \mathbf{X}.
\end{aligned} \tag{6}$$

The task of the tracking is to design a controller u such that

$$\lim_{t \to \infty} \|\mathbf{Y}_m(t) - \mathbf{X}(t)\| = 0, \tag{7}$$

where $\|\cdot\|$ denotes the Euclidean norm.

3.1 *Adaptive fuzzy neural controller*

In a nonadaptive control method, the controller is designed based on the plant model. If the plant model is not known, it is intuitively reasonable to replace it by an estimated model and use this model for designing the controller. This is the basic idea of adaptive controller, in which the controller is designed based on an estimated model of the plant assuming this model is the true model of the plant, and the estimated model parameters are updated by an online algorithm. Now consider the problem of controlling the system (5). In ideal circumstances (i.e., $d(t) = 0$, $\Delta f(\mathbf{X}) = 0$), if the function $f(\mathbf{X})$ is known, we can solve the control problem by the so-called feedback linearization method [34]. In this method, the function $f(\mathbf{X})$ is used to construct the following feedback control law:

$$u = u(\mathbf{X}, t) = -f(\mathbf{X}) + y_m^{(n)}(t) + \mathbf{K}^{\mathrm{T}}\mathbf{E}, \tag{8}$$

where $\mathbf{K} = [k_n \dots k_2\, k_1]^{\mathrm{T}} \in \mathbf{R}^n$ is chosen such that all roots of the polynomial $h(s) = s^n + k_1 s^{n-1} + \cdots + k_n$ are in the open left-half of the complex plane. Applying the control law (8) to the system (5) results in the following error dynamics:

$$e^{(n)} + k_1 e^{(n-1)} + \cdots + k_n e = 0. \tag{9}$$

This implies that starting from any initial conditions, we have $\lim_{t\to\infty} \|\mathbf{E}(t)\| = 0$, i.e., tracking of the reference trajectory is asymptotically achieved. As $f(\mathbf{X})$ is unknown and $d(t) \neq 0$, $\Delta f(\mathbf{X}) \neq 0$, we cannot use them for constructing the control law (8). Therefore, we define the lumped system dynamic as $F(\mathbf{X}) = f(\mathbf{X}) + \Delta f(\mathbf{X}) + d(t)$, and the system (5) can be rewritten as

$$\begin{cases} x^{(n)} = F(\mathbf{X}) + u, \\ y = x. \end{cases} \tag{10}$$

Now, we replace $F(\mathbf{X})$ by its estimates $\hat{F}(\mathbf{X})$ to construct the adaptive controller

$$u_c = u_c(\mathbf{X}, t \,|\, \theta_F) = -\hat{F}(\mathbf{X}|\,\theta_F) + y_m^{(n)}(t) + \mathbf{K}^{\mathrm{T}}\mathbf{E}, \tag{11}$$

where θ_F is the parameters of approximate systems $\hat{F}(\mathbf{X})$. Applying the control law (11) to the system (10), after some manipulation, results in the error dynamic equation

$$\dot{\mathbf{E}} = \mathbf{AE} + \mathbf{B}\big[\hat{F}(\mathbf{X}|\,\theta_F) - F(\mathbf{X})\big], \tag{12}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots 0 \\ \vdots & \ddots & \ddots & \ddots 0 \\ 0 & \cdots & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & -k_1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{13}$$

Lemma 1. *Since* $\mathbf{A}$ *is a stable matrix* (*i.e.,* $\det(s\mathbf{I} - \mathbf{A}) = h(s)$ *is stable*), *there exists a unique* $n \times n$ *positive-definite symmetric matrix* $\mathbf{P}$ *that satisfies the Lyapunov equation* [34]

$$\mathbf{A}^{\mathrm{T}}\mathbf{P} + \mathbf{PA} = -\mathbf{Q}, \tag{14}$$

where $\mathbf{Q}$ *is an arbitrary* $n \times n$ *positive-definite symmetric matrix.* □

