A Stable Neuro-Fuzzy Controller for Output Tracking in Composite Nonlinear Systems

Chih-Hsin Tsai* Jing-Sin Liu** Kuo-Bin Tseng** Wei-Song Lin*

*Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan 106, **China**

**Institute of Information Science
Academia Sinica
Taipei, Taiwan 115, **China**

Abstract. *In this paper, a learnable fuzzy controller is proposed for on-line implementing a decoupling control action of uncertain composite affine nonlinear plants to track a prescribed trajectory. The controller is mainly composed of decentralized fuzzy systems with embedded two-stages rule credit assignments cascaded with an interconnections compensating associative memory network. The controller can be parametrized by a set of linear parameters, which represent a combination of of the credits of rules, locations and shape factors of membership functions. The parameters are tuned by a deadzone adaptation algorithm. It is shown that the adaptive fuzzy controller guarantees a given level of attenuation for tracking error in the presence of unknown but bounded interconnections and disturbances. Simulation of an inverted pendulum is given to demonstrate the effectiveness and robustness of the controller.*

1 Introduction

In the development of control systems design, there is a major need to build the controllers which are capable of incorporating experts knowledge and containing enough intelligence to perform accuracy tasks in uncertain environments. This requires design of controllers whose architectures and consequent control efforts in response to plant outputs and external commands are related to or resulted from experience, that is, the observed input/output behavior of the plant, rather than by reference to a mathematical model-based description of the plant. The controller is then a so-called intelligent controller [1].

One emerging methodology in intelligent controllers design is the use of fuzzy logic [2], [3], mostly due to the fact that fuzzy methods provide an efficient way to cope with uncertainties and to encode and approximate numerical functions. This methodology has received more recognition recently and there have been a number of successful applications of fuzzy methods to a wide variety of practical problems.

However, the majority of fuzzy systems developed so far are static and are designed in an iterative open-loop fashion. Usually, the designer specifies a fuzzy rule base, and then enters an evaluation/editing design loop [4]. Both the performance measures and adaptation strategies are subjective.In addition, if the plant dynamics and the environment change, then the performance of well-designed fuzzy systems will degrade. Therefore, developing automatic learning algorithm is needed for on-line adjusting the rule bases of fuzzy systems in response to variations of operating conditions. On the other hand, in view of the promising capabilities of neural networks in learning, adaptation, fault tolerance, parallelism and generalization, efforts have been made to integrate the fuzzy logic and neural networks into a unified framework . The approach is that if prior knowledge in the form of fuzzy rules can be incorporated to develop a neural network in advance, then the initial performance of the network is improved and requires less training time.

Since the neural networks and fuzzy models are weighted superpositions of nonlinear functions, such as radial basis functions and fuzzy basis functions, they have been utilized to implement on-line approximation of the numerical functions describing the model of the plant dynamics . More recently, neural networks with radial basis functions and fuzzy system in combination with adaptive techniques are used to learn approximate feedback linearizing control action by on-line tuning the parameters involved [10], [11]. This paper studies the output tracking control problem in interconnected affine nonlinear systems. As regards the output tracking problem of uncertain nonlinear systems,there have been many designs of tracking controllers using the feedback linearization technique in nonlinear control theory [12]. To mention some among others, there are variable structure controller [13], robust controller [14] and adaptive controller [15]. Here, we present a neuro-fuzzy approach for synthesizing decoupling control law from sets of input/output membership functions. An adaptive fuzzy controller and its network structure, which is composed of decentralized fuzzy systems with embedded two-stages rule credit assignments cascaded with the interconnections compensating associative memory network , is proposed to realize a kind of decoupling control action for composite affine nonlinear systems.

In Section 2 , the output tracking problem for composite affine nonlinear systems is formulated. In Section 3, we present the concepts of approximate reasoning fuzzy system embedded with two-stages rule credit assignment. In Section 4, the components of the fuzzy controller together with its analytical form are given. In Section 5, a deadzone adaptation algorithm for controller parameters is derived to ensure robustness to approximation errors. In Section 6, simulation of the inverted pendulum is performed to illustrate the effectiveness and robustness of the controller. Finally, conclusions are made in Section 7.

