Proceedings of the 41st IEEE
Conference on Decision and Control
Las Vegas, Nevada USA, December 2002

Nonlinear Identification and Control Using a Generalized Fuzzy Neural Network

Yang Gao and Meng Joo Er, *Member, IEEE*
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
ygao@pmail.ntu.edu.sg, emjer@ntu.edu.sg

Abstract

This paper presents a Robust Adaptive Fuzzy Neural Controller (RAFNC) suitable for identification and control of uncertain MIMO nonlinear systems. The proposed controller has the following salient features: 1) Self-organizing fuzzy neural structure, i.e. fuzzy control rules can be generated or deleted automatically; 2) On-line adaptive learning ability of uncertain nonlinear systems; 3) Fast adaptation and learning speed; 4) Ease of incorporating expert knowledge; 5)Adaptive control, where structure and parameters of the RAFNC can be self-adaptive in the presence of disturbances to maintain high control performance; 6) Robust control, where global stability of the system is established using the Lyapunov approach. Simulation studies on an inverted pendulum and a two-link robot manipulator show that the performance of the proposed RAFNC is superior over many existing schemes.

1 Introduction

In the last few decades, fuzzy logic and neural networks have been greatly adopted in model-free adaptive control of nonlinear dynamic systems [1–4]. Moreover, a few hybrid techniques were applied to adaptation of parameters in fuzzy or neural controllers, like sliding mode technique [1, 5], Bayesian probability [6, 7], genetic algorithms [8] and various types of neural networks [9, 10]. However, it turns out that only adjustment of parameters will not be sufficient in many cases. For example, if the number of fuzzy rules or number of hidden layers and neurons is very large, real-time implementation will be difficult if not impossible. More importantly, this reduces the flexibility and numerical processing capability of the controller and results in redundant or inefficient computation. Therefore, the controller structure needs to be adaptive so that a compact fuzzy or neural control system can be obtained. Various techniques have been attempted to adapt fuzzy or neural control structure, such as genetic algorithms [11], evolution strategies [12], Back-Propagation (BP) neural networks [13] and wavelets [14]. However, they have difficulties in initialization of the control structure and the associated parameters, and the problem associated with either BP or wavelets algorithm is that the learning and adaptation speeds are slow.

This motivates us to investigate an adaptive structure and parameter learning algorithm for constructing a fuzzy or neural control system systematically and automatically. The resulting intelligent controller must have fast on-line adaptability to guarantee good real-time control performance. In line with this objective, a superior Robust Adaptive Fuzzy Neural Controller (RAFNC) is designed and developed in this paper. The RAFNC is built based on a Generalized Fuzzy Neural Network (G-FNN) controller [15, 16] incorporating robust adaptive methodology. The G-FNN controller offers a fast on-line learning algorithm, which can recruit or delete fuzzy control rules or neurons dynamically without predefinition. Its outstanding computational efficiency in terms of learning speed, adaptability and generalization has been verified in some of our latest work [17]. In essence, the G-FNN algorithm enables the G-FNN controller to successfully model the nonlinear system dynamics and its uncertainties online.

The rest of the paper is organized as follows. Section 2 introduces the dynamic model of MIMO nonlinear systems under consideration. Section 3 describes the design procedure of the G-FNN controller and the RAFNC in details. Convergence of the G-FNN controller and global stability of the RAFNC are proven using the Lyapunov theory. Section 4 presents simulation results and discussions on an inverted pendulum system and a two-link robot manipulator. Finally, Section 5 concludes the paper.

2 MIMO Nonlinear System Dynamics

The class of nth-order MIMO nonlinear systems considered in this paper, termed *companion form* or *controllability canonical form*, is given by [18]:

$$\mathbf{z}^{(n)} = \mathbf{F}(\underline{\mathbf{z}}) + \mathbf{G}(\underline{\mathbf{z}})\mathbf{u} + \mathbf{D} \tag{1}$$

where

0-7803-7516-5/02/$17.00 ©2002 IEEE

- $\mathbf{u} \in \Re^{n_i}$ and $\mathbf{z} \in \Re^{n_o}$ are the input and output vectors of the MIMO nonlinear system respectively, with n_i and n_o being the total number of system inputs and outputs respectively.
- $\underline{\mathbf{z}} = [\mathbf{z}^T \quad \dot{\mathbf{z}}^T \quad \ldots \quad \mathbf{z}^{(n-1)T}]^T \in \Re^{n_o n}$ is the state vector of the system.
- $\mathbf{F}(\underline{\mathbf{z}}) \in \Re^{n_o}$ and $\mathbf{G}(\underline{\mathbf{z}}) \in \Re^{n_o \times n_i}$ represent smooth nonlinearities of the dynamic system.
- $\mathbf{D} \in \Re^{n_o}$ is an unknown function representing system uncertainties and external disturbances.

Since the MIMO nonlinear system (1) is controllable, the input gain $\mathbf{G}(\underline{\mathbf{z}})$ needs to be invertible for all $\underline{\mathbf{z}} \in U_c$. The function $\mathbf{G}$ is assumed to be known and bounded. The terms $\mathbf{F}$ and $\mathbf{D}$ are assumed to be bounded.

