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, R.O.C.

‡ Institute of Information Science
Academia Sinica
Nankang, Taipei, Taiwan 115, R.O.C.
email: liu@iis.sinica.edu.tw
fax: 886-2-27824814

*Research partially supported by National Science Council of R.O.C. under contracts NSC 83-0422-E-001-001, NSC 83-0422-E-001-008 and NSC 84-2212-E-001-003. Parts of this work have been presented in *1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, USA, 1996, and First IEEE International Conference on Intelligent Processing Systems, Beijing, China, 1997.*

Abstract

In this paper, a learnable neuro-fuzzy controller is proposed for on-line implementing a decoupling control action for uncertain composite affine nonlinear plants to track a prescribed trajectory. In structure, the controller is composed of decentralized fuzzy systems with embedded two-stages rule credit assignment mechanism cascaded with an interconnections compensating associative memory network and a nonsingularity supervisor. In analytical form, 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 to compensate for uncertainties. It is shown that the incorporation of deadzone in controller guarantees the stability of adaptation in the neuro-fuzzy system and moreover, a given level of attenuation for tracking error in the presence of unknown but bounded interconnections and disturbances. Simulation results of SISO plant, an inverted pendulum, and MIMO plant, a two-link planar robot manipulator, are given to demonstrate the effectiveness and robustness of the neuro-fuzzy controller for output tracking in composite nonlinear systems.

Keywords: neuro-fuzzy systems, tracking, nonlinear systems

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. For example, industrial process control [4], [5], robot control [6] and automobile transmission control [7].

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 [8]. 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 of neuro-fuzzy systems [8] - [15]. By combining the advantages of neural networks and fuzzy systems, neuro-fuzzy systems can be more effective in applications, when compared to either neural networks or fuzzy systems. Some researches have focused on implementing fuzzy systems on a neural network architecture. The layers of the neural networks perform the fuzzification and defuzzification on crisp input/output data and other functions of fuzzy systems. For example, neural-network-based fuzzy logic control and decision systems [13], adaptive-network-based fuzzy inference systems [10], adaptive fuzzy systems [14], compensatory neurofuzzy systems with adaptive fuzzy reasoning [15]. 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 applied to systems control to implement on-line approximation of the numerical functions describing the model of the plant dynamics or the controller [14], [16] -[18].

Though the integrations of neural networks and fuzzy logic have helped in accessing and exploiting better their respective advantages, there are concerns about the stability and performance analysis of the fuzzy neural or neuro-fuzzy systems. If the mathematical model of the plant is known, the stability of fuzzy control systems can be analyzed by using, e.g. the "Circle Criterion" [19], the "Energetistic Stability" [20], the "Expert Lyapunov Function" [21], and the "Nyquist Stability" [22]. On the other hand, if the plant model structure is known, then a Lyapunov stability analysis of the system can be done [14], [16], [17]. For example, [14], [17] applied fuzzy methods where fuzzy rules can be incorporated and extracted in the adaptive fuzzy controller with guaranteed stability.

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 in the literature using basically the feedback linearization technique in nonlinear control theory [23]. To mention some among others, there are , robust controller [24], variable structure controller [25] and adaptive controller [26]. More recently, the neural networks with radial basis functions is combined with adaptive techniques is used to learn approximate feedback linearizing control action by on-line tuning the parameters of neural networks [16], [27], [28]. Here, we present a neuro-fuzzy approach for synthesizing decoupling control law from sets of input/output membership functions. An adaptive fuzzy controller composed of decentralized fuzzy systems with embedded two-stages rule credit assignment mechanism, cascaded with the interconnections compensating associative memory network, along with its network structure is proposed to realize a kind of decoupling control action for achieving output tracking in composite affine nonlinear systems.

In Section 2, the output tracking problem for composite affine nonlinear systems is formulated. In Section 3, the approximate reasoning fuzzy system embedded with two-stages rule credit assignment mechanism is presented. In Section 4, the components of the fuzzy controller together with its analytical form and its mapping to a four-layer network structure are given. In Section 5, a deadzone adaptation algorithm for controller parameters is derived to ensure robustness to approximation errors. The stability and tracking performance of the fuzzy system tuned by deadzone adaptation algorithm is analyzed in Section 6. In Section 7, the simulations of the inverted pendulum and a two-link robot carrying a heavy load are performed to illustrate the effectiveness and robustness of the controller in output tracking problem. Finally, conclusions are made in Section 8.

