

## ONLINE ADAPTIVE FUZZY NEURAL IDENTIFICATION AND CONTROL OF A CLASS OF MIMO NONLINEAR SYSTEMS

Yang Gao\* and Meng Joo Er\*

*\* Instrumentation and System Engineering Lab,  
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 (AFNC) suitable for identification and control of a class 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) Online learning ability of uncertain MIMO nonlinear systems; (3) Fast learning speed; (4) Adaptive control; (5) Robust control, where global stability of the system is established using the Lyapunov approach. Simulation example is included to confirm the validity and performance of the proposed control algorithm.

**Keywords:** Adaptive control, Fuzzy logic, Neural networks, MIMO system

### 1. INTRODUCTION

Design of robust adaptive controllers suitable for real-time control of MIMO nonlinear systems is one of the most challenging tasks for many control engineers, especially when the nonlinear system is required to manoeuvre very quickly under external disturbances. In the last few decades, intelligent control using fuzzy logic and neural networks has undergone rapid development leading to global stability and tracking results for reasonably large classes of nonlinear systems.

Fuzzy logic provides human reasoning capabilities to capture uncertainties, which cannot be described by precise mathematical models. Neural networks offer exciting advantages such as adaptive learning, parallelism, fault tolerance and generalization. They have been proven to be very powerful techniques in the discipline of system control, especially when the controlled system is hard to be modeled mathematically, or when

the controlled system has large uncertainties and strong nonlinearities. Therefore, fuzzy logic and neural networks have been greatly adopted in model-free adaptive control of nonlinear systems (Brown and Harris, 1994; Wang, 1997). Furthermore, a few hybrid techniques were applied to adaptation of parameters in fuzzy and/or neural controllers, like sliding mode control (Chang, 2001), Bayesian probability (MacKay, 1995), genetic algorithms (Park *et al.*, 1994), neuron-like structure (Berenji and Khedkar, 1992), hybrid pi-sigma network (Jin *et al.*, 1995) and RBF neural networks (Jang *et al.*, 1997). 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 or 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 has been attempted to adapt fuzzy or neural control structure, such as genetic algorithms (Jin, 1998), evolution strategies (Jin *et al.*, 1999), backpropagation network (Nie and Linkens, 1995) and wavelets (Cannon and Slotine, 1995). However, they have difficulties in initialization of the control structure and the associated parameters, and problem with either BP or wavelets algorithm is that the learning and adaptation speed are slow.

This motivates us to investigate adaptive learning algorithms for constructing a fuzzy and/or neural control system systematically and automatically. The resulting intelligent controller must have fast online adaptability to guarantee good real-time control performance. In line with this objective, we propose a new Adaptive Fuzzy Neural Controller (AFNC), which is built based on a Generalized Fuzzy Neural Network (G-FNN) controller of (Gao *et al.*, 2001) employing the Generalized Fuzzy Neural Network (G-FNN) learning algorithm. The G-FNN algorithm offers a fast online learning algorithm, which can recruit or delete fuzzy control rules or neurons dynamically without predefinition of the structure. Its outstanding computational efficiency in terms of learning speed, adaptability and generalization has been verified in one of our latest work (Wu *et al.*, 2001). 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 a class of MIMO nonlinear systems under consideration. This is followed by Section 3 that describes the design procedure of the AFNC in details. Convergence of the G-FNN controller and global stability of the closed-loop control system are proven using the Lyapunov theory. Section 4 presents simulation results and discussions on a two-link robot manipulator. Finally, Section 5 concludes the paper.

## 2. MIMO NONLINEAR SYSTEM DYNAMICS

The class of  $n$ th-order MIMO nonlinear systems considered in this paper, termed *companion form* or *controllability canonical form*, is given by (Slotine and Li, 1991):

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

where

- $\mathbf{u} \in \mathfrak{R}^m$  and  $\mathbf{z} \in \mathfrak{R}^m$  are the input and output vectors of the MIMO nonlinear system respectively. Such systems are called *square*

system since they have as many control inputs as outputs to be controlled.

$$\bullet \mathbf{z} = \begin{bmatrix} \mathbf{z} \\ \dot{\mathbf{z}} \\ \vdots \\ \mathbf{z}^{(n-1)} \end{bmatrix} \in \mathfrak{R}^{mn} \text{ is the state vector of}$$

the system.