2 The Output Tracking Problem

We begin our study by defining the class of plants under consideration. Consider a composite affine nonlinear system which is composed of n interconnected SISO affine nonlinear subsystems with each subsystem in a companion form:

$$y_i^{(p)} = f_i(\boldsymbol{x}, t) + \sum_{j=1}^{n} g_{ij}(\boldsymbol{x})u_j + v_i(\boldsymbol{x}, t) \tag{1}$$

$$\dot{z} = h(z, \boldsymbol{x}) \tag{2}$$

where $y_i \in R$, $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T$, $x_i = [y_i, \dot{y}_i, \cdots, y_i^{(p-1)}]^T$, $i = 1, \cdots, n$. z is a vector of appropriate dimension, f_i, g_{ij} are bounded nonlinear functions of the state x, v_i is unknown but bounded interconnection.

(1) can be rewritten compactly as

$$\boldsymbol{y}^{(p)} = \boldsymbol{f}(\boldsymbol{x}, t) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{v}(\boldsymbol{x}, t) \tag{3}$$

0-7803-4253-4/97/$10.00

where

$$
\begin{aligned}
y &= [y_1, \cdots, y_n]^T, \\
f(x,t) &= [f_1(x,t), \cdots, f_n(x,t)]^T, \\
G(x) &= \begin{bmatrix} g_{11}(x) & \cdots & g_{1n}(x) \\ \vdots & \ddots & \vdots \\ g_{n1}(x) & \cdots & g_{nn}(x) \end{bmatrix}, \\
u &= [u_1, \cdots, u_n]^T, \\
v(x,t) &= [v_1(x,t), \cdots, v_n(x,t)]^T,
\end{aligned} \tag{4}
$$

and u, x and z, y represent, respectively, the input, the observable and unobservable state, the output of the composite system (2)-(3). The matrix G is called the decoupling matrix while the dynamics (2) is called the unobservable dynamics of the system [12]. Throughout this paper, we assume the system (3) is decouplable, i.e. G is nonsingular; and the internal dynamics (2) is bounded-input bounded-state (BIBS) stable, i.e. for bounded x the dynamics (2) is bounded.

Given a desired trajectory $y_d(t) \in R^n$ and let $e_i = (y_i - y_{id}, \dot{y}_i - \dot{y}_{id}, \cdots, y_i^{(p-1)} - y_{id}^{(p-1)})^T$ and $e = (e_1, \cdots, e_n)^T$ denote the tracking error of the ith subsystem and the composite system. Then the tracking control problem for system (3) is to design a controller such that an acceptable tracking performance can be achieved (e.g., e is attenuated to a given level of accuracy) while stability is guaranteed. For the cases that the plant dynamics is completely known a priori, or bounds and properties of the functions $f_i(\cdot,t)$, $v_i(\cdot,t)$, and matrix $G(\cdot)$ are available, controllers such as PID, variable structure controller or adaptive and robust controllers could achieve satisfactory tracking performance. However, for the case we investigate here, the plant dynamics contains unknown interconnections between the subsystems, the tracking problem requires a controller with intelligence. In what follows our aim is thus to construct an adaptive fuzzy controller which could on-line learn the decoupling control for stable tracking in the composite affine nonlinear systems. Since the internal dynamics is assumed BIBS stable, it is omitted in the following.

3 Fuzzy System with Two-Stages Rule Credit Assignments

In general, a reasoning-based fuzzy system is composed of four principal components:the fuzzifier, the if-then rule base, the approximate reasoning engine, and the defuzzifier [16], [17]. By introducing a credit assignment mechanism in the rule bases, the approximate reasoning engine, as shown in Fig. 1, is called Fuzzy System with Rule Credit Assignment (*FS-RCA*). It processes the input knowledge by four stages: (i) *rule matching stage:* This stage computes the matching degrees (or firing strength[6]) between the current fuzzy input and the antecedent part of each rule. (ii) *fuzzy implication stage:* This stage determines the corresponding output action (recommendation) of each rule which was adjusted by the (stage I) *rule credit assignment,* and further (iii) modifies each recommendation by giving a credit in the (stage II) *rule credit assignment,* (iv)Finally, the system combines all the recommendations with different matching degrees into output fuzzy sets.