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

(1) can be rewritten compactly as

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

where

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

and $\boldsymbol{u}$, $\boldsymbol{x}$ and z, $\boldsymbol{y}$ represent, respectively, the input, the observable and unobservable state, the output of the composite system (2)-(3). The matrix $\boldsymbol{G}$ is called the decoupling matrix while the dynamics (2) is called the unobservable dynamics of the system [23]. Throughout this paper, we assume the system (3) is decouplable, i.e. $\boldsymbol{G}$ is nonsingular; and the internal dynamics (2) is bounded-input bounded-state (BIBS) stable, i.e. for bounded $\boldsymbol{x}$ the dynamics (2) is bounded.

Given a desired trajectory $\boldsymbol{y}_d(t) \in R^n$ 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 $\boldsymbol{e} = (e_1, \cdots, e_n)^T$ 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., $\boldsymbol{e}$ 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 $\boldsymbol{G}(\cdot)$ are available, the controllers such as PID, variable structure controller or adaptive and robust controllers have beem developed and could achieve satisfactory tracking performance. However, for the class of plants we investigate here, the subsystem dynamics and the interconnections among the subsystems are unknown, the tracking problem requires a controller with learning capabilities.

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

As a preliminary, this section introduces basic concepts of fuzzy systems. 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 [29], [30]. In this paper, we modify the approximate reasoning engine by rewarding or punishing the rules using the technique of credit assignments. The resulting system is designated as by *Fuzzy System with adjustable Rule Credit Assignment* (*FS-RCA*). The configuration of the FS-RCA is shown in Fig. 1, with more details of its diagrammatic representation shown in Fig. 2.

The approximate reasoning engine, as shown in Fig. 2, processes the input knowledge according to the following four stages: (i) *rule matching stage:* This stage computes the matching degrees (or firing strength[10]) 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 $\boldsymbol{s} = (s_1, \cdots, s_n)^T$ represents the input ($\boldsymbol{e}$ or $\boldsymbol{x}$) of the fuzzy system. The jth rule in the ith knowledge rule base, $\boldsymbol{R}_{\boldsymbol{s},i}$ ($\boldsymbol{R}_{\boldsymbol{e},i}$ or $\boldsymbol{R}_{\boldsymbol{x},i}$), for the ith subsystem is defined by a set of linguistic rules