We consider the following Lyapunov function:

$$V = V(\mathbf{E}) = \frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{PE}, \tag{15}$$

where $\mathbf{P}$ is the solution of eq. (11) with a symmetric $\mathbf{Q} > 0$. It is obvious that V is a positive-definite function and for any t, $V(t) \geq 0$. The time derivative of V along the trajectory of eq. (12), by using eq. (14) we get

$$\begin{aligned} \dot{V} &= \frac{1}{2}\dot{\mathbf{E}}^{\mathrm{T}}\mathbf{PE} + \frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{P}\dot{\mathbf{E}} \\ &= -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{QE} + \mathbf{E}^{\mathrm{T}}\mathbf{PB}\big[\hat{F}(\mathbf{X}|\,\theta_F) - F(\mathbf{X})\big]. \end{aligned} \tag{16}$$

Define the modelling error as

$$\varepsilon = \hat{F}(\mathbf{X}|\,\theta_F) - F(\mathbf{X}), \tag{17}$$

and $|\varepsilon| \leq \bar{\varepsilon}$, where $\bar{\varepsilon}$ is a positive constant.

To ensure $\dot{V} \leq 0$, a robust controller u_r is designed as

$$u_r = -\mathrm{sgn}\big(\mathbf{E}^{\mathrm{T}}\mathbf{PB}\big)\,\bar{\varepsilon}. \tag{18}$$

The synthesis controller is

$$u = u_c + u_r. \tag{19}$$

Substituting eq. (19) into system (5), after some manipulation, results in the new error dynamic equation

$$\dot{\mathbf{E}} = \mathbf{AE} + \mathbf{B}\big[\hat{F}(\mathbf{X}|\,\theta_F) - F(\mathbf{X}) + u_r\big]. \tag{20}$$

By substituting eq. (20) into eq. (16), we have

$$\begin{aligned} \dot{V} &= -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{QE} + \mathbf{E}^{\mathrm{T}}\mathbf{PB}\big[\hat{f}(\mathbf{X}|\,\theta_f) - f(\mathbf{X}) + u_r\big] \\ &= -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{QE} + \mathbf{E}^{\mathrm{T}}\mathbf{PB}\varepsilon + \mathbf{E}^{\mathrm{T}}\mathbf{PB}u_r \\ &\leq -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{QE} + \big|\mathbf{E}^{\mathrm{T}}\mathbf{PB}\big|\,\bar{\varepsilon} + \mathbf{E}^{\mathrm{T}}\mathbf{PB}u_r \\ &\leq -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{QE} \leq 0. \end{aligned} \tag{21}$$

From the above analysis we have the following theorem:

Theorem 1. *Under the controller* u, *the error dynamic* (20) *is asymptotically stable*, *i.e.* $\lim_{t\to\infty} \|\mathbf{Y}_m(t) - \mathbf{X}(t)\| = 0$ *is achieved always.* □

3.2 *Adaptive laws*

In this section, our task is to use FNN to approximate the nonlinear function $F(\mathbf{X})$, and develop an adaptive law to adjust the weighting factors of FNN, and apply BP algorithm to online tuning the means and variances of the Gaussian membership functions, such that the tracking error $\mathbf{E}$ converges to zero.

Now consider the following optimal parameter vectors:

$$\theta_F^* = \arg\min_{\theta_F \in \Omega_F} \left[\sup_{X \in U_C} |\hat{F}(\mathbf{X}|\,\theta_F) - F(\mathbf{X})| \right], \tag{22}$$

where Ω_F is the restriction set of θ_F, and is given by

$$\Omega_F = \{\theta_F = [\theta_F^1\ \theta_F^2\ \ldots\ \theta_F^h]^{\mathrm{T}} \in \mathbf{R}^h |\ \|\theta_F\| \le m_F\}, \tag{23}$$

where m_F is the positive constants specified by the designer. By defining the minimum approximation error as

$$\Delta = \Delta(\mathbf{X}, \theta_F) = \hat{F}(\mathbf{X}|\theta_F^*) - F(\mathbf{X}) \tag{24}$$

the error dynamic equation (20) can be written as follows:

$$\dot{\mathbf{E}} = \mathbf{AE} + \mathbf{B}\Delta + \mathbf{B}\big[\hat{F}(\mathbf{X}|\theta_F) - \hat{F}(\mathbf{X}|\theta_F^*) + u_r\big]. \tag{25}$$