3 Nonlinear Identification and Control Scheme

3.1 Robust Adaptive Fuzzy Neural Controller (RAFNC)

The objective of this paper is to design a Robust Adaptive Fuzzy Neural Controller (RAFNC) for nonlinear systems in companion form given by Eq. (1), which guarantees boundedness of all closed-loop variables and tracking of a given desired signal $\mathbf{z}_d(t)$. We define the tracking error $\mathbf{e} = \mathbf{z}_d - \mathbf{z}$ and the dynamic tracking error $\mathbf{s}$ as follows:

$$\begin{aligned} \mathbf{s} &= \mathbf{z}_r^{(n-1)} - \mathbf{z}^{(n-1)} && (2) \\ &= \mathbf{e}^{(n-1)} + \mathbf{\Upsilon}_{n-2}\mathbf{e}^{(n-2)} + \ldots + \mathbf{\Upsilon}_0\mathbf{e} && (3) \end{aligned}$$

where $\mathbf{z}_r^{(n-1)}$ is the so-called "reference" value of $\mathbf{z}^{(n-1)}$ and $\mathbf{\Upsilon}_i$ is a diagonal matrix whose entries are positive constants.

To achieve asymptotic perfect tracking of a desired configuration $\mathbf{z}_d$ with all $\dot{\mathbf{z}}_d \ldots \mathbf{z}_d^{(n)}$ known and bounded, the perfect control law $\mathbf{u}^*$ can be designed as follows:

$$\begin{aligned} \mathbf{u}^* &= \mathbf{G}(\underline{\mathbf{z}})^{\dagger}[\mathbf{z}_r^{(n)} - \mathbf{F}(\underline{\mathbf{z}}) - \mathbf{D}] + \mathbf{K_D}\mathbf{s} && (4) \\ &= \mathbf{\Omega}(\overline{\mathbf{z}}_r) + \mathbf{K_D}\mathbf{s} && (5) \end{aligned}$$

where $\mathbf{G}(\underline{\mathbf{z}})^{\dagger}$ is the pseudoinverse of $\mathbf{G}$ if $n_i \neq n_o$ or the inverse of $\mathbf{G}$ if $n_i = n_o$, the function $\mathbf{\Omega}(.)$ represents the inverse dynamics of the nonlinear system, $\overline{\mathbf{z}}_r = [\underline{\mathbf{z}}^T \quad \mathbf{z}_r^{(n)T}]^T$, and $\mathbf{K_D}$ is to be designed.

As external disturbances and unmodeled dynamics represented by $\mathbf{D}$ are unknown in practice, the perfect control law cannot be implemented. To circumvent this problem, the G-FNN is proposed to obtain an estimate of the inverse dynamics, $\hat{\mathbf{\Omega}}$ [1]. As a result, Eq. (5) can be established as

$$\begin{aligned} \mathbf{u}^* &= \hat{\mathbf{\Omega}}(\overline{\mathbf{z}}_r|\mathbf{W}^*) + \varepsilon + \mathbf{K_D}\mathbf{s} && (6) \\ &= \mathbf{W}^{*T}\mathbf{\Phi}(\overline{\mathbf{z}}_r) + \varepsilon + \mathbf{K_D}\mathbf{s} && (7) \end{aligned}$$

where ε is the *minimum approximation error vector*, $\mathbf{W}^*$ is the optimal value of weight matrix $\mathbf{W} \in \Re^{N_r(N_i+1)\times N_o}$ for a given regressor vector $\mathbf{\Phi} \in \Re^{N_r(N_i+1)}$, $N_i = n_o(n+1)$ and $N_o = n_i$ are the number of inputs and outputs of the G-FNN respectively, and N_r is the total number of fuzzy control rules inside the G-FNN.

Our proposed robust RAFNC is therefore designed to mimic the perfect control law (7). As depicted in Figure 1, a G-FNN controller is connected in parallel with a linear controller to generate a compensated control signal. The control law of the RAFNC is given by

$$\mathbf{u}_c = \mathbf{W}^T\mathbf{\Phi}(\overline{\mathbf{z}}_r) + \varepsilon_N \cdot \mathrm{sgn}(\mathbf{G}^T\mathbf{P}\mathbf{s}) + \mathbf{K_D}\mathbf{s} \qquad (8)$$

where the operator "$\cdot$" denotes element-by-element multiplication of the vector ε_N and $\mathrm{sgn}(\mathbf{G}^T\mathbf{P}\mathbf{s})$, and $\mathbf{P} \in \Re^{n_o \times n_o}$ is a symmetric positive definite matrix.

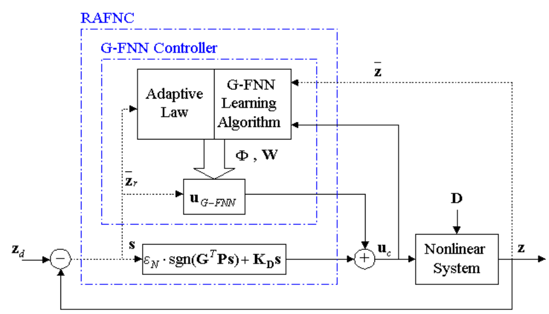

Figure 1: Robust adaptive fuzzy neural control structure.

