International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.12 (2015), pp.17-26
http://dx.doi.org/10.14257/ijsip.2015.8.12.03

Robusts Adaptive Interval Type II Fuzzy Neural Network Control for the Synchronization of Uncertain Chaotic Systems

Xiang-jian Chen[1] and Di Li[2]

[1]Jiangsu University of Science and Technology, School of Computer Science and Engineering, China
[2]China Shipbuilding Industry Corporation 723, China.
[1]cj831209@163.com, [2]Lidi19821111@163.com

Abstract

The proposed RAITIIFNNC system is comprised of a interval type II fuzzy neural network identifier and a robust controller. The identifier is utilized for online estimation of the compound uncertainties. The robust controller is used to attenuate the effects of the approximation error so that the perfect tracking and synchronization of chaotic systems are achieved. All the parameter learning algorithms are derived based on Lyapunov stability theorem to ensure network convergence as well as stable synchronization performance. From the simulation example, to synchronize two Lorenz chaotic systems, it has been shown that the effectiveness of the proposed method has been verified.

Keywords: *Uncertain chaotic systems; Chaos synchronization; Adaptive control; Interval type II fuzzy neural network; Lyapunov stability*

1. Introduction

Since the study of chaos synchronization by Pecora and Carroll [1], this topic has received increasing attention [2]. Chaos synchronization can be applied to many areas such as secure communication, chemical reactions, and biological systems and so on. However, it is difficult to establish an appropriate mathematical model for the design of a model-based synchronization system. Thus, some adaptive control methods are proposed for chaos synchronization. Fuzzy control methodologies have emerged in recent years as promising ways to approach nonlinear control problems. Fuzzy control, in particular, has an impact in the control community, because it can provide a simple approach to the control of plants that are complex, uncertain, ill-defined, and have available heuristic knowledge from domain experts for their controllers design [3-5]. However, neural network adaptive control algorithms have attracted attention due to their inherently parallel and highly redundant processing architecture that makes it possible to develop parallel weight update laws. Consequently, the use of the NNs for system identification and control of complex highly uncertain dynamical systems has become an active area of research [6-8].

The FNN possesses the merits of the low-level learning and computational power of neural network and the high-level human knowledge representation from fuzzy theory. Recently, the FNNs are increasingly receiving attention in solving the control problems [9-13].

In general, Type-1 Fuzzy Logic Systems (FLSs) are unable to handle rule uncertainties directly, when the information that is used to construct the rules in a FLS is uncertain. On the other hand, type-2 FLSs involved in this paper whose antecedent or consequent membership functions are type-2 fuzzy sets can handle rule uncertainties. A type-2 FLS is characterized by IF–THEN rules, but its antecedent or consequent sets are type-2. Hence, type-2 FLSs can be used when the circumstances are too uncertain to determine

ISSN: 2005-4254 IJSIP

exact membership grades such as when training data are corrupted by noise. So the type-2 fuzzy neural network has been successfully studied in some control territory [14-18].

This paper addresses the design and analysis of an intelligent control system for the synchronization of a class of chaotic systems with parameter uncertainty. We solve the problem of adaptive synchronization under the standing assumption that the state of the master system that generates the reference trajectories is known, however, no other knowledge of the master system is required. We also assume that the model of the slave system is known but not the exact dynamics. In order to deal with the uncertainties of chaotic systems, a robust adaptive interval type II fuzzy neural network control (RAITIIFNC) system is investigated to achieve the perfect tracking and synchronization of the chaotic systems. The merits of this RAITIIFNC scheme are that not only the stable tracking performance can be guaranteed but also that appropriate network parameters are found to achieve favorable approximation. The rest of this paper is organized as follows. Section 2 gives the problem formulation for nonlinear chaos synchronization. Section 3 presents an interval type II fuzzy neural network. Section 4 develops the RAITIIFNC system and proves its stability by using Lyapunov’s direct method. The simulation results are given in Section 5.