In this paper, we use fuzzy logic systems, of the form eq. (2), for approximation, and we have

$$\hat{F}(\mathbf{X}|\theta_F) = \theta_F^{\mathrm{T}}\xi_F(\mathbf{X}), \tag{26}$$

where θ_F is the link weights of FNN between layers 3 and 4 and $\xi_F(\mathbf{X})$ is the vector of fuzzy basis functions (FBFs). Now, by substituting eq. (26) into eq. (25), we have

$$\dot{\mathbf{E}} = \mathbf{AE} + \mathbf{B}\Delta + \mathbf{B}[\phi_F^{\mathrm{T}}\xi_F(\mathbf{X}) + u_r], \tag{27}$$

where $\phi_F = \phi_F(t) = \theta_F(t) - \theta_F^*$.

We consider the following Lyapunov-like function candidate:

$$V' = V'(\mathbf{E}, \phi_F) = \frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{PE} + \frac{1}{2\gamma}\phi_F^{\mathrm{T}}\phi_F, \tag{28}$$

where $\mathbf{P}$ is the solution of eq. (14) with a symmetric $\mathbf{Q} > 0$, and γ is a positive constant. It is obvious that V' is a positive-definite function and for any t, $V'(t) \ge 0$. In fact, $V'(t)$ represents a time-varying Euclidean measure of the distance of the plant output from the reference signal plus the distance of the parameters of the fuzzy neural controller from their optimal values. Ideally, if $V'(t) = 0$, it means that the tracking error is zero and the parameters have reached their optimal values, i.e. $\mathbf{E}(t) = 0$ and $\theta_F(t) = \theta_F^*$. The time derivative of V', using eq. (14), along the trajectories of eq. (27) yields

$$\dot{V}'(t) = -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{QE} + \mathbf{E}^{\mathrm{T}}\mathbf{PB}\Delta + \frac{1}{\gamma}\phi_F^{\mathrm{T}}\big[\dot{\theta}_F + \gamma\mathbf{E}^{\mathrm{T}}\mathbf{PB}\xi_F(\mathbf{X})\big] + \mathbf{E}^{\mathrm{T}}\mathbf{PB}u_r, \tag{29}$$

where $\dot{\phi}_F = \dot{\theta}_F$ (since θ_F^* is fixed). Note that we used an unknown but fixed (i.e., time-invariant) plant. In practice, however, the adaptive controller can be applied to slowly time-varying and unknown plants [35].

According to eq. (29), the following adaptive law for adjusting the link weight of FNN between layers 3 and 4 is given:

$$\dot{\theta}_F = -\gamma \mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\xi_F(\mathbf{X}). \tag{30}$$

Applying adaptive law (30) and robust controller (18) to eq. (29), we have

$$\begin{aligned}\dot{V}'(t) &= -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{Q}\mathbf{E} + \mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\Delta + \mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}u_r \\ &\le -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{Q}\mathbf{E} + |\mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}|\bar{\varepsilon} - \mathrm{sgn}(\mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B})\mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\cdot\bar{\varepsilon} \\ &\le -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{Q}\mathbf{E} \le 0. \end{aligned}\tag{31}$$

Note that the adaptive law defined by eq. (30) does not guarantee that $\theta_F \in \Omega_F$. So, the modified adaptive law is given by using the projection algorithm [30]

$$\dot{\theta}_F = \begin{cases} -\gamma\mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\xi_F(\mathbf{X}), \ \|\theta_F\| < m_F \ \text{or} \ \|\theta_F\| = m_F \ \text{and} \ \mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\theta_F^{\mathrm{T}}\xi_F(\mathbf{X}) \ge 0, & \text{(32a)} \\ \Pr\{-\gamma\mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\xi_F(\mathbf{X})\}, \ \|\theta_F\| = m_F \ \text{and} \ \mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\theta_F^{\mathrm{T}}\xi_F(\mathbf{X}) < 0, & \text{(32b)} \end{cases}$$

and the projection operator $\Pr(\cdot)$ is defined by

$$\Pr\{-\gamma\mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\xi_F(\mathbf{X})\} = -\gamma\mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\xi_F(\mathbf{X}) + \gamma\mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\frac{\theta_F\theta_F^{\mathrm{T}}\xi_F(\mathbf{X})}{\|\theta_F\|^2}. \tag{33}$$