3.2 Convergence Analysis of RAFNC

From (1), (7) and (8), the system tracking error equation can be shown to be

$$\begin{aligned} \dot{\mathbf{s}} = \mathbf{A}\mathbf{s} + \mathbf{G}(\underline{\mathbf{z}})[(\mathbf{W}^* - \mathbf{W})^T\mathbf{\Phi}(\overline{\mathbf{z}}_r) \\ + \varepsilon - \varepsilon_N \cdot \mathrm{sgn}(\mathbf{G}^T\mathbf{P}\mathbf{s})] \qquad (9) \end{aligned}$$

where $\mathbf{A} = -\mathbf{G}(\underline{\mathbf{z}})\mathbf{K_D} \in \Re^{n_o \times n_o}$ is designed to be a Hurwitz matrix. Eq. (9) shows that $\mathbf{W}$ need to be further adjusted to minimize the tracking error. The adaptive law of $\mathbf{W}$ is designed as follows:

$$\begin{aligned} \dot{\mathbf{w}}_{jk} = a_j\lambda_j\mathbf{\Phi}_j\mathbf{s}^T\mathbf{P}\mathbf{g}_k - (1-a_j)\eta_{jk}\mathrm{sgn}(\mathbf{w}_{jk}) \\ j = 1\ldots N_r,\ k = 1\ldots N_o \qquad (10) \end{aligned}$$

where $\mathbf{w}_{jk} \in \Re^{N_i+1}$ is the weight vector associated with the jth rule for the kth output variable of $\mathbf{W}$, $\mathbf{\Phi} = [\mathbf{\Phi}_1^T \quad \mathbf{\Phi}_2^T \ldots \mathbf{\Phi}_{N_r}^T]^T$, and $\mathbf{G} = [\mathbf{g}_1 \quad \mathbf{g}_1 \ldots \mathbf{g}_{N_o}]$.

[1] Readers can refer to our recent work [17] for detailed descriptions of the G-FNN architecture, learning algorithm and modeling method.

The term $\lambda_j > 0$ is the learning rate for the weight vector associated with the jth rule. The term $\eta_{jk} = \lambda_j \hat{\eta}_{jk} > 0$ is the time-varying decay rate, which is chosen so that

$$\hat{\eta}_{jk} \geq \text{sum}|\mathbf{\Phi}_j \mathbf{s}^T \mathbf{P} \mathbf{g}_k| \quad (11)$$

where the operator "sum" denotes summation of all the elements of vector $|\mathbf{\Phi}_j \mathbf{s}^T \mathbf{P} \mathbf{g}_k|$.

The variable $a_j \in [0,\ 1]$ is a *structural adaptation parameter*, which is obtained from the G-FNN learning algorithm. It reflects the generation and deletion of fuzzy control rules associated with the G-FNN in the following manner. The jth rule is introduced into the G-FNN at time t by setting $a_j(t) = 1$, and the corresponding weight vector is initialized to $\mathbf{w}_{jk}(0)$ acquired by the G-FNN learning algorithm. The jth rule is targeted for deletion from the G-FNN at time t by setting $a_j(t) = 0$, which causes the adaptation mechanism to drive each of the output weights associated with the jth rule node to zero in a time bounded by $\max(\mathbf{w}_{jk})/\eta$. When $a_j(t) = 0$ and weight vectors of the jth rule node have reached zero, the rule is considered deleted from the G-FNN, and its associated memory and computational resources may be freed for reuse.

The symmetric positive definite matrix $\mathbf{P}$ is designed to satisfy the following relationship:

$$\mathbf{PA} + \mathbf{A}^T\mathbf{P} = -\mathbf{Q} \quad (12)$$

where $\mathbf{Q}$ is a positive definite matrix and is selected by the user.

To guarantee stability of the control system, the G-FNN must converge, which requires the parameters of the G-FNN to be bounded. Eq. (3) of [17] shows that the outputs of the G-FNN are bounded if the weights $\mathbf{W}$ are bounded. Define the constraint set $\mathbf{\Gamma}$ for $\mathbf{W}$ as follows:

$$\mathbf{\Gamma} \triangleq \{||\mathbf{w}_{jk}|| \leq ||\mathbf{w}_{jk}(0)||\} \quad (13)$$

where $||.||$ denotes two-norm of a vector. According to the projection algorithm of [2], the adaptation law (10) can be modified as follows:

$$\dot{\mathbf{w}}_{jk} = \begin{cases} a_j\lambda_j \mathbf{\Phi}_j \mathbf{s}^T \mathbf{P} \mathbf{g}_k - (1-a_j)\eta_{jk}\text{sgn}(\mathbf{w}_{jk}) \\ \text{if } (||\mathbf{w}_{jk}|| < ||\mathbf{w}_{jk}(0)||) \text{ or} \\ (||\mathbf{w}_{jk}|| = ||\mathbf{w}_{jk}(0)|| \\ \text{and } \mathbf{w}_{jk}^T \mathbf{\Phi}_j \mathbf{s}^T \mathbf{P} \mathbf{g}_k \leq 0) \\ \\ a_j\lambda_j (\mathbf{I} - \frac{\mathbf{w}_{jk}\mathbf{w}_{jk}^T}{||\mathbf{w}_{jk}||^2}) \mathbf{\Phi}_j \mathbf{s}^T \mathbf{P} \mathbf{g}_k \\ -(1-a_j)\eta_{jk}\text{sgn}(\mathbf{w}_{jk}) \\ \text{if } (||\mathbf{w}_{jk}|| = ||\mathbf{w}_{jk}(0)|| \\ \text{and } \mathbf{w}_{jk}^T \mathbf{\Phi}_j \mathbf{s}^T \mathbf{P} \mathbf{g}_k > 0) \end{cases} \quad (14)$$

Concerning the boundedness of the weights of the G-FNN, we have

Theorem 3.1 If the initial values of the weights $\mathbf{w}_{jk}(0) \in \mathbf{\Gamma}$, the adaptation law (14) guarantees $\mathbf{w}_{jk}(t) \in \mathbf{\Gamma}$, $\forall t > 0$.

Proof: Consider the following Lyapunov function

$$V_{jk} = \frac{1}{2}\mathbf{w}_{jk}^T \mathbf{w}_{jk} \quad (15)$$

Taking the derivative of the Lyapunov function with respect to time

$$\dot{V}_{jk} = \mathbf{w}_{jk}^T \dot{\mathbf{w}}_{jk} \quad (16)$$

When $(||\mathbf{w}_{jk}|| = ||\mathbf{w}_{jk}(0)||$ and $\mathbf{w}_{jk}^T \mathbf{\Phi}_j \mathbf{s}^T \mathbf{P} \mathbf{g}_k \leq 0)$, $\dot{V}_{jk} \leq 0$. Thus, it can be guaranteed that $||\mathbf{w}_{jk}|| \leq ||\mathbf{w}_{jk}(0)||$. When $(||\mathbf{w}_{jk}|| = ||\mathbf{w}_{jk}(0)||$ and $\mathbf{w}_{jk}^T \mathbf{\Phi}_j \mathbf{s}^T \mathbf{P} \mathbf{g}_k > 0)$, $\dot{V}_{jk} \leq 0$. Thus, $||\mathbf{w}_{jk}|| \leq ||\mathbf{w}_{jk}(0)||$ is also guaranteed. Since the initial value $\mathbf{w}_{jk}(0) \in \mathbf{\Gamma}$, $\mathbf{w}_{jk}$ is bounded by the constraint set $\mathbf{\Gamma}$ for all $t \geq 0$. □

3.3 Stability Analysis of RAFNC

Concerning the stability of the closed-loop system, we have the following theorem.

Theorem 3.2 Consider the MIMO nonlinear dynamic system represented by (1). If the robust control law of (8) and the adaptive law of (14) are applied, asymptotic stability is guaranteed.

Proof: We consider the following Lyapunov function candidate, which is based on (9),

$$V(t) = \frac{1}{2}\mathbf{s}^T\mathbf{P}\mathbf{s} + \frac{1}{2}tr[(\mathbf{W}^* - \mathbf{W})^T \mathbf{\Lambda}^{-1}(\mathbf{W}^* - \mathbf{W})] \quad (17)$$

where $\mathbf{\Lambda} \in \Re^{N_r(N_i+1)\times N_r(N_i+1)}$ is a diagonal matrix whose jth $N_i + 1$ entries are the learning rate λ_j.

Taking the derivative of $V(t)$ and using (9) and (14), we have

$$\begin{aligned} \dot{V}(t) &= \frac{1}{2}\dot{\mathbf{s}}^T\mathbf{P}\mathbf{s} + \frac{1}{2}\mathbf{s}^T\mathbf{P}\dot{\mathbf{s}} - tr[(\mathbf{W}^* - \mathbf{W})^T\mathbf{\Lambda}^{-1}\dot{\mathbf{W}}] \\ &= -\frac{1}{2}\mathbf{s}^T\mathbf{Q}\mathbf{s} - \mathbf{s}^T\mathbf{P}\mathbf{G}[(\varepsilon_N \pm |\varepsilon|)\cdot \text{sgn}(\mathbf{G}^T\mathbf{P}\mathbf{s})] \\ &\quad + \sum_{k=1}^{N_o} \mathbf{s}^T\mathbf{P}\mathbf{g}_k \sum_{j=1}^{N_r} (\mathbf{w}_{jk}^* - \mathbf{w}_{jk})^T \mathbf{\Phi}_j \\ &\quad - \sum_{k=1}^{N_o}\sum_{j=1}^{N_r} (\mathbf{w}_{jk}^* - \mathbf{w}_{jk})^T \lambda_j^{-1} \dot{\mathbf{w}}_{jk} \end{aligned} \quad (18)$$