2. Problem Formulation

Consider the uncertain chaotic system in the form of

$$\dot{X} = (A + \Delta A_x)X + f(X) + d_x(t) \tag{1}$$

Where $X = [x_1, x_2, \cdots, x_n] \in R^n$ is the n-dimensional state vector of the system, $A \in R^{n\times n}$ is the a constant matrix, $f(x) \in R^n$ is an unknown smooth nonlinear vector function, $\Delta A_x \in R^{n\times n}$ is the unknown random-varying term, and $d_x(t) \in R^n$ is the external disturbance with unknown upper bound term. In this paper, it is assumed that system (1) exhibits chaotic dynamics. The response system is:

$$\dot{Y} = (A + \Delta A_y)Y + f(Y) + d_y(t) + u \tag{2}$$

Where $Y \in R^n$ represents an n-dimensional state vector of the system, $u \in R^n$ represents the control input, and generally

$$\Delta A_x(t) \neq \Delta A_y(t); d_x(t) \neq d_y(t) \tag{3}$$

The control objective is to force the state vector X to follow the desired trajectory Y. Let us define the tracking error as $e = [e_1, e_2, \cdots e_n]^T \in R^n = Y - X$. If the tracking error vector $e(t)$ trends to zero as t tends to infinity, then the chaotic systems (1) and (2) are synchronous. The dynamical expression of the tracking error that can be derived by subtracting (2) from (1) as

$$\dot{e} = Ae + \Delta A_y(t)Y + f(Y) + d_y(t) - \Delta A_x(t) - f(X) - d_x(t) + u \tag{4}$$

For further development, let us define the compound uncertainty as:

$$D(Z) = \Delta A_y(t)Y + f(Y) + d_y(t) - \Delta A_x(t) - f(X) - d_x(t) \tag{5}$$

Where $Z = [X, Y]^T \equiv [z_1, z_2, \cdots, z_r]^T \in R^{2n\times 1}$. Then Eq. (4) can be rewritten as:

$$\dot{e} = Ae + D(Z) + u \tag{6}$$

From the viewpoint of control theory, the synchronization between systems (1) and (2) is equivalent to the asymptotical stability of the error system (4) at *e*=0. Hence, an ideal controller can be designed as:

$$u = Ke - D(Z) \tag{7}$$

where *K* is a feedback matrix to be determined so that the eigen values of the matrix *A*+*K* lie strictly in the open left half of the complex plane. However, the compound uncertainty ***D***(***Z***) is unknown for the random-varying parameters and external disturbance, the ideal controller (7) cannot be precisely obtained. Alternatively, a robust controller with the ability to online identify the unknown uncertainties can resolve the mentioned problem. Therefore, this paper will propose SAFNC system with adaptive algorithm to make the chaotic systems (1) and (2) synchronous despite the existence of the uncertainties.

3. Structure Learning Algorithm of ITIIFNN

The structure of ITIIFNN [16-17] is depicted in Figure 1; each rule in ITIIFNN is the first Takagi-Sugeno-Kang (TSK) type. The detailed mathematical functions of each layer are introduced as follows:

Layer 1 (Input layer): This layer defines the input variables which first enter the ITIIFNN.

Layer 2 (Membership layer): In this layer, each node performs an interval type-II fuzzy MF. The FOU of this MF can be represented as an interval bound by lower MF $\underline{\mu}_i^j(x_i)$ and

upper MF $\overline{\mu}_i^j(x_i)$:

$$\overline{\mu}_i^j(x_i) = \begin{cases} N(\underline{\zeta}_i^j, \sigma_i^j, x_i), & x_i < \underline{\zeta}_{j1}^i \\ 1, & \underline{\zeta}_i^j \le x_i < \overline{\zeta}_i^j \\ N(\overline{\zeta}_i^j, \sigma_i^j, x_i), & x_i > \overline{\zeta}_i^j \end{cases}$$

$$\underline{\mu}_i^j(x_i) = \begin{cases} N(\underline{\zeta}_i^j, \sigma_i^j, x_i), & x_j \le \dfrac{\underline{\zeta}_i^j + \overline{\zeta}_i^j}{2} \\ N(\overline{\zeta}_i^j, \sigma_i^j, x_i), & x_j > \dfrac{\underline{\zeta}_i^j + \overline{\zeta}_i^j}{2} \end{cases} \tag{8}$$

Where $\zeta_i^j \in [\underline{\zeta}_i^j, \overline{\zeta}_i^j]$ and σ_i^j are, respectively, the mean and the standard deviation of the Gaussian MF of the *j*th partition for the *i*th input variable x_i. That is, the output of each node can be represented as an interval $[\underline{\mu}_i^j, \overline{\mu}_i^j]$.