Let $s = (s_1, \cdots, s_n)^T$ represents the input (e or x) of the fuzzy system. The jth rule in the ith knowledge rule base, $R_{s,i}$ ($R_{e,i}$ or $R_{x,i}$), for the ith subsystem is defined by a set of linguistic rules of the following type:

$$
R^j_{s,i}: \quad IF \ s_1 \ is \ A^j_{i,1} \ AND \cdots AND \ s_n \ is \ A^j_{i,n} \quad THEN \quad u_i \ is \ B^j_i \tag{5}
$$

where $A^j_{i,k}$ is reference antecedent fuzzy set of s_k, and B^j_i is reference consequent fuzzy set of the outputs of the fuzzy system. This set of fuzzy if-then rules forms a control rule base whose antecedent parts are related to the measurement and whose consequent parts determine the control action. The quality of control action is inferred by a fuzzy inference engine and is evaluated by the credit assignments

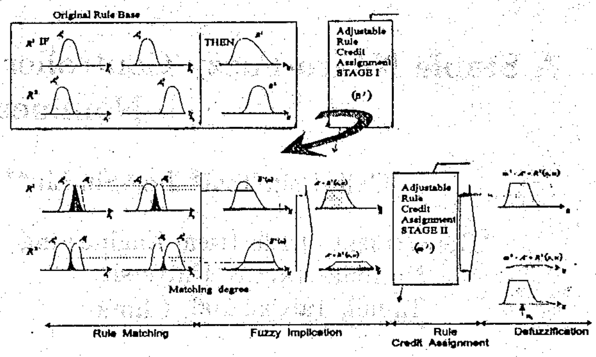

Figure 1: Diagrammatic representation of *FS-RCA*.

mechanism. The basic idea of rule credit assignment is to reward "good" rules by increasing their certainty of the consequent fuzzy set while punish "bad" ones by decreasing their certainty. There are two rule credit assignment stages presented in the fuzzy system of Fig. 1. First, at stage I, we reshape the consequent fuzzy set B^j_i of the original fuzzy rule base. This paper uses LR parametrization for the consequent membership functions. Thus after stage I rule credit assignment, the B^j_i membership function becomes another

$$
\tilde{B}^j_i(u_i) = \begin{cases} \left(1 + \left(\frac{c^j_{u,i} - u_i}{\beta^j_{ii} a^j_{L,i}}\right)^{b^j_{L,i}}\right)^{-1}, & if \ u_i \le c^j_{u,i} \\ \left(1 + \left(\frac{c^j_{u,i} - u_i}{\beta^j_{ii} a^j_{R,i}}\right)^{b^j_{R,i}}\right)^{-1}, & if \ u_i \ge c^j_{u,i} \end{cases} \tag{6}
$$

where $c^j_{u,i}, a^j_{R,i}, a^j_{L,i}$ are called, respectively, the center (where $\tilde{B}^j_i(u_i)$ achieves its maximum (one)), right and left spread of B^j_i membership function; β^j_{ii} is the credit. Note that reducing (or increasing) β^j_{ii} makes the definition of the linguistic term represented by $\tilde{B}^j_i(u_i)$ more precise (or broader).

By fuzzy implication inference, the corresponding output action (recommendation) of each rule is defined as

$$
A'_i \circ R^j_{u,i}(s, u_i) := Sup_{u \in U}[A^j_i(s) * I(A^j_i(s), \tilde{B}^j_i(u_i))] \tag{7}
$$

where U denotes the domain of input, $A'_i(s)$ is an arbitrary fuzzy set input to the fuzzy system,

$$
A^j_i(s) := A^j_{i,1}(s_1) * \cdots * A^j_{i,2n}(s_{2n}) \tag{8}
$$

denotes the matching degree, $*$ is the algebraic product, I the implication function and $\tilde{B}^j_i(u_i)$ is a reshaped $B^j_i(u_i)$ function of the original rule base as a consequence of stage I credit assignment. On the other hand, the stage II credit assignment is imposed on the fuzzy output where we have determined the corresponding output action of each rule. Here, we refine them by giving a credit, ω^j_{ii}, to the jth rule. Thus, upon imposing stage II credit assignment, the output fuzzy set becomes:

$$
\omega^j_{ii} \cdot A'_i \circ R^j_{u,i}(s, u_i) \tag{9}
$$

where "$\cdot$" is the multiplication operation.

Using the weighted center-average defuzzification method, the defuzzification of a single-input single-out fuzzy system can be easily extended to the case of multi-input multi-output fuzzy system embedded with credit assignments mechanism. As an extension, the defuzzification of multi-input multi-output fuzzy system is defined by

$$
u(t) = \begin{bmatrix} \Sigma_j \omega^j_{11} A^j_1 & \cdots & \Sigma_j \omega^j_{1n} A^j_1 \\ \vdots & & \vdots \\ \Sigma_j \omega^j_{n1} A^j_n & \cdots & \Sigma_j \omega^j_{nn} A^n_n \end{bmatrix}^{-1} \begin{bmatrix} \Sigma_j \omega^j_{11} A^j_1 \cdot \tilde{c}^j_{u,1} \\ \vdots \\ \Sigma_j \omega^j_{nn} A^j_n \cdot \tilde{c}^j_{u,n} \end{bmatrix}
$$