Under condition 1 of (14), (18) becomes

$$\begin{aligned} \dot{V}(t) &= -\frac{1}{2}\mathbf{s}^T\mathbf{Q}\mathbf{s} - \mathbf{s}^T\mathbf{P}\mathbf{G}[(\varepsilon_N \pm |\varepsilon|)\cdot \text{sgn}(\mathbf{G}^T\mathbf{P}\mathbf{s})] \\ &\quad + \sum_{k=1}^{N_o}\sum_{j=1}^{N_r} (1-a_j)(\mathbf{w}_{jk}^* - \mathbf{w}_{jk})^T \end{aligned}$$

$$
\begin{aligned}
& [\mathbf{\Phi}_j \mathbf{s}^T \mathbf{P}\mathbf{g}_k + \hat{\eta}_{jk}\mathrm{sgn}(\mathbf{w}_{jk})] \\
= & -\frac{1}{2}\mathbf{s}^T\mathbf{Q}\mathbf{s} - \mathbf{s}^T\mathbf{P}\mathbf{G}[(\varepsilon_N \pm |\varepsilon|)\cdot \mathrm{sgn}(\mathbf{G}^T\mathbf{P}\mathbf{s})] \\
& + \sum_{k=1}^{N_o}\sum_{j=1}^{N_r} v_{jk} \qquad (19)
\end{aligned}
$$

When $a_j = 1$, i.e. the jth rule is generated, $v_{jk} = 0$. When $a_j = 0$, i.e. the jth rule is going to be deleted and $\mathbf{w}^*_{jk} = 0$, $v_{jk} = -\mathbf{w}^T_{jk}[\mathbf{\Phi}_j \mathbf{s}^T \mathbf{P}\mathbf{g}_k + \hat{\eta}_{jk}\mathrm{sgn}(\mathbf{w}_{jk})] \leq 0$ using (11). Therefore, (19) becomes

$$
\dot{V}(t) \leq -\frac{1}{2}\mathbf{s}^T\mathbf{Q}\mathbf{s} - \mathbf{s}^T\mathbf{P}\mathbf{G}[(\varepsilon_N \pm |\varepsilon|)\cdot \mathrm{sgn}(\mathbf{G}^T\mathbf{P}\mathbf{s})] \leq 0 \quad (20)
$$

Under condition 2 of (14), (18) becomes

$$
\begin{aligned}
\dot{V}(t) = & -\frac{1}{2}\mathbf{s}^T\mathbf{Q}\mathbf{s} - \mathbf{s}^T\mathbf{P}\mathbf{G}[(\varepsilon_N \pm |\varepsilon|)\cdot \mathrm{sgn}(\mathbf{G}^T\mathbf{P}\mathbf{s})] \\
& + \sum_{k=1}^{N_o}\sum_{j=1}^{N_r}(1-a_j)(\mathbf{w}^*_{jk} - \mathbf{w}_{jk})^T \\
& [\mathbf{\Phi}_j \mathbf{s}^T \mathbf{P}\mathbf{g}_k + \hat{\eta}_{jk}\mathrm{sgn}(\mathbf{w}_{jk})] \\
& - \sum_{k=1}^{N_o}\sum_{j=1}^{N_r} a_j\Big(1 - \frac{\mathbf{w}^{*T}_{jk}\mathbf{w}_{jk}}{||\mathbf{w}_{jk}||^2}\Big)\mathbf{w}^T_{jk}\mathbf{\Phi}_j \mathbf{s}^T\mathbf{P}\mathbf{g}_k \\
= & -\frac{1}{2}\mathbf{s}^T\mathbf{Q}\mathbf{s} - \mathbf{s}^T\mathbf{P}\mathbf{G}[(\varepsilon_N \pm |\varepsilon|)\cdot \mathrm{sgn}(\mathbf{G}^T\mathbf{P}\mathbf{s})] \\
& + \sum_{k=1}^{N_o}\sum_{j=1}^{N_r} v_{jk} - \sum_{k=1}^{N_o}\sum_{j=1}^{N_r} u_{jk} \qquad (21)
\end{aligned}
$$

Since $\mathbf{w}^*_i \in \mathbf{\Gamma}$ and $1 - \frac{(\mathbf{w}^*_i)^T\mathbf{w}_i}{||\mathbf{w}_i||^2} \geq 0$, when $a_j = 1$, i.e. the jth rule is generated, $v_{jk} = 0$ and $u_{jk} \geq 0$. When $a_j = 0$, i.e. the jth rule is going to be deleted and $\mathbf{w}^*_{jk} = 0$, $v_{jk} \leq 0$ and $u_{jk} = 0$. Therefore, (21) becomes

$$
\dot{V}(t) \leq -\frac{1}{2}\mathbf{s}^T\mathbf{Q}\mathbf{s} - \mathbf{s}^T\mathbf{P}\mathbf{G}[(\varepsilon_N \pm |\varepsilon|)\cdot \mathrm{sgn}(\mathbf{G}^T\mathbf{P}\mathbf{s})] \leq 0 \quad (22)
$$

Equations (17), (20) and (22) show that $V(t) \geq 0$ and $\dot{V}(t) \leq 0$. Furthermore, (20) and (22) imply that $\dot{V}(t) = 0$ if and only if $V(t) = 0$. Therefore, global stability is guaranteed by the Lyapunov theorem. By using Barbalat's lemma [18], it can be shown that $\mathbf{s}(t) \to 0$ as $t \to \infty$. As a result, the control system is asymptotically stable. Moreover, the tracking error of the system will converge to zero. □

4 Simulation Studies

In this section, two simulation examples are given to verify the validity of the proposed RAFNC for nonlinear systems. Further comparisons with some newly developed adaptive fuzzy or neural controllers further show its advantages in both transient and steady-state performance.