Layer 3 (Rule layer): Each node in this layer corresponds to one fuzzy rule and performs a fuzzy meet operation to obtain a firing strength F^i which is computed as follows:

$$F^i = [\underline{f}^i, \overline{f}^i] = [\prod_{j=1}^{n} \overline{\mu}_{ji}, \prod_{j=1}^{n} \underline{\mu}_{ji}] \tag{9}$$

Layer 4 (Type-reduction layer): This layer is used to implement the type-reduction, and center-of-sets type-reduction method is adopted here. The centroid of type-II fuzzy set which can be represented by $[\omega_l^j, \omega_r^j]$ which represents link weights should be set first before the computation of y_l, y_r . The outputs y_l, y_r can be computed using Kamik-Mendel iterative algorithms as:

$$y_l = \frac{\sum_{j=1}^{L} \underline{f}^{j} w_l^j + \sum_{j=L+1}^{N} \overline{f}^{j} w_l^j}{\sum_{j=1}^{L} \underline{f}^{j} + \sum_{j=L+1}^{N} \overline{f}^{j}} = W_l^T g_l \tag{10}$$

$$y_r = \frac{\sum_{j=1}^{R} \underline{f}^{j} w_r^j + \sum_{j=R+1}^{N} \overline{f}^{j} w_r^j}{\sum_{j=1}^{R} \underline{f}^{j} + \sum_{j=R+1}^{N} \overline{f}^{j}} = W_r^T g_r \tag{11}$$

Where $W_r = [\omega_r^1, \cdots, \omega_r^N]^T$, $W_l = [\omega_l^1, \cdots, \omega_l^N]^T$ and

$$g_l = \left[\frac{f_l^1}{\sum_{i=1}^{N} f_l^i}, \frac{f_l^2}{\sum_{i=1}^{N} f_l^i}, \cdots, \frac{f_l^N}{\sum_{i=1}^{N} f_l^i} \right]^T$$

$$g_r = \left[\frac{f_r^1}{\sum_{i=1}^{N} f_r^i}, \frac{f_r^2}{\sum_{i=1}^{N} f_r^i}, \cdots, \frac{f_r^N}{\sum_{i=1}^{N} f_r^i} \right]^T \tag{12}$$

Layer 5 (Output layer): Each output node corresponds to one output variable and act as a defuzzifier. Hence, the defuzzified output shown as:

$$y = \frac{y_l + y_r}{2} = \frac{1}{2}(W_l^T g_l + W_r^T g_r) = W^T g(x, \zeta, \sigma) \tag{13}$$

Remark: Initially, there are no fuzzy rules in ITIIFNN. All of the rules are generated online by the structure learning that not only helps automate rule generation, but also locates good initial rule positions for subsequent parameter learning. Furthermore, the structure and parameter adjustment are performed simultaneously.

4. Design of the RAITIIFNN Identifier and Robust Controller

Since the compound uncertainty ***D***(***Z***) is unknown, in the following, a RAITIIFNN identifier is constructed to estimate the dynamics of this uncertain term ***D***(***Z***). The inputs of the RAITIIFNN are the elements in the vector ***Z***, and the output of the RAITIIFNN is the vector ***D***(***Z***). For the ease of notation, define adjustable parameter vectors ***W***, ***m***, and ***ó*** collecting all parameters of the SOFNN as

$$W = [w_{lr1}, w_{lr2}, \cdots, w_{lrn}]^T \in R^{1\times N} \tag{14}$$

$$\mu = [\mu_{lr1}^1, \cdots, \mu_{lrv}^1, \mu_{lr1}^2, \cdots, \mu_{lrv}^2, \mu_{lr1}^N, \cdots, \mu_{lrv}^N]^T \in R^{Nv\times 1} \tag{15}$$

$$\sigma = [\sigma_{lr1}^1, \cdots, \sigma_{lrv}^1, \sigma_{lr1}^2, \cdots, \sigma_{lrv}^2, \sigma_{lr1}^N, \cdots, \sigma_{lrv}^N]^T \in R^{Nv\times 1} \tag{16}$$

Where $w_j = [w_j^1, w_j^2, \cdots, w_j^N]^T$, $j = 1,2,\cdots,n$ .Then, the outputs of the RAITIIFNN can be rewritten in the following vector form as:

$$Y = [y_1, y_2, \cdots, y_n]^T = Wg(Z,\mu,\sigma) \equiv D_{ITIIFNN}(Z,W,\xi,\sigma) \tag{17}$$