- $\mathbf{F}(\mathbf{z}) \in \mathfrak{R}^m$  and  $\mathbf{G}(\mathbf{z}) \in \mathfrak{R}^{m \times m}$  represent smooth nonlinearities of the dynamic system.
- $\mathbf{D} \in \mathfrak{R}^m$  is an unknown function representing system uncertainties and external disturbances.

Since we require that the MIMO nonlinear system (1) is controllable, the input gain  $\mathbf{G}(\mathbf{z})$  needs to be invertible for all  $\mathbf{z} \in U_c \subset \mathfrak{R}^{mn}$ . Functions  $\mathbf{F}$ ,  $\mathbf{G}$  and  $\mathbf{D}$  are assumed to be bounded.

## 3. ADAPTIVE FUZZY NEURAL IDENTIFICATION OF NONLINEAR SYSTEMS

In the context of using Generalized Fuzzy Neural Networks (G-FNN) directly for nonlinear control, G-FNN is viewed as a means of system identification, or even a framework for knowledge representation. The knowledge about system dynamics and mapping characteristics are implicitly stored within the network. In this section, adaptive fuzzy neural identification of MIMO nonlinear systems using G-FNN is elaborated<sup>1</sup>.

Inverse modelling of dynamical systems plays a crucial role in a range of control problems, which will become apparent in the next section. In this paper, direct modelling of system's inverse dynamics by the G-FNN as illustrated in Figure 1, is attempted. It can be easily derived from (1) that the inverse dynamics of the nonlinear system is given by:

$$\mathbf{u}(\bar{\mathbf{z}}) = \mathbf{G}(\mathbf{z})^{-1}[\mathbf{z}^{(n)} - \mathbf{F}(\mathbf{z}) - \mathbf{D}] = \mathbf{\Omega}(\bar{\mathbf{z}}) \quad (2)$$

where vector  $\bar{\mathbf{z}} = \begin{bmatrix} \mathbf{z} \\ \mathbf{z}^{(n)} \end{bmatrix} \in \mathfrak{R}^{m(n+1)}$ . The G-FNN is trained to obtain an estimate of the inverse dynamics,  $\hat{\mathbf{\Omega}}$ , i.e. the one-to-one mapping relationship from  $\bar{\mathbf{z}}$  to  $\mathbf{u}$ . This is achieved by applying the G-FNN learning algorithm, which is able to determine the appropriate structure and parameters of a fuzzy neural system to estimate the mapping relationship.

The resulting G-FNN can therefore be used to estimate the input signal  $\mathbf{u}_{G-FNN}$  of the nonlinear system given the desired output vector  $\bar{\mathbf{z}}$ . From (3) (Wu *et al.*, 2001) and (2), the output of the G-FNN can be shown to be

<sup>1</sup> Detailed description of G-FNN architecture and G-FNN learning algorithm can be referred to (Wu *et al.*, 2001).

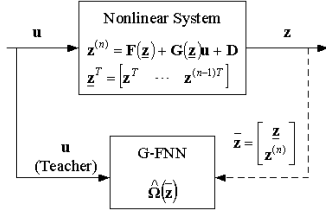


Fig. 1. G-FNN inverse modelling of a nonlinear system

$$\mathbf{u}_{G-FNN}(\bar{\mathbf{z}}|\mathbf{W}) = \hat{\Omega}(\bar{\mathbf{z}}|\mathbf{W}) = \mathbf{W}^T \Phi(\bar{\mathbf{z}}) \quad (3)$$

The optimal parameter  $\mathbf{W}^*$  for a given regressor vector  $\Phi(\bar{\mathbf{z}})$  is defined as follows:

$$\mathbf{W}^* = \arg \min [\sup_{\bar{\mathbf{z}} \in U_c} \|\Omega(\bar{\mathbf{z}}) - \hat{\Omega}(\bar{\mathbf{z}}|\mathbf{W})\|] \quad (4)$$