(10)

where $\tilde{c}^{j}_{u,i}$ denotes

$$\tilde{c}^{j}_{u,i} \equiv \textit{the centroid of the set}\{u : \tilde{B}^{j}_{i}(u) \geq A^{j}_{i}(s)\} \tag{11}$$

(using the local mean-of-maximum (LMOM) implication method [18]), and ω^{j}_{ii} represents the credit assigned to $R^{j}_{s,i}$ for subsystem i, while ω^{j}_{ik}, $i \neq k$, is used to counteract the dynamic interactions between the subsystems i and k.

4 The Tracking Controller with Layered Network Structure

Combining the approximate reasoning engine described in Section 3 with layered structure, we construct a fuzzy controller that could on-line implement decoupling control law for composite nonlinear systems (3). As shown in Fig.2, the proposed controller is an intergration of (a)decentralized FS-RCAs, (b)an interconnections compensating associative memory network for counteracting the unknown interconnections among the subsystems and (c)a nonsingularity supervisor for monitoring the feasibility of cascading the components (a) and (b). The aim of this fuzzy system is to on-line compute an approximately decoupling control action to achieve nearly decoupled trajectory tracking for each subsystem.

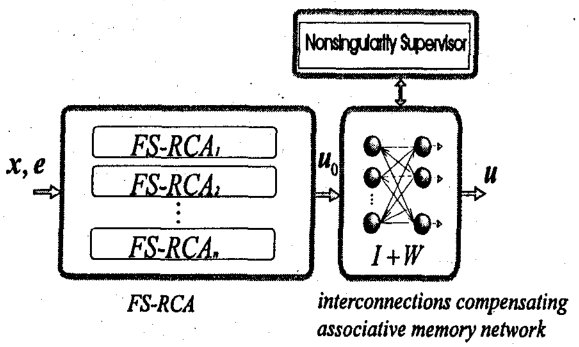

Figure 2: Components of the proposed controller

- *The decentralized FS-RCAs*
  In view of the defuzzification formula (10), the defuzzification of the decentralized FS-RCAs can be defined as

$$u_0(t) = \hat{D}^{-1}(x, \Theta^{(\omega)}) \left(\hat{f}(x, \Theta^{(ca)}) + r(e,t)\right) \tag{12}$$

where

$$\hat{D}(x, \Theta^{(\omega)}) = diag(\omega^{T}_{11}\hat{g}_{\omega 1}(x) + 1, \ldots, \omega^{T}_{nn}\hat{g}_{\omega n}(x) + 1)$$

$$\hat{f} = \begin{bmatrix} -\theta^{(ca)^T}_{1} \hat{f}_{\theta 1}(x) \\ \vdots \\ -\theta^{(ca)^T}_{n} \hat{f}_{\theta n}(x) \end{bmatrix}, \quad r = \begin{bmatrix} y^{(p)}_{1d} - \alpha^{T}_{1} e_1 \\ \vdots \\ y^{(p)}_{nd} - \alpha^{T}_{n} e_n \end{bmatrix}, \tag{13}$$

with $\Theta^{(\omega)} = (\theta^{(\omega)}_{1}, \cdots, \theta^{(\omega)}_{n})^T$, $\theta^{(\omega)}_{i} = (\omega_{i1}, \cdots, \omega_{in})^T$, $\omega_{ir} = (\omega^{1}_{ij}, \cdots, \omega^{m}_{ij})^T$, $\hat{g}_{\omega i} = (\hat{g}^{1}_{\omega i}, \cdots, \hat{g}^{m}_{\omega i})^T$, $\Theta^{(ca)} = (\theta^{(ca)}_{1}, \cdots, \theta^{(ca)}_{n})^T$, $\theta^{(ca)}_{i} = (\omega^{1}_{ii}c^{1}_{u}, \cdots, \omega^{m}_{ii}c^{m}_{u}, a^{1}_{LR,i}, \cdots, a^{m}_{LR,i})^T$, $\hat{f}_{\theta i} = (-A^{1}_{i}, \cdots, -A^{m}_{i}, \hat{f}^{1}_{LR,i}, \cdots, \hat{f}^{m}_{LR,i})^T$, $\hat{f}^{j}_{LR,i} = A^{j}_{i}\sqrt{(A^{j}_{i})^{-1} - 1}$.