4.1 Example 1: An Inverted Pendulum System

In this example, the effectiveness of the proposed RAFNC design is illustrated in tracking control of an inverted pendulum system used in [2, 18], whose dynamic equation is given by

$$
\ddot{\theta} = \frac{g\sin\theta - \frac{ml\dot{\theta}\cos\theta\sin\theta}{m_c+m}}{l(\frac{4}{3} - \frac{m\cos^2\theta}{m_c+m})} + \frac{\frac{\cos\theta}{m_c+m}}{l(\frac{4}{3} - \frac{m\cos^2\theta}{m_c+m})}u + d \quad (23)
$$

where u represents the input force, θ represents the angle of the pendulum with respect to the vertical line and d represents plant uncertainties.

In this simulation, the system parameters are chosen as follow: mass of the cart, $m_c = 1kg$; mass of the pole, $m = 0.1kg$; half-length of the pole, $l = 0.5m$ and acceleration due to gravity, $g = 9.8m/s^2$. The initial position is chosen as $\theta(0) = \dot{\theta}(0) = \ddot{\theta}(0) = 0$. The desired angle trajectory is assumed to be $\theta_d = 0.1\sin(2\pi t)rad$ and plant uncertainties are given by $d = 0.2\sin(4\pi t)rad/s^2$.

The G-FNN learning algorithm operates at a sampling rate of 50 *sample/sec*, whose parameters are designed as follow: $e_{\max} = 0.1$, $e_{\min} = 0.005$, $d_{\max} = \sqrt{\ln(\frac{1}{0.5})}$, $d_{\min} = \sqrt{\ln(\frac{1}{0.8})}$, $K_{s,\min} = 0.9$, $K_{mf} = 0.5$, $K_{err} = 0.01$, and $N_d = 100$. The rest of the RAFNC operate at a sampling rate of 1000 *sample/sec*, whose parameters are designed as: $\mathbf{\Upsilon}_0 = 2$, $\mathbf{K_D} = 8$, $\mathbf{P} = 8$, $\varepsilon_N = 0.01$ and $\lambda_j = 0.5$.

Using the G-FNN learning algorithm, the fuzzy neural structure and parameters are generated simutaniously and automatically. In this simulation study, a total of three fuzzy rules are generated online as shown in Figure 2. The corresponding Gaussian fuzzy membership functions were obtained with respect to the input training variable as shown in Figure 3. It can be seen that the membership functions are evenly distributed over the input training interval. This is in line with the aspiration of "local representation" in fuzzy logic. Figure 4 shows the tracking result of the control system. The result clearly demonstrates that the pole angle was able to track the desired trajectory from second trial onwards.

4.2 Example 2: An Articulated Two-Link Robot Manipulator

In this simulation example, the performance of the proposed RAFNC is verified in tracking control of a two-link robot manipulator. The dynamic equation is given by[2]

$$
\ddot{\theta} = -\mathbf{M}(\theta)^{-1}\mathbf{H}(\theta,\dot{\theta}) + \mathbf{M}(\theta)^{-1}\tau - \mathbf{M}(\theta)^{-1}\tau_{\mathbf{d}} \quad (24)
$$

[2]Readers can refer to [18] for parameters of the manipulator used in this simulation.

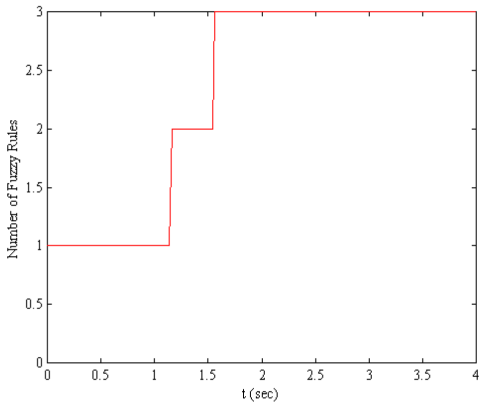

Figure 2: Number of fuzzy rules generated.

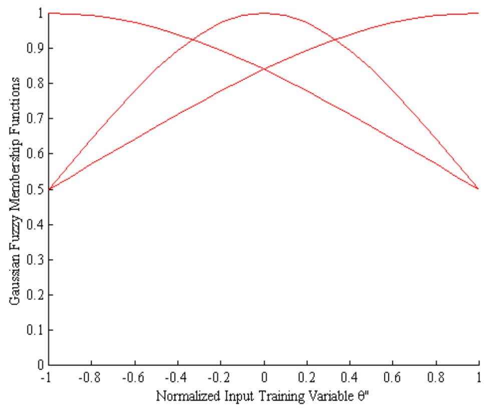

Figure 3: Gaussian fuzzy membership functions w.r.t input training variables.