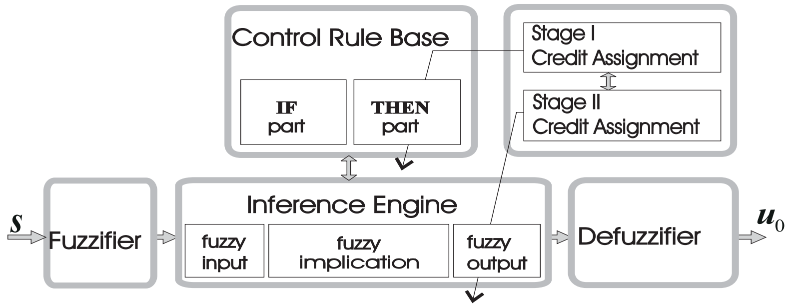

Figure 1: Fuzzy System with Two-Stages Rule Credit Assignment

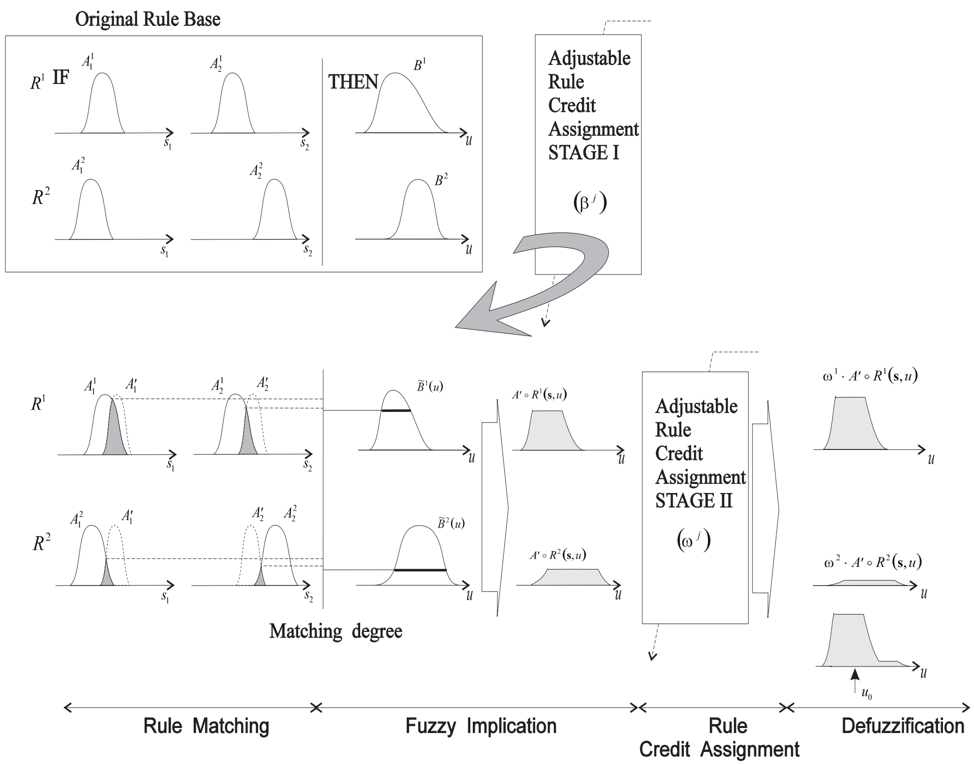

Figure 2: Diagrammatic representation of *FS-RCA*

of the following type:

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

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 mechanism, as shown in Fig. 1.

For the simplicity of presentation, we consider an FS-RCA with single input only. The extension to multi-input case is presented in next section. Now, we describe each phase of the fuzzy information processing in Fig. 2.

A. Rule Matching and Fuzzy Implication Phases

There are many different kinds of fuzzy logic which may be used in a fuzzy inference machine [14], [31] – [33]. Let $\boldsymbol{A}'(\boldsymbol{s})$ denote an arbitrary fuzzy set input to the fuzzy system. By fuzzy implication inference, the corresponding output action (recommendation) of each rule $R^j_{\boldsymbol{S}}$ determines a fuzzy set in the output space U based on the generalized modus ponens [34], [35]:

$$
\begin{array}{ll}
\textit{jth rule, } R^j_{\boldsymbol{S}}: & IF\ \ s_1\ is\ A^j_1\ AND \cdots AND\ s_n\ is\ A^j_n \\
 & \qquad THEN\quad u\ is\ B^j \\
\textit{Input of FS-RCA:} & \boldsymbol{s}\ \ is\ \boldsymbol{A}' \\
\hline
\textit{Corresponding recommendation:} & u\ is\ \boldsymbol{A}' \circ R^j_{\boldsymbol{S}}
\end{array}
$$

$$
\boldsymbol{A}'(\boldsymbol{s}) \circ R^j_{\boldsymbol{S}}(\boldsymbol{s}, u) = \operatorname*{Sup}_{\boldsymbol{u} \in U} \left[\boldsymbol{A}'(\boldsymbol{s}) * I(\boldsymbol{A}^j(\boldsymbol{s}), \tilde{B}^j(u)) \right], \tag{6}
$$

where $\boldsymbol{A}^j(\boldsymbol{s}) = A^j_1(s_1) * \cdots * A^j_n(s_n)$ denotes the matching degree, $*$ is the T-norm [14]. I the implication function (such as min-operation, product-operation, arithmetic [14], [32], mean-of-maximum [33], local mean-of-maximum implications [31]). Note that $\tilde{B}^j(u)$ denotes a reshaped consequent fuzzy set of $B^j(u)$ in the original rule base (5) after stage I rule credit assignment.

B. Two-Stages Credit Assignment Phase

However, there are situations that inappropriate rules get high matching degrees in the antecedent part, and the the performance of fuzzy system will degrade. To overcome the problem, the stage II rule credit assignment is introduced at the output fuzzy sets to enhance the performance of the approximate reasoning engine.

In summary, there are two rule credit assignment stages introduced in the fuzzy system. At stage I, we reshape the consequent fuzzy sets in the original if-then rule base. Basically, the use of credit assignment in stage I is to reward (or punish) a rule by increasing (or decreasing) the certainty of its consequent fuzzy set before using the rule base. The concept of credit assignment is shown in Fig. 3.