Based on the powerful approximation ability, there exists an optimal RAITIIFNN to learn the compound uncertainty ***D***(***Z***) in Eq. (5) such that

$$\begin{aligned} D(Z) = Wg(Z,\mu,\sigma) &\equiv D^*_{ITIIFNN}(Z,W^*,\xi^*,\sigma^*) + \varsigma(Z) \\ &= W^* g^*(Z,\xi^*,\sigma^*) + \varsigma(Z) \\ &= \arg\min_{W,\xi,\sigma}\left[\sup\left\|D_{ITIIFNN}(Z,W,\xi,\sigma)\right\|\right] + \varsigma(Z) \end{aligned} \tag{18}$$

For chaotic attractors are bounded in the phase space. The approximation error $\varsigma(Z)$ is bounded, then, Design the RAITIIFNN identifier as

$$\hat{D}_{ITIIFNN}(Z,\hat{W},\hat{\xi},\hat{\sigma}) = \hat{W}\hat{g}(\hat{W},\hat{\xi},\hat{\sigma}) \tag{19}$$

Where W^*, ξ^*, σ^* are the optimal parameters of W, ξ, σ , in the RAITIIFNN, respectively; and $\hat{W}, \hat{\xi}, \hat{\sigma}$ are the estimates of the optimal parameters. For ease of notation, define an adjustable parameter vector collecting all parameters of the RAITIIFNN as $\hat{W}_{ITIIFNN} = [\hat{W}, \hat{\xi}, \hat{\sigma}]^T$. If the parameter vector $\hat{W}_{ITIIFNN}$ can be stably tuned to approach the optimal parameter vector $W^*_{ITIIFNN} = [W^*, \xi^*, \sigma^*]^T$

The parameter error vector is defined as $\tilde{W}_{ITIIFNN} = W^*_{ITIIFNN} - \hat{W}_{ITIIFNN}$ .In this study, the linearization technique is employed to transform the nonlinear output of the SAIT2NFN into a partially linear form. Then, the Taylor series expansion of $g^*(Z,\xi^*,\sigma^*)$ to $(\hat{\xi},\hat{\sigma})$ can be expressed as:

$$\begin{aligned} g^*(Z,\xi^*,\sigma^*) &= \hat{g}(Z,\hat{\xi},\hat{\sigma}) + \left.\frac{\partial g}{\partial \xi}\right|_{\substack{\xi=\hat{\xi} \\ \sigma=\hat{\sigma}}}(\xi^* - \hat{\xi}) + \left.\frac{\partial g}{\partial \xi}\right|_{\substack{\xi=\hat{\xi} \\ \sigma=\hat{\sigma}}}(\sigma^* - \hat{\sigma}) + O(Z,\tilde{\xi},\tilde{\sigma}) \\ &= \hat{g} + g'_{\xi}\tilde{\xi} + g'_{\sigma}\tilde{\sigma} + O(Z,\tilde{\xi},\tilde{\sigma}) \end{aligned} \tag{20}$$

where $O(Z,\tilde{\xi},\tilde{\sigma})$ is the high order residual term. Accordingly, using Eqs.(20)and (18), the approximation error vector $\hat{\varsigma}(Z)$ can be written as:

$$\begin{aligned} \hat{\varsigma}(Z) &= W^{*T}(\hat{g} + g'_{\xi}\tilde{\xi} + g'_{\sigma}\tilde{\sigma} + O(Z)) - \hat{W}^T\hat{g} + \varepsilon_L(Z) \\ &= (\tilde{W} + \hat{W})^T(\hat{g} + g'_{\xi}\tilde{\xi} + g'_{\sigma}\tilde{\sigma}) - \hat{W}^T\hat{g} + W^{*T}O(Z) + \varepsilon_L(Z) \\ &= \tilde{W}^T\left[\hat{g} + g'_{\xi}(\xi^* - \hat{\xi}) + g'_{\sigma}(\sigma^* - \hat{\sigma})\right] + \hat{W}^T(g'_{\xi}\tilde{\xi} + g'_{\sigma}\tilde{\sigma}) + W^{*T}O(Z) + \varepsilon_L(Z) \\ &= \tilde{W}^T\left[\hat{g} - g'_{\xi}\tilde{\xi} - g'_{\sigma}\tilde{\sigma}\right] + \tilde{W}^T(g'_{\xi}\tilde{\xi} + g'_{\sigma}\tilde{\sigma}) + d_o(Z) \\ &= \Psi'^T_{ITIIFNN}\tilde{W}_{ITIIFNN} + d_o(Z) \end{aligned}$$