According to the universal approximation theorem, there exists  $\mathbf{W}^*$  such that  $\hat{\Omega}(\bar{\mathbf{z}}|\mathbf{W}^*)$  can approximate  $\Omega(\bar{\mathbf{z}})$  as close as possible (Chang, 2001). In this paper, we assume that the approximation error is negligible, and therefore (2) can be approximated by the G-FNN as follows:

$$\mathbf{u}(\bar{\mathbf{z}}) = \mathbf{u}_{G-FNN}(\bar{\mathbf{z}}|\mathbf{W}^*) = \mathbf{W}^{*T} \Phi(\bar{\mathbf{z}}) \quad (5)$$

#### 4. ADAPTIVE FUZZY NEURAL CONTROL OF NONLINEAR SYSTEMS

##### 4.1 Adaptive Fuzzy Neural Control Structure

The objective of this paper is to design a robust Adaptive Fuzzy Neural Controller (AFNC) for a class of MIMO nonlinear systems in companion form (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  $\mathbf{e}$  and the tracking error vector  $\underline{\mathbf{e}}$  as follows:

$$\mathbf{e} = \mathbf{z}_d - \mathbf{z} \in \mathfrak{R}^m, \quad \underline{\mathbf{e}} = \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \\ \vdots \\ \mathbf{e}^{(n-1)} \end{bmatrix} \in \mathfrak{R}^{mn} \quad (6)$$

If the parameters of the plant dynamics are well known, the perfect control law  $\mathbf{u}^*$  can be designed by the well-known *feedback linearization* method (Slotine and Li, 1991) as follows:

$$\mathbf{u}^* = \mathbf{G}(\mathbf{z})^{-1} [\mathbf{z}_d^{(n)} - \mathbf{F}(\mathbf{z}) - \mathbf{D} + \mathbf{K}\underline{\mathbf{e}}] \quad (7)$$

$$= \mathbf{u}(\bar{\mathbf{z}}_d) + \mathbf{G}(\mathbf{z})^{-1} \mathbf{K}\underline{\mathbf{e}} \quad (8)$$

where the vector  $\bar{\mathbf{z}}_d = \begin{bmatrix} \mathbf{z}_d \\ \mathbf{z}_d^{(n)} \end{bmatrix} \in \mathfrak{R}^{m(n+1)}$  and the matrix  $\mathbf{K} = [\mathbf{K}_n \quad \mathbf{K}_{n-1} \quad \dots \quad \mathbf{K}_1] \in \mathfrak{R}^{m \times mn}$

which contains real numbers. Substituting (7) into (1) yields

$$\mathbf{e}^{(n)} + \mathbf{K}_1 \mathbf{e}^{(n-1)} + \dots + \mathbf{K}_n \mathbf{e} = 0 \quad (9)$$

which implies that the tracking error will converge to zero with proper choice of  $\mathbf{K}$ . This can be done by choosing  $\mathbf{K}$  such that all roots of the polynomial  $\mathbf{s}^n + \mathbf{K}_1 \mathbf{s}^{n-1} + \dots + \mathbf{K}_n = 0$  are in the open left-half plane.

However, external disturbances and unmodeled dynamics represented by  $\mathbf{D}$  are unknown in practice. Therefore, we cannot implement the perfect control law. To circumvent this problem, the G-FNN is proposed to generate an optimal control law to approximate the perfect control law. The executed perfect control law is obtained by substituting (5) into (8) and it is given by

$$\mathbf{u}^* = \mathbf{W}^{*T} \Phi(\bar{\mathbf{z}}_d) + \mathbf{G}(\mathbf{z})^{-1} \mathbf{K}\underline{\mathbf{e}} \quad (10)$$

The configuration of our proposed robust AFNC system is depicted in Figure 2. The G-FNN controller is connected in parallel with the PD-like controller to generate a compensated control signal. The control law of the AFNC is given by

$$\mathbf{u}_c = \mathbf{W}^T \Phi(\bar{\mathbf{z}}_d) + \mathbf{K}\underline{\mathbf{e}} \quad (11)$$