- *The Interconnections Compensating Associative Memory Network*
  To compensate the unknown interconnections among the subsystems and disturbances acting on each subsystem to achieve decoupled tracking behavior, the interconnections compensating associative memory network is cascaded with decentralized FS-RCAs. Basically, the interconnections compensating associative memory network recombines the output of the decentralized FS-RCAs, u_0, into a new vector u, the control action, by the operator M defined as

$$u(t) = M(u_0) = (I_n + W)(u_0) \tag{14}$$

$$W = -(I_n + \hat{C}^{-1}\hat{D})^{-1} \tag{15}$$

where

$$\hat{C}(x, \Theta^{(\omega)}) = \begin{bmatrix} 0 & \omega^{T}_{12}\hat{g}_{\omega 1}(x) & \cdots & \omega^{T}_{1n}\hat{g}_{\omega 1}(x) \\ \omega^{T}_{21}\hat{g}_{\omega 2}(x) & 0 & \cdots & \omega^{T}_{2n}\hat{g}_{\omega 2}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{T}_{n1}\hat{g}_{\omega n}(x) & \omega^{T}_{n2}\hat{g}_{\omega n}(x) & \cdots & 0 \end{bmatrix} \tag{16}$$

- *The Nonsingularity Supervisor*
  Since the weight matrix W in (15) will likely be singular, the nonsingularity supervisor is used to monitor the feasibility of cascading the decentralized fuzzy systems and the interconnections compensating network. This is done via the function of nonsingularity supervisor by slightly perturbing $\hat{C}$ to another nonsingular $\hat{C}$ during the whole control process.

Using (12), (14), (15) and applying the Matrix Inversion Lemma [19] $(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1}$, the defuzzified output of the fuzzy controller resolves into

$$u(t) = \left(I_n - (I_n + \hat{C}^{-1}\hat{D})^{-1}\right)\hat{D}^{-1}\left(\hat{f} + r\right) \tag{17}$$

$$= \hat{G}^{-1}\left(\hat{f}(x, \Theta^{(ca)}) + r(e,t)\right) \tag{18}$$

where $\hat{G} = \hat{C} + \hat{D}$. The invertibility of where $\hat{G} = \hat{C} + \hat{D}$. The invertibility of $\hat{G}$ can be guaranteed by proper choices of controller parameters (see Section 5).

The fuzzy control processing can be adapted to a parallel neural network structure where each node contains the knowlege of fuzzy membership functions and each connection represents the credit of a fuzzy rule. With the network structure, the fuzzy controller has a total of four layers:

- *Layer 1:* Each node denotes the input e or x to the fuzzy system.
- *Layer 2:* Each node calculates the rule matching degree $\hat{g}^{j}_{i}(s) = A^{j}_{i}(s)$.
- *Layer 3:* Each node in this layer obtains the singleton implication fuzzy set and computes its location $\tilde{c}^{j}_{u,i}$ by (11).
- *Layer 4:* This layer contains n nodes, which calculate the decoupling control according to (18).

Each layer corresponds to a sub-stage of the approximate reasoning fuzzy system with adjustable rule credit assignment shown in Fig. 1. This structure allows the input be fuzzified/defuzzified in a parallel way by simultaneously matching membership functions encoded in the nodes.

5 Tracking Performance

In this section, we investigate the closed-loop system (Fig. 3). It is assumed that, given any uniform bounds ϵ_f, ϵ_g, there exist parameter vectors $\theta^{*}_{1}, \cdots, \theta^{*}_{n}$ such that the network approximation errors satisfy

$$\begin{aligned} \max_{x} \left\| f(x,t) - \hat{f}(x, \Theta^{(ca)*}) \right\| &\leq \epsilon_f, \\ \max_{x} \left\| G(x) - \hat{G}(x, \Theta^{(\omega)*}) \right\| &\leq \epsilon_g. \end{aligned} \tag{19}$$