where

$$\Psi'_{ITIIFNN} = \left[(\hat{g} - g'_{\xi}\tilde{\xi} - g'_{\sigma}\tilde{\sigma}), \hat{W}^T g'_{\xi}\tilde{\xi}, \hat{W}^T g'_{\sigma}\tilde{\sigma}\right]^T$$

$$d_o(Z) = \tilde{W}^T(g_{\xi}^{'}\tilde{\xi} + g_{\sigma}^{'}\tilde{\sigma}) + W^{*T}O(Z) + \varepsilon_L(Z)$$

Assume that the bound of the overall approximation error $d_o(Z)$ is given, *i.e.*, $\left\|d_o(Z)\right\| \leq \bar{d}_o$. Now, the NFN-based controller is designed as:

$$u_{ITIIFNN} = Ke - \hat{D}(Z, \hat{W}, \hat{\xi}, \hat{\sigma}) \tag{21}$$

The parameter learning occurs concurrently with the structure-learning. For each piece of incoming data, all free SAIT2NNN parameters are tuned, whether the rules are newly generated or originally existent, the update laws of the consequent parameters W_l^j, W_r^j are computed as:

$$\begin{aligned} W_l^j(t+1) &= W_l^j(t) + \Delta W_l^j(t) \\ &= W_l^j(t) - \eta_w \frac{\partial E(t)}{\partial W_l^i} \\ W_r^j(t+1) &= W_r^j(t) + \Delta W_r^j(t) \\ &= W_r^j(t) - \eta_w \frac{\partial E(t)}{\partial W_r^i} \end{aligned} \tag{22}$$

Where η_w is the learning-rate parameter for the weighting interval factors. Since the type-reduction nodes and the rule nodes do the type-reduction and product operation, respectively, only e error signals should be propagated in these two layers. In the member layer, the update laws for parameters in input MFs are derived as:

$$\begin{aligned} \underline{\xi}_i^j(t+1) &= \underline{\xi}_i^j(t) + \Delta\underline{\xi}_i^j(t) \\ &= \underline{\xi}_i^j(t) - \eta_{\xi}\frac{\partial E(t)}{\partial \underline{\xi}_i^j} \\ &= \underline{\xi}_i^j(t) + \frac{1}{2}\eta_{\xi}e(t)\left[\frac{(w_{lr}^j - D(Z)_{lr})}{\sum_{j=1}^N f_{lr}^j} \times (D(Z)(t) - \underline{\xi}_i^j(t)) \times \underline{f}^j \Big/ (\sigma_i^j(t))^2\right] \end{aligned} \tag{23}$$

$$\begin{aligned} \bar{\xi}_i^j(t+1) &= \bar{\xi}_i^j(t) + \Delta\bar{\xi}_i^j(t) \\ &= \bar{\xi}_i^j(t) - \eta_{\xi}\frac{\partial E(t)}{\partial \bar{\xi}_i^j} \\ &= \bar{\xi}_i^j(t) + \frac{1}{2}\eta_{\xi}e(t)\left[\frac{(w_{lr}^j - D(Z)_{lr})}{\sum_{j=1}^N f_{lr}^j} \times (D(Z)(t) - \bar{\xi}_i^j(t)) \times \bar{f}^j \Big/ (\sigma_i^j(t))^2\right] \end{aligned} \tag{24}$$

$$\begin{aligned} \sigma_i^j(t+1) &= \sigma_i^j(t) + \Delta\sigma_i^j(t) \\ &= \sigma_i^j(t) - \eta_{\sigma}\frac{\partial E(t)}{\partial \sigma_i^j} \\ &= \sigma_i^j(t) + \frac{1}{2}\eta_{\sigma}e(t)\left[\frac{(w_{lr}^j - D(Z)_{lr})}{\sum_{j=1}^N f_{lr}^j} \times (D(Z)(t) - \xi_i^j(t))^2 \times f_{lr}^i \Big/ (\sigma_i^j(t))^3\right] \end{aligned} \tag{25}$$

Lemma 1

Let ***x*** and ***y*** be the real vectors of appropriate dimensions. if ‖*x*‖≥‖*y*‖, we have $(x-y)^T x \geq 0$

Proof

Since

$$(x-y)^T x = \frac{1}{2}(\|x\|^2 - \|y\|^2 + \|x-y\|^2)$$

and using ‖*x*‖≥‖*y*‖, we get

$$(x-y)^T x = \frac{1}{2}(\|x\|^2 - \|y\|^2 + \|x-y\|^2) \geq 0$$

This completes the proof.