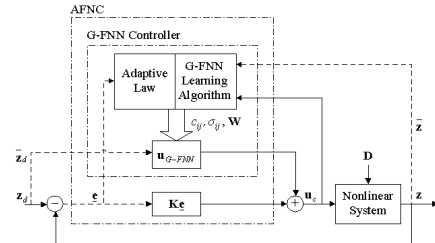


Fig. 2. Adaptive Fuzzy Neural Control Structure

In other words, the G-FNN controller is formed in such a way that it captures the inverse dynamic of the controlled system, i.e. the mapping relationship from  $\bar{\mathbf{z}}$  to  $\mathbf{u}_c$ . This is achieved by applying the G-FNN learning algorithm, which is able to determine the appropriate structure and parameters of a fuzzy neural system to estimate the desired mapping relationship. Its online learning ability enables parameters of the G-FNN, like  $c_{ij}$ ,  $\sigma_{ij}$  and  $\mathbf{W}(0)$ , to be obtained during real-time control of the plant. After the initial value of the weight vector  $\mathbf{W}(0)$  is obtained from the G-FNN learning algorithm,  $\mathbf{W}$  is further adjusted and obtained by the adaptive law derived in Section 4.2. This is to compensate for modeling errors of the G-FNN learning algorithm and fulfill stability requirement of the entire control system. The stability of the AFNC system is proven in Section 4.3.

#### 4.2 Convergence Analysis of AFNC

From (1), (10) and (11), the system tracking error equation can be shown to be

$$\dot{\underline{\mathbf{e}}} = \mathbf{\Lambda}\underline{\mathbf{e}} + \mathbf{B}(\mathbf{u}^* - \mathbf{u}_c) \quad (12)$$

$$\mathbf{\Lambda} = \begin{bmatrix} 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{I} \\ -\mathbf{K}_n & -\mathbf{K}_{n-1} & \dots & -\mathbf{K}_1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{G}(\underline{\mathbf{z}}) \end{bmatrix}$$

Substituting (10) and (11) into (12), we have

$$\dot{\underline{\mathbf{e}}} = \mathbf{A}\underline{\mathbf{e}} + \mathbf{B}[(\mathbf{W}^* - \mathbf{W})^T \Phi(\underline{\mathbf{z}}_d)] \quad (13)$$

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{I} \\ -\mathbf{G}(\underline{\mathbf{z}})\mathbf{K}_n & -\mathbf{G}(\underline{\mathbf{z}})\mathbf{K}_{n-1} & \dots & -\mathbf{G}(\underline{\mathbf{z}})\mathbf{K}_1 \end{bmatrix}$$

Equation (13) shows that only  $\mathbf{W}$  need to be further adjusted to minimize the tracking error. The adaptive law of  $\mathbf{W}$  is designed as follows:

$$\dot{\mathbf{w}}_i = \kappa \Phi \underline{\mathbf{e}}^T \mathbf{P} \mathbf{b}_i \quad i = 1 \dots m \quad (14)$$

where  $\mathbf{W} = [\mathbf{w}_1 \ \mathbf{w}_2 \dots \mathbf{w}_m]$ ,  $\mathbf{B} = [\mathbf{b}_1 \ \mathbf{b}_2 \dots \mathbf{b}_m]$ ,  $\kappa$  is a positive constant,  $m$  is the number of input variables of the nonlinear system or number of output variables of the G-FNN, i.e. each  $\mathbf{w}_i$  is a column vector associated with each output variable, and  $\mathbf{P}$  is a symmetric positive definite matrix that satisfies the following relationship:

$$\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P} = -\mathbf{Q} \quad (15)$$

where  $\mathbf{Q}$  is a symmetric 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. (3) (Wu *et al.*, 2001) 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:

$$\mathbf{\Gamma} = \{\|\mathbf{w}_i\| \leq \|\mathbf{w}_i(0)\|\} \quad i = 1 \dots m \quad (16)$$

where  $\|\cdot\|$  denotes two-norm of a vector. According to the projection algorithm (Wang, 1997), the adaptation law (14) can be modified as follows:

$$\dot{\mathbf{w}}_i = \begin{cases} \kappa \Phi \underline{\mathbf{e}}^T \mathbf{P} \mathbf{b}_i & \text{if } (\|\mathbf{w}_i\| < \|\mathbf{w}_i(0)\|) \\ \text{or} & \\ (\|\mathbf{w}_i\| = \|\mathbf{w}_i(0)\| \text{ and } \mathbf{w}_i^T \Phi \underline{\mathbf{e}}^T \mathbf{P} \mathbf{b}_i \leq 0) & (17) \\ \kappa \left( \mathbf{I} - \frac{\mathbf{w}_i \mathbf{w}_i^T}{\|\mathbf{w}_i\|^2} \right) \Phi \underline{\mathbf{e}}^T \mathbf{P} \mathbf{b}_i & \\ \text{if } (\|\mathbf{w}_i\| = \|\mathbf{w}_i(0)\| \text{ and } \mathbf{w}_i^T \Phi \underline{\mathbf{e}}^T \mathbf{P} \mathbf{b}_i > 0) & \end{cases}$$

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

*Theorem 4.1.* If the initial values of the weights  $\mathbf{w}_i(0) \in \mathbf{\Gamma}$ , the adaptation law (17) guarantees  $\mathbf{w}_i(t) \in \mathbf{\Gamma}$ ,  $\forall t > 0$ .

*Proof:* Consider the following Lyapunov function

$$V_b = \frac{1}{2} \mathbf{w}_i^T \mathbf{w}_i \quad (18)$$

Taking the derivative of the Lyapunov function with respect to time

$$\dot{V}_b = \mathbf{w}_i^T \dot{\mathbf{w}}_i \quad (19)$$

When  $(\|\mathbf{w}_i\| = \|\mathbf{w}_i(0)\| \text{ and } \mathbf{w}_i^T \Phi \underline{\mathbf{e}}^T \mathbf{P} \mathbf{b}_i \leq 0)$ ,  $\dot{V}_b \leq 0$ . Thus, it can be guaranteed that  $\|\mathbf{w}_i\| \leq \|\mathbf{w}_i(0)\|$ . When  $(\|\mathbf{w}_i\| = \|\mathbf{w}_i(0)\| \text{ and } \mathbf{w}_i^T \Phi \underline{\mathbf{e}}^T \mathbf{P} \mathbf{b}_i > 0)$ ,  $\dot{V}_b = 0$ . Thus,  $\|\mathbf{w}_i\| \leq \|\mathbf{w}_i(0)\|$  is also guaranteed. Since the initial value  $\|\mathbf{w}_i\| \leq \|\mathbf{w}_i(0)\|$  using the LLS method discussed in Appendix A.2,  $\mathbf{W}$  is bounded by the constraint set  $\mathbf{\Gamma}$  for all  $t \geq 0$ .  $\square$

#### 4.3 Stability Analysis of AFNC

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

*Theorem 4.2.* Consider the MIMO nonlinear dynamic system represented by (1). If the robust control law of (11) and the adaptive law of (17) are applied, asymptotic stability is guaranteed.

*Proof:* We consider the following Lyapunov function candidate, which is based on (13),

$$V(t) = \frac{1}{2} \underline{\mathbf{e}}^T \mathbf{P} \underline{\mathbf{e}} + \frac{1}{2} \kappa^{-1} \text{tr}[(\mathbf{W}^* - \mathbf{W})^T (\mathbf{W}^* - \mathbf{W})] \quad (20)$$

Taking the derivative of the Lyapunov function in (20) and using (13) and (15), we have

$$\begin{aligned} \dot{V}(t) &= \frac{1}{2} \dot{\underline{\mathbf{e}}}^T \mathbf{P} \underline{\mathbf{e}} + \frac{1}{2} \underline{\mathbf{e}}^T \mathbf{P} \dot{\underline{\mathbf{e}}} - \kappa^{-1} \text{tr}[(\mathbf{W}^* - \mathbf{W})^T \dot{\mathbf{W}}] \\ &= -\frac{1}{2} \underline{\mathbf{e}}^T \mathbf{Q} \underline{\mathbf{e}} + \underline{\mathbf{e}}^T \mathbf{P} \mathbf{B} (\mathbf{W}^* \\ &\quad - \mathbf{W})^T \Phi - \kappa^{-1} \text{tr}[(\mathbf{W}^* - \mathbf{W})^T \dot{\mathbf{W}}] \end{aligned} \quad (21)$$