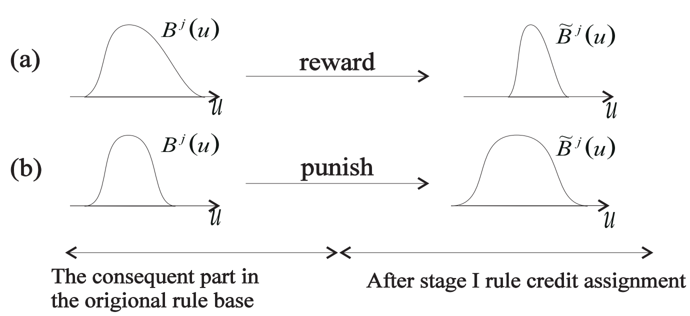

Figure 3: The stage I rule credit assignment: (a) reward and (b) punish rules

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 , to the jth rule, then the output fuzzy set becomes:

$$
\omega^j \cdot \boldsymbol{A}'(\boldsymbol{s}) \circ R^j_{\boldsymbol{S}}(\boldsymbol{s}, u) \tag{7}
$$

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

C. Defuzzification Phase

The crisp output u_0 generated by the fuzzy system is obtained by defuzzification of output fuzzy set, which is obtained from the combination of the recommendation of each rule. Define the overall output fuzzy set F inferred by the fuzzy system as follows:

$$F(u) = \bigcup_j \omega^j \cdot \boldsymbol{A}'(\boldsymbol{s}) \circ R^j_{\boldsymbol{s}}(\boldsymbol{s}, u) \tag{8}$$

where $\bigcup$ denotes the T-conorm [36]. Some of the commonly used defuzzification methods are defined in terms of $F(u)$:

a) *Center of Area:*

$$u_0^{COA} = defuzz(F(u)) = \frac{\int F(u)udu}{\int F(u)du} \tag{9}$$

b) *Center Average* [14]:

$$u_0^{CA} = \frac{\sum_j \omega^j \cdot \boldsymbol{A}'(\boldsymbol{s}) \circ R^j_{\boldsymbol{s}}(\boldsymbol{s}, \bar{u}^j) \cdot \bar{u}^j}{\sum_j \omega^j \cdot \boldsymbol{A}'(\boldsymbol{s}) \circ R^j_{\boldsymbol{s}}(\boldsymbol{s}, \bar{u}^j)} \tag{10}$$

where $\bar{u}^j$ is the point in U at which $\boldsymbol{A}'(\boldsymbol{s}) \circ R^j_{\boldsymbol{s}}(\boldsymbol{s}, u)$ achieves its maximum value.

c) *Max-Criteria:*

$$u_0^{MC} = u, \qquad such\ that\ F(u)\ is\ the\ maximum. \tag{11}$$

4 The Fuzzy Controller

By combining the approximate reasoning engine described in Section 3, a fuzzy controller that could on-line implement decoupling control law for composite nonlinear systems (3) is constructed. As will become clear later, the construction of the fuzzy controller is a deterministic nonlinear system, thus allowing the analysis to be carried out for performance evaluation. This is in contrast to the constructions of [37] and [38] whose linguistic constructs prevent an analytic evaluation.

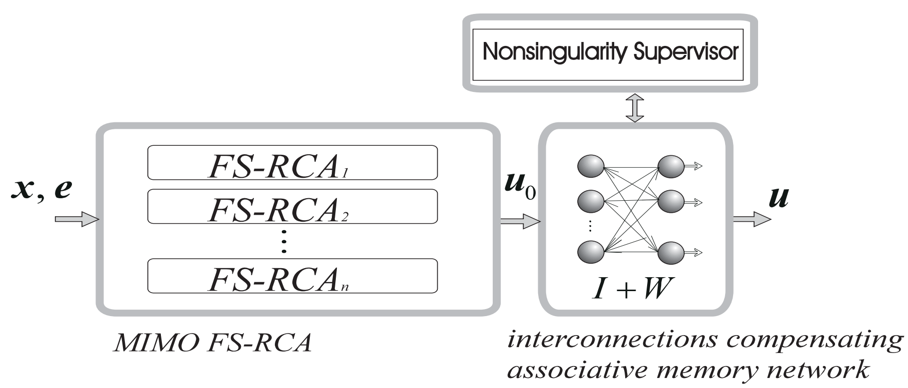

Figure 4: Components of the proposed controller

As shown in Fig.4, the proposed controller is composed 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 behavior for each subsystem.

4.1 Analytical Form of FS-RCA

First, consider the multi-input-single-output FS-RCA presented in previous section. For derivation of fuzzy control as a kind of decoupling control law, we construct a class of fuzzy system by specifying its four components. This class is that the following assumptions hold.