Now we design the robust controller as:

$$u_{rj} = -\operatorname{sgn}(e^T p_j)\overline{d}_o \qquad (j = 1,2,\cdots,n) \tag{26}$$

Then de synthesis controller as:

$$u = u_r + u_{ITIIFNN} \tag{27}$$

Theorem 1

Consider the uncertain chaotic system(1)and the expectation chaotic system(2), if the NFN-based controller is designed as (27) and the adaptation laws of the NFN parameters are designed as (23), (24) and (25), then all NFN parameters and the tracking error **e** are uniformly ultimately bounded.

Proof

Consider the Lyapunov function as

$$V(e(t), \tilde{W}, \tilde{\xi}, \tilde{\sigma}) = \tilde{W}^T_{ITIIFNN}(T)P^{-1}(t)\tilde{W}_{ITIIFNN} \tag{28}$$

Where P is chosen to be a real symmetric positive matrix satisfying $A^T P + pA = -Q < 0$,and it is the input covariance matrix, which is a positive define symmetric matrix and has the following property:

$$P(t+1) = P(t) - \frac{P(t)\Psi'_{ITIIFNN}(t+1)\Psi'^T_{ITIIFNN}(t+1)P(t)}{1+\Psi'^T_{ITIIFNN}(t+1)P(t)\Psi'_{ITIIFNN}(t+1)}$$

$$P^{-1}(t+1) = P^{-1}(t) + \Psi'_{ITIIFNN}(t+1)\Psi'^T_{ITIIFNN}(t+1) \tag{29}$$

The time derivative of the Lyapunov function *V* with respect to time is given by

$$\begin{aligned}\Delta V(t) &= \tilde{W}^{T}_{ITIIFNN}(t+1)P^{-1}(t+1)\tilde{W}_{ITIIFNN}(t+1)-\tilde{W}^{T}_{ITIIFNN}(t)P^{-1}(t)\tilde{W}_{ITIIFNN}(t)\\ &= \frac{-[\Psi'^{T}_{ITIIFNN}(t+1)\tilde{W}_{ITIIFNN}(t)]^{2}-\Psi'^{T}_{ITIIFNN}(t+1)\tilde{W}_{ITIIFNN}(t)d_{o}(t)}{1+\Psi'^{T}_{ITIIFNN}(t+1)P(t)\Psi'_{ITIIFNN}(t+1)}\\ &= \frac{-[e(t)-d_{o}(t)]^{2}-[e(t)-d_{o}(t)]d_{o}(t)}{1+\Psi'^{T}_{ITIIFNN}(t+1)P(t)\Psi'_{ITIIFNN}(t+1)}-[[e(t)-d_{o}(t)]d_{o}(t)]\\ &\quad+\frac{\Psi'^{T}_{ITIIFNN}(t+1)P(t)\Psi'_{ITIIFNN}(t+1)e(t)d_{o}(t)}{1+\Psi'^{T}_{ITIIFNN}(t+1)P(t)\Psi'_{ITIIFNN}(t+1)}+\bar{d}_{o}\end{aligned} \tag{30}$$

Define a set as:

$$\Omega_{e}(\bar{d}_{o})=\left\{e(t):e(t)\le\sqrt{[1+\Psi'^{T}_{ITIIFNN}(t+1)P(t)\Psi'_{ITIIFNN}(t+1)]\bar{d}_{o}}\right\} \tag{31}$$

by Barbalat's Lemma, the stability of the learning scheme is guaranteed, and $\lim_{t\to\infty}e(t)=0,\lim_{t\to\infty}\hat{W}_{ITIIFNN}(t)=0$. If $e(t)\in\Omega_{e}(\bar{d}_{o})$,it is possible that $\Delta V(t)>0$,which implies that the weight vector $\hat{W}_{ITIIFNN}(t)$ may drift to infinity. Thus, to ensure the robustness of the learning algorithm, a dead zone can be introduced in the parameter equation as follows:

$$\begin{aligned}&\hat{W}_{ITIIFNN}(t+1)=\hat{W}_{ITIIFNN}(t)+a(t)P(t+1)\Psi'_{ITIIFNN}(t+1)\times e(t)\\ &a(t)=\begin{cases}0 & e(t)\in\Omega_{e}(\bar{d}_{o})\\ 1 & otherwise\end{cases}\end{aligned} \tag{32}$$