Under condition 1 of (17), (21) becomes

$$\begin{aligned} \dot{V}(t) &= -\frac{1}{2} \underline{\mathbf{e}}^T \mathbf{Q} \underline{\mathbf{e}} + \underline{\mathbf{e}}^T \mathbf{P} \mathbf{B} (\mathbf{W}^* - \mathbf{W})^T \Phi \\ &\quad - \text{tr}[\underline{\mathbf{e}}^T \mathbf{P} \mathbf{B} (\mathbf{W}^* - \mathbf{W})^T \Phi] \\ &= -\frac{1}{2} \underline{\mathbf{e}}^T \mathbf{Q} \underline{\mathbf{e}} \leq 0 \end{aligned} \quad (22)$$

Under condition 2 of (17) and assuming that  $w_i^* \in \Gamma$ , (21) becomes

$$\begin{aligned} \dot{V}(t) = & -\frac{1}{2}\underline{\mathbf{e}}^T \mathbf{Q}\underline{\mathbf{e}} \\ & - \sum_{i=1}^m [(1 - \frac{(\mathbf{w}_i^*)^T \mathbf{w}_i}{\|\mathbf{w}_i\|^2}) \mathbf{w}_i^T \Phi \underline{\mathbf{e}}^T \mathbf{P} \mathbf{b}_i] \end{aligned} \quad (23)$$

Since  $\mathbf{w}_i^* \in \Gamma$  and  $1 - \frac{(\mathbf{w}_i^*)^T \mathbf{w}_i}{\|\mathbf{w}_i\|^2} \geq 0$

$$\dot{V}(t) \leq -\frac{1}{2}\underline{\mathbf{e}}^T \mathbf{Q}\underline{\mathbf{e}} \leq 0 \quad (24)$$

(20), (22) and (24) show that  $V(t) \geq 0$  and  $\dot{V}(t) \leq 0$ . Furthermore, (22) shows the  $\dot{V}(t) = 0$  if and only if  $\underline{\mathbf{e}} = 0$ , and (23) implies 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 (Slotine and Li, 1991), it can be shown that  $\underline{\mathbf{e}}(t) \rightarrow 0$  as  $t \rightarrow \infty$ . As a result, the control system is asymptotically stable. Moreover, the tracking error of the system will converge to zero.  $\square$

## 5. SIMULATION STUDIES

In this simulation example, we verify the validity of the proposed AFNC design on tracking control of a two-link robot manipulator depicted in Figure 3. The dynamic equation of the manipulator used in (Slotine and Li, 1991) is given by

$$\begin{aligned} \ddot{\mathbf{q}} = & -\mathbf{M}(\mathbf{q})^{-1} \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{M}(\mathbf{q})^{-1} \boldsymbol{\tau} \\ & - \mathbf{M}(\mathbf{q})^{-1} \boldsymbol{\tau}_d \end{aligned} \quad (25)$$

where

$$\begin{aligned} \mathbf{M}(\mathbf{q})_{11} &= a_1 + 2a_3 \cos q_2 + 2a_4 \sin q_2 \\ \mathbf{M}(\mathbf{q})_{12} &= a_2 + a_3 \cos q_2 + a_4 \sin q_2 \\ \mathbf{M}(\mathbf{q})_{21} &= a_2 + a_3 \cos q_2 + a_4 \sin q_2 \\ \mathbf{M}(\mathbf{q})_{22} &= a_2 \\ \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}})_1 &= -(a_3 \sin q_2 - a_4 \cos q_2) \dot{q}_1 \dot{q}_2 \\ & \quad - (a_3 \sin q_2 - a_4 \cos q_2) (\dot{q}_1 + \dot{q}_2) \dot{q}_2 \\ \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}})_2 &= (a_3 \sin q_2 - a_4 \cos q_2) \dot{q}_1^2 \\ a_1 &= I_1 + m_1 l_{c1}^2 + I_e + m_e l_{ce}^2 + m_e l_1^2 \\ a_2 &= I_e + m_e l_{ce}^2 \\ a_3 &= m_e l_1 l_{ce} \cos \delta_e \\ a_4 &= m_e l_1 l_{ce} \sin \delta_e \end{aligned}$$