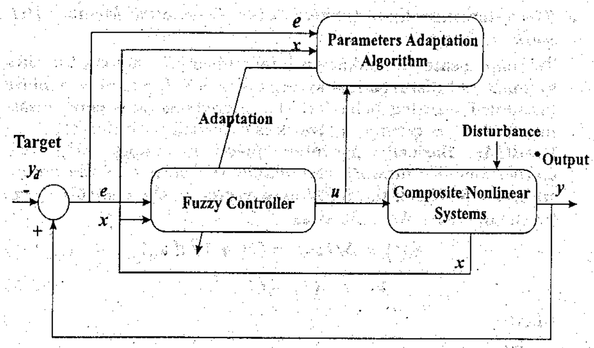

Figure 3: The closed-loop adaptive fuzzy control system

However, the parameter vectors which ensure the above approximation accuracy are unknown and must be on-line estimated. Consider the parameters update algorithm that works for each subsystem in parallel

$$\Delta\theta_i = R_i^{-1} w_i b_i^T P_i e_i \tag{20}$$

where $P_i = P_i^T > 0$ is the solution of the following Riccati-like equation

$$P_i A_i + A_i^T P_i + \frac{1}{\rho^2} P_i b_i b_i^T P_i + Q_i = 0 \tag{21}$$

with $\rho > 0, Q_i > 0$ and

$$R_i = Block\ diag\ (R_i^{(c)}, R_i^{(a)}, R_i^{(\omega)}),$$

$$R_i^{(\omega)} = Block\ diag\ (R_i^{(1)}, \cdots, R_i^{(n)}),$$

where $R_i^{(c)}, R_i^{(a)}, R_i^{(1)}, \cdots, R_i^{(n)} > 0$.

A continuous version of (20) could be expressed as

$$\dot{\theta}_i = R_i^{-1} w_i b_i^T P_i e_i \tag{22}$$

Unboundedness of vector $\theta_i(t)$ due to the presence of disturbance (called *parameter drift* in adaptive control) could usually occur when using the adaptation algorithm (22). However, it is noted that the parameter drift phenomenon can be avoided by suitably modifying the adaptation algorithm using the deadzone technique so that the adaptation could be turned off whenever the tracking error is smaller than a threshold. The incorporation of dead zones in parameter tuning algorithm guarantees the boundedness of approximation errors of the nonlinear matrix functions involved in decoupling control to be approximated. Thus, a deadzone of size d_0 is employed in the adaptation algorithm to achieve stopping the adaptation if necessary and to counteract the modeling error and the parameter estimation errors. Now, we present a deadzone modification of the adaptation algorithm (22). Suppose the parameter $\theta_i(t)$ is required to be inside a bounding set M_i during adaptation. Let $\theta_{i\perp} = \theta_i / \|\theta_i\|$. Define $P = Block\ diag\ (P_1, \cdots, P_n)$, and $b = Block\ diag\ (b_1, \cdots, b_n)$. Then the modified deadzone adaptation algorithm is :

$$\begin{aligned} \dot{\theta}_i &= 0, \qquad\qquad if\ e^T Pbb^T Pe \le d_0^2 \\ &= (I - d_{M_i}(\theta_i)\theta_{i\perp}\theta_{i\perp}^T) R_i^{-1} w_i b_i^T P_i e_i, \\ &\qquad\qquad otherwise \end{aligned} \tag{23}$$

where we define the distance measure between a set and a vector

$$d_{M_i}(\theta_i) := \left\{ \begin{array}{ll} 0 & if\ \theta_i^T (R_i^{-1} w_i b_i^T P_i e_i) \le 0 \\ min\,[1, dist(\theta_i, M_i)/\varepsilon^*], \varepsilon^* > 0 & otherwise \end{array} \right. \tag{24}$$

and $dist(\cdot,\cdot)$ denotes the distance between the arguments.

Let the parameters estimation error be defined as $\tilde{\theta}_i = \theta_i - \theta_i^*$ where θ_i represents actual parameter used in the controller and is tuned by deadzone adaptation algorithm. In terms of $\tilde{\theta}_i$ the error equation can be rewritten as

$$\dot{e}_i = A_i e_i - b_i w_i^T \tilde{\theta}_i + b_i \xi_i \tag{25}$$

where

$$\begin{aligned} \xi_i &= \left(f_i(x,t) - \theta_i^{(ca)*T} \hat{f}_{\theta i} \right) \\ &+ \sum_{j=1}^{n} \left(g_{ij}(x) - \omega_{ij}^{*T} \hat{g}_{\omega i}(x) - 1 \right) u_j(t) + v_i \end{aligned} \tag{26}$$

represents the lumped disturbance term of i-th subsystem due to the network approximation error and external disturbance. In the case there exist $\bar{\epsilon}_g$ and $\bar{\delta}_g$ small enough such that $\epsilon_g \le \bar{\epsilon}_g$ and $\|\tilde{\Theta}^{(\omega)}\| \le \bar{\delta}_g$, we can show that $\hat{G}^{-1}(x, \Theta^{(\omega)})$ exists, which in turn, guarantees the feasibility of the controller (18).