5. Computer Simulation

In this section, we apply the proposed RAITIIFNNC system to synchronize between systems (1) and (2). To demonstrate the effectiveness of the proposed method, we choose the well-known Lorenz chaotic system as an example [12]. Consider the uncertain Lorenz chaotic system with random-variable parameters in the form of

$$\begin{cases}\dot{x}_{1}=(-10+rad_{x1}(t))(x_{1}-x_{2})+d_{x1}(t)\\ \dot{x}_{2}=-x_{1}x_{3}+(28+rad_{x2}(t))x_{1}-x_{2}+d_{x2}(t)\\ \dot{x}_{3}=x_{1}x_{2}-(8/3+rad_{x3}(t))x_{3}+d_{x3}(t)\end{cases} \tag{33}$$

and the controlled response system of system (33) is:

$$\begin{cases}\dot{y}_{1}=(-10+rad_{y1}(t))(y_{1}-y_{2})+d_{y1}(t)+u_{1}\\ \dot{y}_{2}=-y_{1}y_{3}+(28+rad_{y2}(t))y_{1}-y_{2}+d_{y2}(t)+u_{2}\\ \dot{y}_{3}=y_{1}y_{2}-(8/3+rad_{y3}(t))y_{3}+d_{y3}(t)+u_{3}\end{cases} \tag{34}$$

where $rad_{xi}(t)\in[-0.1,0.1], rad_{yi}(t)\in[-0.1,0.1]$ denote the random-varying term, $d_{xi}(t), d_{yi}(t)$ denote the external Gaussian white noise, u_{i} denotes the control inputs, and *i*=1,2,3. In general, $rad_{xi}(t)\ne rad_{yi}(t)$ and $d_{xi}(t)\ne d_{yi}(t)$ our control objective is to drive the system state $Y=[y_{1},y_{2},y_{3}]^{T}$ to follow the system state $X=[x_{1},x_{2},x_{3}]^{T}$ The RAITIIFNNC used here for approximating ***D(Z)***, and its inputs are the elements in the vector $\boldsymbol{Z}=[x_1, x_2, x_3, y_1, y_2, y_3]^T$. For the optimal control scheme, the weighting matrices are chosen as follows:

$$K = \begin{bmatrix} 15 & -20 & 0 \\ 14 & -16 & 0 \\ 0 & 0 & -3 \end{bmatrix}, \quad P = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \quad Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$

The control law is shown in Eq. (27), and the update laws of network parameters are presented in Eqs. (22), (23) (24) and (25). The parameter values used in the adaptive control system are chosen as: $\eta_w = 0.01, \eta_\xi = 0.01, \eta_\sigma = 0.01$ and $\bar{d}_o = 0.1$. As the RAITIIFNN starts to learn, the means of Gaussian membership functions are selected as the initial system states, and the output weights are randomly chosen in [−1,1]. The initial states of systems (33) and (34) are randomly selected as X=[6,−1,8] and Y=[−2,4,−5], respectively. The simulation results are shown in Figure 2, Figure 3. From Figure 2, the proposed controller can make the tracking errors converge to zeros as time goes; it means that the trajectories of the controlled chaotic system achieve synchronization

6. Conclusion

Thanks to the rule uncertainties and the training data corrupted by noise, the circumstances are too uncertain to determine exact membership grades. A new stable robust adaptive interval type II fuzzy neural controller in which linguistic fuzzy control rules can be directly incorporated into the controller is developed to synchronize two different chaotic systems. The RAITIIFNN control system consists of the type II fuzzy neural network identifier and the robust controller to enhance the tracking performance in spite of the system uncertainties. All adaptive learning laws in the RAITIIFNN control system are derived in the sense the Lyapunov stability theorem to ensure the convergence and stability of the control system. Finally, this method has been applied to control the Lorenz chaotic system with random-variable parameters. The computer simulation results show that the RAITIIFNNC can perform successful control and achieve desired performance.