Lemma 2. *Let* $\mathbf{x}$ *and* $\mathbf{y}$ *are real vectors of appropriate dimensions. If* $\|\mathbf{x}\| \ge \|\mathbf{y}\|$, *we have*

$$(\mathbf{x}-\mathbf{y})^{\mathrm{T}}\mathbf{x} \ge 0. \tag{34}$$

Proof. Since

$$(\mathbf{x}-\mathbf{y})^{\mathrm{T}}x = \frac{1}{2}(\|\mathbf{x}\|^2 - \|\mathbf{y}\|^2 + \|\mathbf{x}-\mathbf{y}\|^2), \tag{35}$$

and using $\|\mathbf{x}\| \ge \|\mathbf{y}\|$, we get

$$(\mathbf{x}-\mathbf{y})^{\mathrm{T}}\mathbf{x} \ge \frac{1}{2}(\|\mathbf{x}\|^2 - \|\mathbf{y}\|^2) \ge 0. \tag{36}$$

This completes the proof. □

Theorem 2. *Consider that the error dynamic system* (20) *with the controller u is given by eq.* (19), *where* u_c *is given by eq.* (11), u_r *given by eq.* (18), *and the adaptive law of the FNN weights by eq.* (32). *Then the error dynamics system* (20) *is of asymptotical stability, that is*, $\lim_{t\to\infty}\|\mathbf{Y}_m(t) - \mathbf{X}(t)\| = 0$.

Proof. The time derivative of V' along the trajectories of eq. (27) yields

$$\dot{V}'(t) = -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{Q}\mathbf{E} + \mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\Delta + \frac{1}{\gamma}\phi_F^{\mathrm{T}}[\dot{\theta}_F + \gamma\mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\xi_F(\mathbf{X})] + \mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}u_r. \tag{37}$$

If $\theta_F \in \Omega_F$, using eq. (32a) yields

$$\frac{1}{\gamma}\phi_F^{\mathrm{T}}[\dot{\theta}_F + \gamma\mathbf{E}^{\mathrm{T}}\mathbf{P}\mathbf{B}\xi_F(\mathbf{X})] = 0. \tag{38}$$

If $\theta_F \notin \Omega_F$, then $\|\theta_F\| \geq m_F \geq \left\|\theta_F^*\right\|$. By using eq. (32b) and Lemma 2, we have

$$\frac{1}{\gamma}\phi_F^{\mathrm{T}}[\dot{\theta}_F + \gamma \mathbf{E}^{\mathrm{T}}\mathbf{PB}\xi_F(\mathbf{X})] = \mathbf{E}^{\mathrm{T}}\mathbf{PB}\theta_F\xi_F(\mathbf{X})\left(\theta_F - \theta_F^*\right)\theta_F^{\mathrm{T}}\big/\|\theta_F\|^2 \leq 0. \quad (39)$$

Then from eqs (37)–(39), we have

$$\begin{aligned}\dot{V}'(t) &\leq -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{QE} + \mathbf{E}^{\mathrm{T}}\mathbf{PB}\Delta + \mathbf{E}^{\mathrm{T}}\mathbf{PB}u_r \\ &\leq -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{QE} + |\mathbf{E}^{\mathrm{T}}\mathbf{PB}|\bar{\varepsilon} - \mathrm{sgn}(\mathbf{E}^{\mathrm{T}}\mathbf{PB})\mathbf{E}^{\mathrm{T}}\mathbf{PB}\cdot\bar{\varepsilon} \\ &\leq -\frac{1}{2}\mathbf{E}^{\mathrm{T}}\mathbf{QE} \leq 0. \qquad (40)\end{aligned}$$

Based on Lyapunov stability theorem, we have $\lim_{t\to\infty}\mathbf{E}(t) = 0$. This completes the proof. □