The following theorem shows the performance of the neuro-fuzzy controller with the deadzone parameters adaptation algorithm.

Theorem 1 Consider the composite nonlinear systems (3) with unknown but bounded $f_i(x,t)$, $v_i(x,t)$, $i = 1, \cdots, n$ and nonsingular matrix G. Assume that the controller (18) is adopted with the deadzone adaptation algorithm . Then in the bounded state space $x \in \Omega = \{x : \|x\|\ is\ bounded\}$, we have θ_i and the control input $u(t)$ are bounded. Let $\xi = (\xi_1(x,t), \cdots, \xi_n(x,t))^T$, $Q = Block\ diag(Q_1, \cdots, Q_n)$. Assume that there exists $\bar{\xi} = Sup_{x,t} \|\xi_i(x,t)\|^2$, then e converges to the residual set $\{e : e^T Q e \le \rho^2 \bar{\xi}\ or\ e^T Pbb^T Pe \le d_0^2\}$. Moreover, for the case that ϵ_f and ϵ_g are small enough such that $\|\xi\| \le \frac{1}{2\rho^2} d_0$, then e converges to the deadzone $\{e : e^T Pbb^T Pe \le d_0^2\}$.

Proof. Omitted (cf. [20]). Q.E.D.

6 Simulation

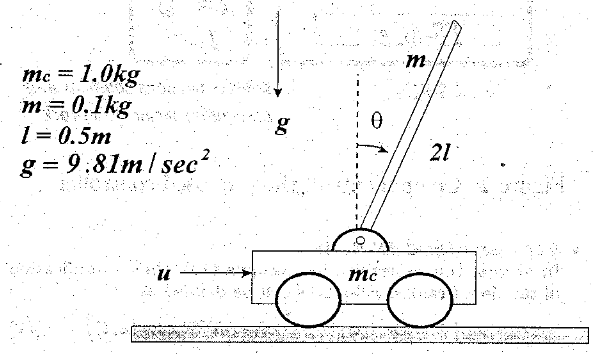

Figure 4: The inverted pendulum system.

Consider the inverted pendulum depicted in Fig.4. Suppose the movement of both the pole and the cart is restricted to the vertical plane and the cart is allowed to move infinitely in the left or right direction. The state of the system is described by the pole's angle, θ, and its angular velocity, $\dot{\theta}$. Its state equation can be expressed as:

$$\dot{x}_1 = x_2,$$

$$\dot{x}_2 = \frac{g sin x_1 - \frac{m l x_2^2 cos x_1 sin x_1}{m_c + m}}{l(\frac{4}{3} - \frac{m cos^2 x_1}{m_c + m})} + \frac{\frac{cos x_1}{m_c + m}}{l(\frac{4}{3} - \frac{m cos^2 x_1}{m_c + m})} u + v,$$

$$y = x_1,$$

where $x_1 = \theta$, $x_2 = \dot{\theta}$, g is the acceleration due to gravity, m_c is the mass of the cart, m is the mass of the pole, l is the half-length of the pole, u is the applied force and v is the external disturbance.

This system is unstable if the control u is set to be 0.

Simulations are performed for disturbance of size $|v(t)| \leq 0.5$. Parameters of the neuro-fuzzy controller are set as $\rho = 0.02$, $Q = 10I_{2\times2}$, $R = Block\ diag[0.02I_{25\times25},\ 0.01I_{25\times25},\ 1.0I_{25\times25}]$; $\epsilon* = 0.05$, $M_i = \{\theta : |c_u^j| \leq 15, |a_{LR}^j| \leq 6, |\omega^j| \leq 2\}$ and the PD gain $\alpha_1 = 10$, $\alpha_2 = 100$. Initially, c_{u0}^j are chosen randomly in the interval $[-12, 12]$, and $a_{L0}^j = a_{R0}^j = 2$, $\omega_0^j = 1$. Fig. 5 shows the membership functions of x_1 , x_2 and u used in simulation. Fig. 6 shows the simulation result. Simulations show that the neuro-fuzzy controller achieves quite satisfactory tracking performance.

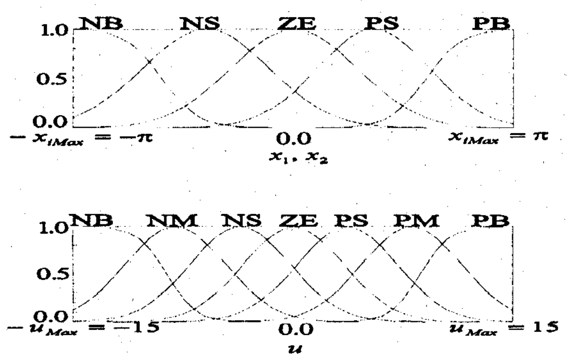

Figure 5: Membership functions for x_1,x_2 and u. N represents negative, P positive, ZE approximately zero, S small, M medium, B big.