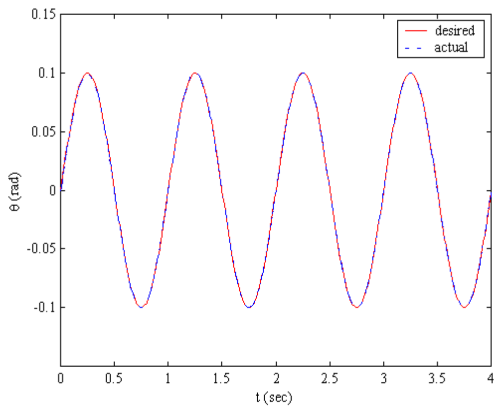

Figure 4: Tracking response using RAFNC.

where $\mathbf{M}$ is the 2×2 manipulator inertia matrix, which is symmetric positive definite, $\mathbf{H}$ is the 2×1 vector of centrifugal, Coriolis, friction forces and gravity, τ is the 2×1 vector of input torque generated by the joint motor, $\ddot{\theta}$, $\dot{\theta}$ and θ are 2×1 vectors of output link acceleration, velocity and position respectively, and $\tau_{\mathbf{d}}$ is the 2×1 vector of unknown terms arising from unmodeled dynamics and external disturbances.

Initial conditions are chosen as $\theta_1(0) = \theta_2(0) = 0\ rad$, $\dot{\theta}_1(0) = \dot{\theta}_2(0) = 0\ rad/sec$ and $\ddot{\theta}_1(0) = \ddot{\theta}_2(0) = 0\ rad/sec^2$. The desired trajectories are $\theta_{d1} = \frac{\pi}{6}(1 - \cos(2\pi t))\ rad$ and $\theta_{d2} = \frac{\pi}{4}(1 + \cos(2\pi t))\ rad$, and disturbances are $\tau_{d1} = 100\sin(2\pi t)\ Nm$ and $\tau_{d2} = 50\sin(2\pi t)\ Nm$, which are comparable to the control torques of the robot manipulator.

The G-FNN learning algorithm operates at a sampling rate of 50 $sample/sec$, whose parameters are designed as follow: $e_{\max} = 0.1$, $e_{\min} = 0.005$, $d_{\max} = \sqrt{\ln(\frac{1}{0.5})}$, $d_{\min} = \sqrt{\ln(\frac{1}{0.8})}$, $K_{s,\min} = 0.9$, $K_{mf} = 0.5$, $K_{err} = 0.01$, and $N_d = 100$. The rest of the RAFNC operate at a sampling rate of 1000 $sample/sec$. The parameters of the RAFNC are designed as follow: $\mathbf{\Upsilon}_0 = diag[25/7, 25/7]$, $\mathbf{K_D} = diag[7,7]$, $\mathbf{P} = diag[7,7]$, $\varepsilon_N = [0.05\ \ 0.05]^T$, and $\lambda_j = 0.4$.

Simulation studies are carried out to compare the performance of the RAFNC with three different controllers that were developed in similar control structure, i.e. 1) a Computed Torque Controller (CTC) of [18], 2) an Adaptive Neural Controller (ANC) of [3], and 3) an Adaptive Fuzzy Controller (AFC) of [4]. Table 1 shows properties of the designed RAFNC with respect to the other three controllers. Figures 5 compares tracking performance of the RAFNC with the rest. Simulaton results show that the RAFNC achieves better transient and steady-state performance with much compact fuzzy neural structure.

Remark 4.1 Concerning the control law (8) of the RAFNC, besides the robust term containing ε_N, the RAFNC for 2rd-order systems with the same number of input and output variables (i.e. 2nd-order *square* systems) consists of a feedforward G-FNN controller and a feedback PD controller with $\mathbf{K_p} = \mathbf{K_D}\mathbf{\Upsilon}_0$ and $\mathbf{K_v} = \mathbf{K_D}$.

5 Conclusions

In this paper, a fuzzy neural identification and control scheme for nonlinear systems was proposed and its adaptive capability to handle modeling errors and external disturbances was demonstrated. The error convergence rate with the RAFNC was found to be fast. Asymptotic stability of the control system is established using the Lyapunov approach. Computer simulation studies of an inverted penulum system and a two-link robot manipulator verify the flexibility, adaptation and tracking performance of the proposed RAFNC.