References

[1] L. M. Pecora and T. L. Carroll, “Synchronization in chaotic system”, Phys. Rev. Lett., vol. 64, **(1990)**, pp. 821–830.

[2] S. Boccaletti, J. Kurths, G. Osipov, D. L. Valladaresb and C.S. Zhou, “The synchronization of chaotic systems”, Phys. Rep, vol. 366 **(2002)**, pp. 1–101.

[3] H.G. Zhang, M. Li, J. Yang and D.D. Yang, “Fuzzy model-based robust networked control for a class of nonlinear systems”, IEEE Trans. Syst. Man Cybern. A, vol. 39 **(2009)**, pp. 437–447.

[4] H.G. Zhang and Y.B. Quan, “Modeling, identification and control of a class of nonlinear system”, IEEE Trans. Fuzzy Syst., vol. 9 **(2001)**, pp. 349–354.

[5] S.C. Tong and Y.M. Li, “Observer-based fuzzy adaptive control for strict-feedback nonlinear system”, Fuzzy Sets Syst., vol. 160, **(2009)**, pp. 1749–1764.

[6] N. Hovakimyan, A. Calise and N. Kim, “Adaptive output feedback control of a class of multi-input multi-output systems using neural networks”, Int. J. Control., vol. 77, **(2004)**, pp. 1318–1329.

[7] T.S. Li, D. Wang, G. Feng, S.-C. Tong and A. DS., “approach to robust adaptive NN tracking control for strict-feedback nonlinear system”, IEEE Trans. Syst. Man Cybern. B, vol. 40, **(2010)**, pp. 915–923.

[8] T.S. Li, G. Feng, D. Wang and S.-C. Tong, “Neural-network-based simple adaptive control of uncertain MIMO nonlinear systems”, IET Cont. Theory Appl., vol. 4, **(2010)**, pp. 1543–1557.

[9] C.M. Lin and C.F, “Hsu. Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings”, IEEE Trans. Fuzzy Syst., vol. 12, **(2004)**, pp. 733–742.

[10] Y.G. Leu, W.Y. Wang and T.T., “Lee. Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems”, IEEE Trans. Neural Networks, vol. 16, **(2005)**, pp. 853–861.

[11] C.-S. Chen and H.-H. Chen, “Robust adaptive neural-fuzzy-network control for the synchronization of uncertain chaotic systems”, Nonlinear Anal.–Real World Appl., vol. 10, **(2009)**, pp. 1466–1479.

[12] D. Lin and X. Wang, “Self-organizing adaptive fuzzy neural control for the synchronization of uncertain chaotic systems with random-varying parameters”, Neurocomputing, vol. 74, no. 12–13, **(2011)** June, pp. 2241-2249.

[13] C.-S. Chen and H.-H. Chen, “Robust adaptive neural-fuzzy-network control for the synchronization of uncertain chaotic systems”, Nonlinear Analysis: Real World Applications, vol. 10, no. 3, **(2009)** June, pp. 1466-1479.

[14] T.-C. Lin, M.-C. Chen and M. Roopaei, “Synchronization of uncertain chaotic systems based on adaptive type-2 fuzzy sliding mode control”, Engineering Applications of Artificial Intelligence, vol. 24, no. 1, **(2011)** February, pp. 39-49.

[15] S. P. Fard and Z. Zainuddin, “Interval type-2 fuzzy neural networks version of the Stone–Weierstrass theorem”, Neurocomputing, vol. 74, no. 14–15, **(2011)** July, pp. 2336-2343.

[16] J. R. Castro, O. Castillo, P. Melin and A. Rodríguez-Díaz, “A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks”, Information Sciences, vol. 179, no. 13, **(2009)** June 13, pp. 2175-2193.

[17] F.-J. Lin, S.-Y. Chen, P.-H. Chou and P.-H. Shieh, “Interval type-2 fuzzy neural network control for X–Y–Theta motion control stage using linear ultrasonic motors”, Neurocomputing, vol. 72, no. 4–6, **(2009)** January, pp. 1138-1151.

[18] C.-S. Chen and W.-C. Lin, “Self-adaptive interval type-2 neural fuzzy network control for PMLSM drives”, Expert Systems with Applications, vol. 38, no. 12, **(2011)** November–December, pp. 14679-14689.