Based on [36], the BP algorithm for tuning online the means and variances of the Gaussian membership functions is discussed. Consider that the structure of FNN is 2-6-9-1, and the Gaussian membership function is defined by

$$\mu A_{jk} = -\frac{(x_j - m_{jk})^2}{(\sigma_{jk})^2}, \quad j = 1, 2;\ k = 1, 2, 3, \quad (41)$$

where m_{jk} and σ_{jk} are respectively, the mean and the variance of the Gaussian function in the kth term of the jth input linguistic variable x_j. The cost function to be minimized is defined as

$$J = J(t) = \frac{1}{2}(e_1(t))^2, \quad (42)$$

$$d_l^3 = \frac{-\partial J}{\partial net_l^3} = \frac{-\partial J}{\partial net_1^4}\frac{\partial net_1^4}{\partial net_l^3} = e_1\cdot\theta_l, \quad l = 1, 2, \ldots, 9, \quad (43)$$

$$\begin{aligned}d_i^2 &= \frac{-\partial J}{\partial net_i^2} = \frac{-\partial J}{\partial y_i^2}\frac{\partial y_i^2}{\partial net_i^2} = -\left(\sum_p \frac{\partial J}{\partial net_p^3}\frac{\partial net_p^3}{\partial y_i^2}\right)\cdot\frac{\partial y_i^2}{\partial net_i^2} \\ &= \left(\sum_p d_p^3\cdot y_{i'}^2\right)\cdot y_i^2, \quad i = 1, 2, \ldots, 6, \qquad (44)\end{aligned}$$

where the subscript p denotes the rule node in connection with the ith node in layer 2, and i' denotes the other node in layer 2 which connects the pth node in layer 3. Then, the adaptive rule of m_{jk} is

$$\Delta m_{jk} = -\frac{\partial J}{\partial m_{jk}} = -\frac{\partial J}{\partial net_k^2}\frac{\partial net_k^2}{\partial m_{jk}} = \delta_k^2\cdot\frac{2(y_j^1 - m_{jk})}{\sigma_{jk}^2}, \quad (45)$$

and the adaptive rule of σ_{jk} is

$$\Delta\sigma_{jk} = -\frac{\partial J}{\partial \sigma_{jk}} = -\frac{\partial J}{\partial net_k^2}\frac{\partial net_k^2}{\partial \sigma_{jk}} = \delta_k^2\cdot\frac{2(y_j^1 - m_{jk})^2}{\sigma_{jk}^3}, \quad (46)$$

where $j = 1, 2$, $k = 1, 2, 3$. The parameters of the Gaussian membership functions can be modified as

$$m_{jk}(t+1) = m_{jk}(t) + \eta \Delta m_{jk}(t), \tag{47}$$

and

$$\sigma_{jk}(t+1) = \sigma_{jk}(t) + \eta \Delta \sigma_{jk}(t), \tag{48}$$

where η is the learning rate.

Note that net_q^L and y_q^L denote the summed net input of the qth node and the output of the qth node, respectively, and the superscript denotes the layer number.

Based on the above discussions, the design algorithm of AFNC is described as follows:

Step 1: Initialize the adjustable parameters of FNN, which include the weighting factors in the consequent part, and the means and variances of the Gaussian membership functions in the antecedent part of fuzzy implications.

Step 2: Specify the feedback gain vector $\mathbf{K}$ such that all roots of the polynomial $h(s) = s^n + k_1 s^{n-1} + \cdots + k_n$ are in the open left-half of the complex plane.

Step 3: Specify a $n \times n$ positive-definite symmetric matrix $\mathbf{Q}$, solving eq. (14) to obtain matrix $\mathbf{P}$.

Step 4: Computing the output of FNN (i.e., $\hat{F}(\mathbf{X}|\theta_F)$).

Step 5: Computing the controller u, and apply the controller u to the system (5).

Step 6: Use adaptive law (32) to adjust the weighting factors θ_F, and use learning algorithms (45)–(48) to adjust the means and variances of the Gaussian membership functions in the antecedent part of fuzzy implications.

4. Computer simulations

In this section, we apply the proposed AFNC to control the Duffing chaotic system [37]

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -0.1x_2 - x_1^3 + 12\cos(t). \end{aligned} \tag{49}$$

It can be shown that without control, the system is chaotic. The chaotic motion of the Duffing system is shown in figure 2.