where  $\mathbf{M}$  is the  $2 \times 2$  inertia matrix of the manipulator,  $\mathbf{Q}$  is the  $2 \times 1$  vector of centrifugal, Coriolis, friction forces and gravity,  $\boldsymbol{\tau}$  is the  $2 \times 1$  vector of input torque generated by the joint motor,  $\ddot{\mathbf{q}}$ ,  $\dot{\mathbf{q}}$  and  $\mathbf{q}$  are  $2 \times 1$  vectors of output link acceleration, velocity and position respectively, and  $\boldsymbol{\tau}_d$  is

the  $2 \times 1$  vector of unknown terms arising from unmodeled dynamics and external disturbance, .

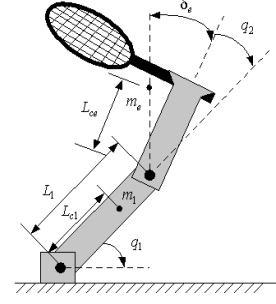


Fig. 3. Articulated Two-Link Robot Manipulator

Parameters of the two-link planar manipulator used for the simulation are: mass of link 1,  $m_1 = 1kg$ ; length of link 1,  $L_1 = 1m$ ; mass of link 2 and payload,  $m_e = 2kg$ ; angle of payload with respect to link 2,  $\delta_e = 30^\circ$ ; centroidal moment of inertia of link 1,  $I_1 = 0.12kgm^2$ ; length of CG of link 1 from the axis of rotation,  $L_{c1} = 0.5m$ ; centroidal moment of inertia of link 2 and payload,  $I_e = 0.25kgm^2$ ; length of CG of link 2 and payload from the axis of rotation,  $L_{ce} = 0.6m$ . Initial conditions are chosen as  $q_1 = q_2 = 0 rad$ ,  $\dot{q}_1 = \dot{q}_2 = 0 rad/sec$  and  $\ddot{q}_1 = \ddot{q}_2 = 0 rad/sec^2$ . The desired trajectory is an ellipse in  $q_1 - q_2$  plane not starting from the initial position with  $q_{d1} = 1.5 \sin(2\pi t) rad$  and  $q_{d2} = 0.7 \cos(2\pi t) rad$ . In a real system, there always exist uncertainties and disturbances. Hence, we deliberately introduce disturbances  $\tau_{d1} = 100 \sin(2\pi t) Nm$  and  $\tau_{d2} = 50 \sin(2\pi t) Nm$ , which are comparable to the control torques of the robot manipulator. The gains of the PD-like controller are selected as  $\mathbf{K}_2 = diag[25, 25]$  and  $\mathbf{K}_1 = diag[7, 7]$ .

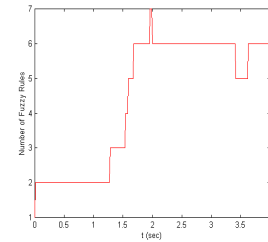


Fig. 4. Number of Fuzzy Rules Generated

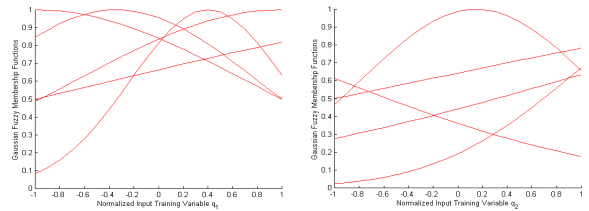


Fig. 5. Gaussian Fuzzy Membership Functions w.r.t Input Training Variable

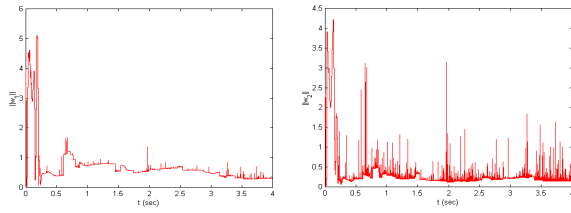


Fig. 6. Norm of the Weight Vectors  $\mathbf{w}_1$  and  $\mathbf{w}_2$  for AFNC