Table 1: Comparison of RAFNC with Other Controllers

	Robot Model	K_p	K_v	Fuzzy Rules or Hidden Neurons
CTC [18]	Require	25	10	0 (Fixed)
ANC [3]	Require	25	10	121 (Fixed)
AFC [4]	Require	25	35	4 (Fixed)
RAFNC	Not Require	25	7	5 (Adaptive)

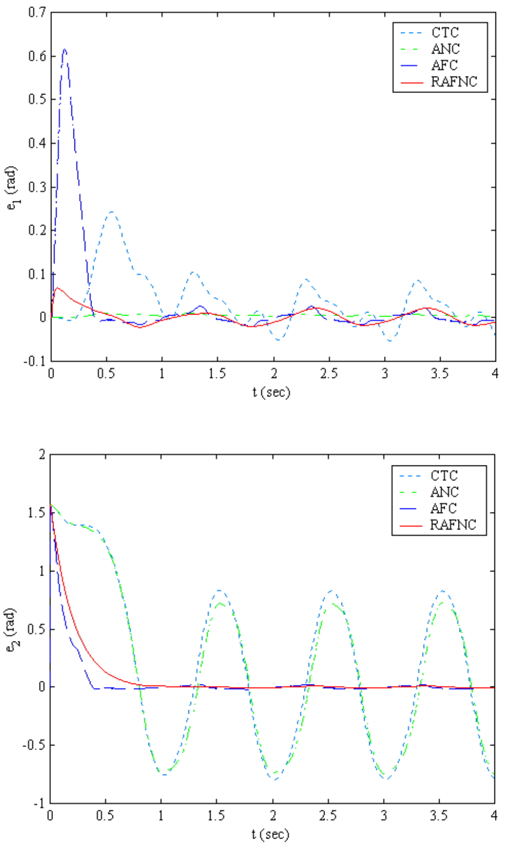

Figure 5: Comparison of tracking errors of RAFNC with other controllers.

References

[1] S. S. Ge, T. H. Lee and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*, World Scientific Publishing, 1998.

[2] L. X. Wang, *A Course in Fuzzy Systems and Control*, New Jersey: Prentice Hall, 1997.

[3] G. Feng, "A New Stable Tracking Control Scheme for Robotic Manipulators." *IEEE Trans. System, Man and Cybernetics*, Vol. 27, No. 3, pp. 510-516, 1997.

[4] M. J. Er and S. H. Chin, "Hybrid Adaptive Fuzzy Controllers of Robot Manipulators with Bounds Estimation," *IEEE Trans. Industrial Electronics*, Vol. 47, No. 5, pp. 1151-1160, 2000.

[5] J. C. Wu and T. S. Liu, "A Sliding Mode Approach to Fuzzy Control Desgin," *IEEE Trans. on Control Systems Technology*, Vol. 4, pp. 459-468, 1996.

[6] Y. Pan, G. J. Klir and B. Yuan, "Bayesian Inference Based on Fuzzy Probabilities," *Proc. 5th IEEE Int. Conf. on Fuzzy Systems*, New Orleans, Louisiana, USA, Vol. 3, pp. 1693-1699, 1996.

[7] D. J. C. MacKay, "Bayesian Neural Networks and Density Networks," *Nuclear Instruments and Methods in Physics Research*, Vol. 354, No. 1, pp. 73-80, 1995.

[8] D. Park, A. Kandel and G. Langholz, "Genetic-Based New Fuzzy reasoning models with Application to Fuzzy Control," *IEEE Trans. on System, Man, and Cybernetics*, Vol. 24, pp. 39-47, 1994.

[9] F. J. Lin, W. J. Hwang and R. J. Wai, "A Supervisory Fuzzy Neural Network Control System for Tracking Periodic Input," *IEEE Trans. Fuzzy Systems*, Vol. 7, No. 1, pp. 41-52, 1999.

[10] J. S. R. Jang, C. T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, NJ: Prentice Hall, 1997.

[11] Y. Jin, "Decentralized Adaptive Fuzzy Control of Robot Manipulators," *IEEE Trans. on System, Man, and Cybernetics*, Vol. 28, pp. 47-58, 1998.

[12] Y. Jin, W. V. Seelen and B. Sendhoff, "On Generating FC^3 Fuzzy Rule Systems from Data Using Evolution Strategies," *IEEE Trans. on System, Man, and Cybernetics*, Vol. 29, No. 6, pp. 829-845, 1999.

[13] J. Nie, and D. Linkens, *Fuzzy-Neural Control: Principles, Algorithms and Applications*, NJ: Prentice Hall, 1995.

[14] M. R. Cannon and J. -J. E. Slotine, "Space-Frequency Localized Basis Function Networks for Nonlinear System Estimation and Control," *Neurocomputing*, Vol. 9, 1995.

[15] Y. Gao, M. J. Er and S. Yang "Adaptive Fuzzy Neural Control of Robot Manipulators," *IEEE Trans. on Industrial Electronics*, Vol. 48, No. 6, pp. 1274-1278, 2001.

[16] M. J. Er and Y. Gao "Robust Adaptive Control of Robot Manipulators Using Generalized Fuzzy Neural Networks," *IEEE Trans. on Industrial Electronics*, to appear, 2002.

[17] S. Q. Wu, M. J. Er and Y. Gao, "A Fast Approach for Automatic Generation of Fuzzy Rules by Generalized Dynamic Fuzzy Neural Networks," *IEEE Trans. on Fuzzy Systems*, Vol. 9, No. 4, pp. 578-594, 2001.

[18] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*, New Jersey: Prentice Hall, 1991.