Consider that the system (49) is perturbed by external noise and parameters uncertainty, and is given by

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -0.1x_2 - x_1^3 + 12\cos(t) + \Delta f(X) + d(t) + u(t), \end{aligned} \tag{50}$$

where $d(t) = 0.1\sin(t)$, $\Delta f(X) = 0.1x_1 + \sin(t)x_2$. We now use the AFNC to control system (50), such that the state x_1 tracks the reference trajectory $y_{m1} = y_m(t) = \sin(t)$, and x_2 tracks $y_{m2} = \dot{y}_m(t) = \cos(t)$. We choose $k_1 = 2$ and $k_2 = 1$ (i.e. $s^2 + k_1 s + k_2$ is stable), and $\mathbf{Q} = \text{diag}(10,10) > 0$. Then, by solving eq. (14), we get

$$\mathbf{P} = \begin{bmatrix} 15 & 5 \\ 5 & 5 \end{bmatrix}. \tag{51}$$

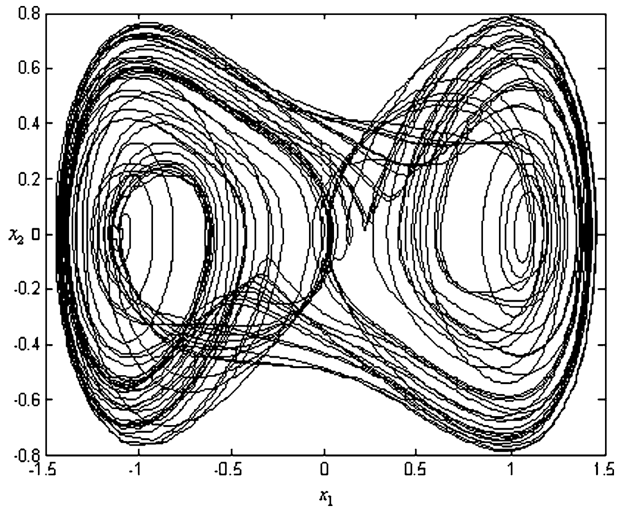

Figure 2. The chaotic attractor of the Duffing system.

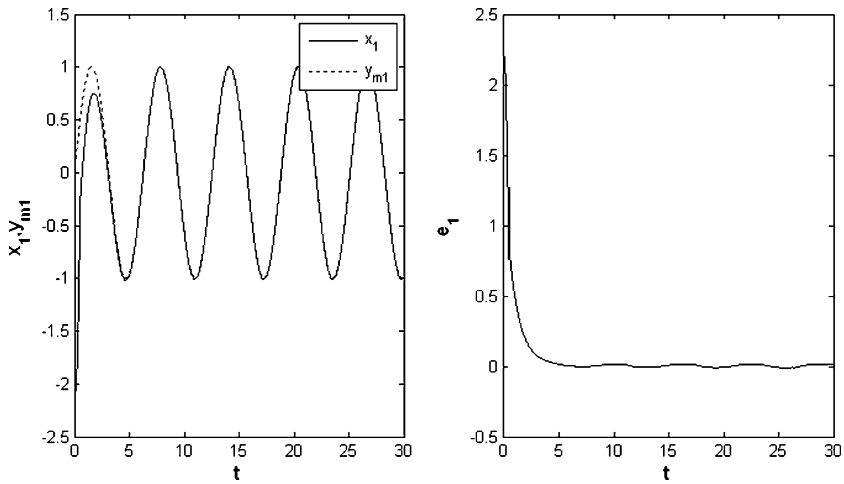

Figure 3. Trajectories of $x_1(t)$, $y_m(t)$ and the corresponding error $e_1(t)$.

We also choose $\gamma = 2$, $\eta = 0.01$ and $\bar{\varepsilon} = 0.1$. In this paper, we use an FNN to represent the approximation of unknown function $F(\mathbf{X})$. The structure of FNN is 2-6-9-1. The initial FNN parameters are selected randomly, i.e. $\theta_F \in [-12,12]$, $\mathbf{m}_F \in [-2,2]$, $\boldsymbol{\sigma}_F \in [0,1]$. Here, θ_F represents the link weigh-vector, $\mathbf{m}_F$ represents the mean vector of the Gaussian membership functions and $\boldsymbol{\sigma}_F$ represents the variance vectors of the Gaussian membership functions. We choose the initial system states $\mathbf{X}(0) = [-2\ -2]$. The simulation results are shown in figures 3–5.