7 Conclusion

For composite affine nonlinear systems, this paper proposes a learnable tracking controller composed of decentralized approximate reasoning fuzzy system with adjustable rule credit assignment cascaded with an interconnections compensating associative memory network and a nonsingularity supervisor. The fuzzy controller can be naturally mapped into a four-layer network structure. The weights of network, which represents a combination of the credits of rules, shapes and locations of membership functions, are tuned via a deadzone adaptation algorithm. The controller achieves on-line computation of decoupling control which guarantees the boundedness of network weights and stability and the attenuation of tracking error to a prescribed level.

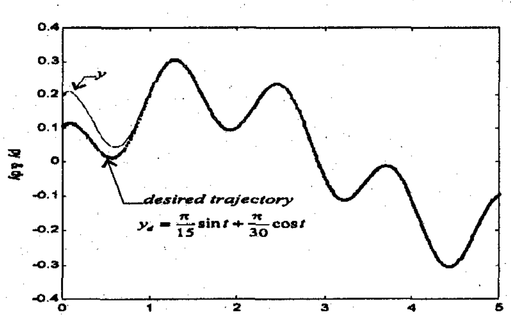

Figure 6: Simulation result of the inverted pendulum.

References

[1] Y.-H. Pao, S. M. Phillips, and S. Dejan J, "Neural-net computing and the intelligent control of systems," *Int. J. Control*, vol. 31, pp. 127–134, 1986.

[2] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 3, no. 1, pp. 28–44, 1973.

[3] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *IEE Proceeding*, vol. 121, no. 12, pp. 1585–1588, 1974.

[4] M. Brown, P. E. An, and C. J. Harris, "On the condition of adaptive neurofuzzy models," in *Proc. Int. Conference on Fuzzy Systems*, vol. 2, pp. 663–670, 1995.

[5] S. I. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE Trans. Neural Network*, vol. 3, no. 5, pp. 801–806, 1992.

[6] J.-S. R. Jang, "Self-learning fuzzy controllers based on temporal backpropagation," *IEEE Trans. Neural Network*, vol. 3, no. 5, pp. 714–723, 1992.

[7] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence.* Englewood Cliffs, NJ: Prentice-Hall, 1992.

[8] H. Ishibuchi, R. Fujioka, and H. Tanaka, "Neural networks that learn from fuzzy if-then rules," *IEEE Trans. Fuzzy Systems*, vol. 1, no. 2, pp. 85–97, 1993.

[9] C. Lin and C. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Computers*, vol. 40, no. 12, pp. 1320–1336, 1991.

[10] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis.* NJ: Prentice-Hall, 1994.

[11] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Network*, vol. 3, no. 6, pp. 837–863, 1992.

[12] A. Isidori, *Nonlinear Control Systems: An Introduction.* Berlin: Springer-Verlag, 1989.

[13] B. F. R. and J. Hedrick, "Control of multivariable nonlinear system by sliding mode method," *Int. J. Control*, vol. 46, no. 3, pp. 1019–1040, 1987.

[14] S. Behtash, "Robust output tracking for nonlinear systems," *Int. J. Control*, vol. 51, no. 6, pp. 1381–1407, 1990.

[15] S. Sastry and A. Isidori, "Adaptive control of linearizable systems," *IEEE Trans. Automatic Control*, vol. 34, no. 11, pp. 1123–1131, 1989.

[16] D. A. Linkens and J. Nie, "A unified real time approximate reasoning approach for use in intelligent control. part 1: Theoretical development," *Int. J. Control*, vol. 56, pp. 347–364, 1992.

[17] D. A. Linkens and J. Nie, "Neural network-based approximate reasoning: Principles and implementation," *Int. J. Control*, vol. 56, pp. 399–414, 1992.

[18] H. R. Berenji and P. S. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. Neural Network*, vol. 3, no. 5, pp. 724–740, 1992.

[19] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge, U.K.: Cambridge University Press, 1985.

[20] C.-H. Tsai, J.-S. Liu, and W.-S. Lin, "A neuro-fuzzy logic controller for trajectory tracking of uncertain robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, pp. 1929–1934, 1996.