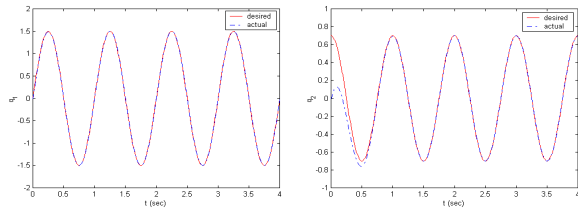


Fig. 7. Control Responses for the 2-Link Manipulator using AFNC

Using the G-FNN learning algorithm, the fuzzy neural structure and parameters are generated simultaneously and automatically. In this simulation, a total of six fuzzy rules are generated online as shown in Figure 4. The corresponding Gaussian fuzzy membership functions were obtained with respect to the input training variables as shown in Figure 5. 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. The weight vectors  $\|\mathbf{w}_1\|$  and  $\|\mathbf{w}_2\|$  are bounded throughout the control process as shown in Figure 6 even with online structure and parameter learning. Figure 7 shows control responses using AFNC. The results clearly demonstrate that the two links were able to track the desired trajectories from third trial onwards. It should be highlighted that for  $q_2$ , whose initial position was set to be 100% error from the desired initial link position, i.e.  $q_2(0) = 0$ , the link was still able to catch up with the desired trajectory very fast without any overshoot.

## 6. CONCLUSIONS

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

## REFERENCES

- Berenji, H. R. and P. Khedkar (1992). Learning and tuning fuzzy controllers through reinforcement. *IEEE Trans. on Neural Network* **3**, 724–739.
- Brown, M. and C. Harris (1994). *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall. London.
- Cannon, M. R. and J. J. E. Slotine (1995). Space-frequency localized basis function networks for nonlinear system estimation and control. *Neurocomputing*.
- Chang, Y. C. (2001). Adaptive fuzzy-based tracking control for nonlinear siso systems via VSS and  $H^\infty$  approaches. *IEEE Trans. Fuzzy Systems* **9**, 278–292.
- Gao, Y., M. J. Er and S. Yang (2001). Adaptive fuzzy neural control of robot manipulators. *IEEE Trans. on Industrial Electronics* **48**, 1274–1278.
- Jang, J. S. R., C. T. Sun and E. Mizutani (1997). *Neuro-Fuzzy and Soft Computing*. Prentice Hall. New Jersey.
- Jin, Y. (1998). Decentralized adaptive fuzzy control of robot manipulators. *IEEE Trans. on System, Man, and Cybernetics* **28**, 47–58.
- Jin, Y., J. Jiang and J. Zhu (1995). Neural network based fuzzy identification with application to modelling and control of complex systems. *IEEE Trans. on System, Man, and Cybernetics* **25**, 990–997.
- Jin, Y., W. V. Seelen and B. Sendhoff (1999). On generating  $fc^3$  fuzzy rule systems from data using evolution strategies. *IEEE Trans. on System, Man, and Cybernetics* **29**, 829–845.
- MacKay, D. J. C. (1995). Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research* **354**, 73–80.
- Nie, J. and D. Linkens (1995). *Fuzzy-Neural Control: Principles, Algorithms and Applications*. Prentice Hall. New Jersey.
- Park, D., A. Kandel and G. Langholz (1994). Genetic-based new fuzzy reasoning models with application to fuzzy control. *IEEE Trans. on System, Man, and Cybernetics* **24**, 39–47.
- Slotine, J. J. E. and W. Li (1991). *Applied Nonlinear Control*. Prentice Hall. New Jersey.
- Wang, L. X. (1997). *A Course in Fuzzy Systems and Control*. Prentice Hall. New Jersey.
- Wu, S. Q., M. J. Er and Y. Gao (2001). A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks. *IEEE Trans. on Fuzzy Systems* **9**, 578–594.