A1 *Membership Function*: Let the fuzzy sets A_1^j, A_2^j, $\cdots$, A_n^j and B^j in the rule base be in LR parametrization, or

$$A_k^j(s_k) = \begin{cases} L\left((c_{s,k}^j - s_k)/a_{L,s}^j\right), & if\ s_k \le c_{s,k}^j \\ R\left((s_k - c_{s,k}^j)/a_{R,s}^j\right), & if\ s_k > c_{s,k}^j, s_k \in S_k, k = 1, \cdots, n \end{cases} \quad (12)$$

or

$$B^j(u) = \begin{cases} L\left((c_u^j - u)/a_{L,u}^j\right), & if\ u \le c_u^j \\ R\left((u - c_u^j)/a_{R,u}^j\right), & if\ u > c_u^j, u \in U \end{cases} \quad (13)$$

where $c_{s,k}^j$(or c_u^j), $a_{L,s}^j$, $a_{R,s}^j$ (or $a_{L,u}^j$, $a_{R,u}^j$), denote, respectively, the center, left spread, and right spread of membership function A_k^j (or B^j).

Typical L (or R) functions, defined in terms of some generic argument x, are

$$L(x)(or R(x)) = \begin{cases} max(0, 1 - |x|^b), & TYPE\ I\ \ or \\ exp(-|x|^b), & TYPE\ II\ \ or \\ (1 + |x|^b)^{-1}, & TYPE\ III\ \ or \\ a_3x^3 + a_2x^2 + a_1x + a_0, & TYPE\ IV \end{cases} \quad (14)$$

where $b > 0$ controls the curvature. This paper uses TYPE III LR parametrization as membership functions.

A2 *Fuzzifier*: The fuzzifier maps a crisp input $\boldsymbol{s} = \boldsymbol{s}^0 = (s_1^0, \cdots, s_n^0)^T \in S$ to a fuzzy set $\boldsymbol{A}'(\boldsymbol{s})$ in S. In this paper, we use the singleton fuzzifier, i.e., $\boldsymbol{A}'(s) = 1$ for $s_k = s_k^0$ and $\boldsymbol{A}'(s) = 0$, otherwise, $k = 1, \cdots, n$, where s_k^0 is the support of the singleton fuzzy set.

A3 *Fuzzy Implication*: This paper adopts the local mean-of-maximum (LMOM) [31] method as the implication method in the fuzzy system. Then the operation of implication in (6) can be written explicitly as

$$I(\boldsymbol{A}^j(\boldsymbol{s}), \tilde{B}^j) = \begin{cases} \boldsymbol{A}^j(\boldsymbol{s}), & for\ \ u = \tilde{c}_u^j \\ 0, & otherwise \end{cases} \quad (15)$$

where $\tilde{c}_u^j$ denotes the location of the singleton implication fuzzy set. It is defined by (see Fig. 5)

$$\tilde{c}_u^j = the\ centroid\ of\ the\ set\ \{u :\ \tilde{B}^j(u) \ge A_1^j(s_1) * \cdots * A_n^j(s_n)\} \quad (16)$$

A4 *T-norm*: We assume that the operation "$*$" is the (algebraic) product.

A5 *Defuzzification*: In this paper, the center-average defuzzification method [14] is used to map the fuzzy set $F(u)$ in U to a crisp point u_0 given by (10).

Then under the assumptions *A2*, *A3* and *A4*, for input $\boldsymbol{s} = \boldsymbol{s}^0$, (6) can be written as

$$\begin{aligned} \boldsymbol{A}'(\boldsymbol{s}^0) \circ R_{\boldsymbol{S}}^j(\boldsymbol{s}^0, u) &= \operatorname*{Sup}_{u \in U} \left[\boldsymbol{A}' * I(\boldsymbol{A}^j(\boldsymbol{s}^0), \tilde{B}^j(u))\right] \\ &= I(\boldsymbol{A}^j(\boldsymbol{s}^0), \tilde{B}^j(u)) \\ &= \begin{cases} \boldsymbol{A}^j(\boldsymbol{s}^0), & for\ \ u = \tilde{c}_u^j \\ 0, & otherwise \end{cases} \end{aligned} \quad (17)$$