From figures 3 and 4, we can see that the tracking performance is achieved. To illustrate the control energy, the control input is given by figure 5. The control input is appropriate

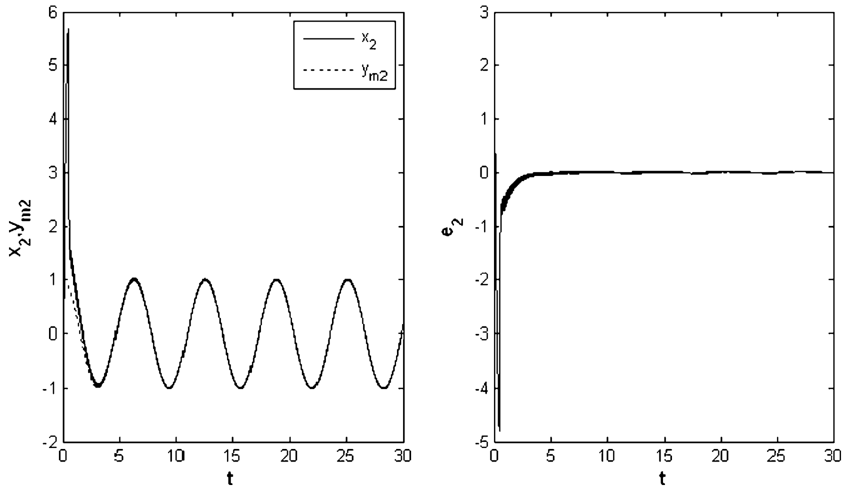

Figure 4. Trajectories of $x_2(t)$, $\dot{y}_m\ (t)$ and the corresponding error $e_2(t)$.

for tracking the external reference signal. In [6], small continuous-time perturbation drive chaotic systems approximately to an unstable periodic orbit (UPO). The control input is larger than the small perturbation as the control object is different.

5. Conclusions

In this paper, we have developed an adaptive fuzzy neural control scheme for a class of unknown chaotic systems. To design the AFNC, exact knowledge of the system is not

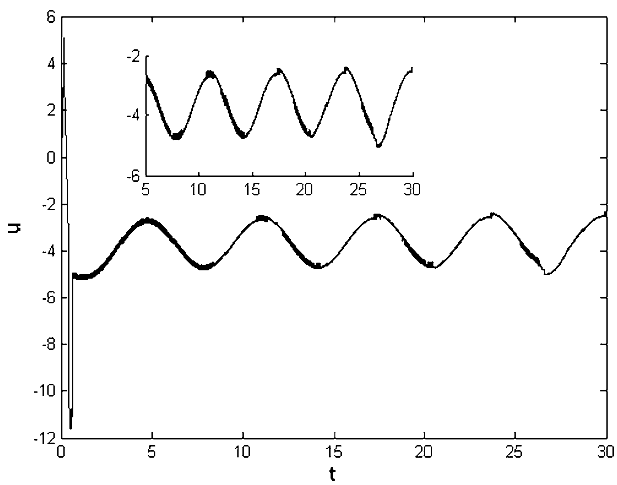

Figure 5. The control input.

needed. In addition, the online tuning parameters include the weighting factors in the consequent part, and the means and variances of the Gaussian membership functions in the antecedent part of the fuzzy implications. The overall adaptive scheme guarantees that the output of the closed-loop system asymptotically track the desired output trajectory. Finally, this method has been applied to control the Duffing chaotic system to track a reference trajectory. The computer simulation results show that the AFNC can perform successful control and achieve the desired performance.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (Nos 60573172, 60973152), the Superior University Doctor Subject Special Scientific Research Foundation of China (No. 20070141014) and the Natural Science Foundation of Liaoning province (No. 20082165).

References

[1] G Chen and X Yu (Eds), *Chaos control: Theory and applications* (Springer-Verlag, Heidelberg, 2003)
[2] G Chen and J Lü, *Dynamics of the Lorenz system family: Analysis, control, and synchronization* (Science Press, Beijing, 2003)
[3] E Ott, C Grebogi and J A Yorke, *Phys. Rev. Lett.* **64**, 1196 (1990)
[4] G Chen and X Dong, *From chaos to order: Methodologies, perspectives and applications* (World Scientific, Singapore, 1998)
[5] G Chen and X Dong, *IEEE Trans. Circuits Syst. I* **40**, 591 (1993)
[6] P Mitra, *IEEE Trans. Circuits Syst. I* **47**, 1649 (2000)
[7] G Chen and X Yu, *IEEE Trans. Circuits Syst. I* **46**, 767 (1999)
[8] X Guan, C Chen, Z Fan and H Peng, *Int. J. Bifurcat. Chaos* **13**, 193 (2003)
[9] K Keiji, H Michio and K Hideki, *Phys. Lett.* **A245**, 511 (1998)
[10] M Jang, C L Chen and C K Chen, *Int. J. Bifurcat. Chaos* **12**, 1437 (2002)
[11] C Fuh and P Tung, *Phys. Lett.* **A218**, 240 (1996)
[12] X Yu, *Chaos, Solitons and Fractals* **8**, 1577 (1997)
[13] Y Yu and S Zhang, *Chaos, Solitons and Fractals* **15**, 897 (2003)
[14] M Bernardo, *Phys. Lett.* **A214**, 139 (1996)
[15] M Feki, *Chaos, Solitons and Fractals* **15**, 883 (2003)
[16] Y J Cao, *Phys. Lett.* **A270**, 171 (2000)
[17] X Wang and M Wang, *Int. J. Mod. Phys.* **B22**, 4069 (2008)
[18] B Chen, X Liu and S Tong, *Chaos, Solitons and Fractals* **34**, 1180 (2007)
[19] J H Kim, C H Hyun, E Kim and M Park, *IEEE Trans. Fuzzy Syst.* **15**, 359 (2007)
[20] H G Zhang, M Li, J Yang and D D Yang, *IEEE Trans. Syst. Man Cybern.* **A39**, 437 (2009)
[21] X Wang and J Meng, *Chaos* **18**, 033102 (2008)
[22] M Roopaei, M Z Jahromi and S Jafari, *Chaos* **19**, 013125 (2009)
[23] H-T Yau and C-S Shieh, *Nonlinear Anal.-Real World Appl.* **9**, 1800 (2008)
[24] Y Gao and M J Er, *IEEE Trans. Fuzzy Syst.* **11**, 462 (2003)
[25] Y Gao and M J Er, *IEEE Trans. Neural Netw.* **16**, 475 (2005)
[26] Y J Liu, W Wang, S C Tong and Y S Liu, *IEEE Trans. Syst. Man Cybern.* **A40**, 170 (2010)
[27] Y J Liu and Y Q Zheng, *Nonlinear Dyn.* **57**, 431 (2009)

[28] D Lin and X Wang, *Fuzzy Sets Syst.* **161**, 2066 (2010)
[29] M Jamshidi, N Vadiee and T J Ress, *Fuzzy logic and control* (Prentice-Hall, Englewood Cliffs, NJ, 1993)
[30] L X Wang, *Adaptive fuzzy systems and control: Design and stability analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1994)
[31] Y G Leu, T T Lee and W Y Wang, *IEEE Trans. Syst. Man Cybern.* **B27**, 1034 (1997)
[32] C T Lin and C S G Lee, *IEEE Trans. Computer* **40**, 1320 (1991)
[33] A Isidori, *Nonlinear control systems* (Springer-Verlag, New York, 1989)
[34] J E Slotine and W Li, *Applied nonlinear control* (Prentice-Hall, Englewood Cliffs, NJ, 1991)
[35] M Hojati and S Gazor, *IEEE Trans. Fuzzy Syst.* **10**, 198 (2002)
[36] Y C Chen and C C Teng, *Fuzzy Sets Syst.* **73**, 291 (1995)
[37] L X Wang, *IEEE Trans. Fuzzy Syst.* **1**, 146 (1993)