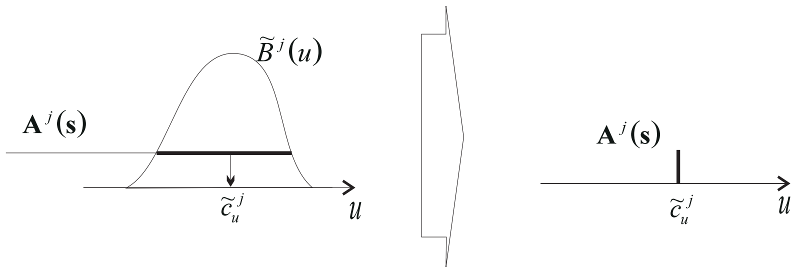

Figure 5: The modified local mean-of-maximum implication method

where we denote $\boldsymbol{A}^j(\boldsymbol{s}^0)$ as matching degree corresponding to the numerical input $\boldsymbol{s}^0$.

Thus after defuzzification by the center-average method (10), for input $\boldsymbol{s}^0$, the output of the FS-RCA becomes

$$u_0(t) = \frac{\sum_j \omega^j \cdot \boldsymbol{A}^j(\boldsymbol{s}^0) \cdot \tilde{c}_u^j}{\sum_j \omega^j \cdot \boldsymbol{A}^j(\boldsymbol{s}^0)} \tag{18}$$

This is the output of fuzzy system.

Next, by imposing further assumptions of each phase in the FS-RCA, (18) will be in the form of feedback linearizing control law or decoupling control law.

- *Membership Function:* For the derivation of analytical form of the fuzzy controller as a decoupling control, two types of antecedent fuzzy sets are used. For $\boldsymbol{s} = \boldsymbol{x}$, A_k^j is the Gaussian membership function

$$A_k^j(x_k) = exp(-(x_k - c_{s,k}^j)^2) \tag{19}$$

For $\boldsymbol{s} = \boldsymbol{e}$, a class of bell-shape functions is chosen with

$$A_k(e_k) + \tilde{A}_k(e_k) = 1 \tag{20}$$

for two arbitrary neighboring membership functions $A_k(e_k)$ and $\tilde{A}_k(e_k)$. It is reasonable and can be achieved by the LR parametrization using an appropriate pair of $a_{L,s}^j$ and $a_{R,s}^j$. With this choice, one can show the Fact 1:

Fact 1 *Let the membership functions A_k^j, $k = 1, \cdots, n$ of rule base $R_{\boldsymbol{e}}^j$, $j = 1, \cdots, N$ be chosen such that (20) hold. Then we have*

$$\sum_{j=1}^N \boldsymbol{A}^j(\boldsymbol{e}) = 1 \tag{21}$$

where N is the number of rules in $R_{\boldsymbol{e}}^j$, $\boldsymbol{A}^j(\boldsymbol{e}) = A_1^j(e_1) \cdots A_n^j(e_n)$.

Proof: see Appendix A. Q.E.D

- *Rule Credit Assignment (Stage I):* After stage I rule credit assignment, the consequent membership function becomes

$$\tilde{B}^j(u) = \begin{cases} \left(1 + (\frac{c_u^j - u}{\beta^j a_L^j})^{b_L^j}\right)^{-1}, & if\ u \le c_u^j \\ \left(1 + (\frac{u - c_u^j}{\beta^j a_R^j})^{b_R^j}\right)^{-1}, & if\ u > c_u^j \end{cases} \tag{22}$$

where c_u^j, a_R^j, a_L^j are called, respectively, the center (where $\tilde{B}^j(u)$ achieves its maximum (one)), right and left spread of B^j membership function; β^j is the credit. Note that reducing (or increasing) β^j makes the definition of the linguistic term represented by $\tilde{B}^j(u)$ more precise (or broader).

- *Rule Matching Phase:* The rule matching degree is calculated as $\hat{g}_i^j(\boldsymbol{s}) = \boldsymbol{A}_i^j(\boldsymbol{s})$ for ith rule in jth rule base $R_{\boldsymbol{s}}^j$.

- *Fuzzy Implication Phase:*This phase computes the location $\tilde{c}_u^j$ of a singleton implication fuzzy set. For $\boldsymbol{R}_{\boldsymbol{x}}^j$, using *A3* and (22) with $b_L^j = b_R^j = 2$, we have (see Appendix B)

$$\tilde{c}_{u,i}^j = c_{u,i}^j - \beta_{ii}^j a_{LR}^j \sqrt{(\boldsymbol{A}_i^j(\boldsymbol{x}))^{-1} - 1} \tag{23}$$

where $a_{LR}^j = (a_L^j - a_R^j)/2$ is the difference of left and right spread of fuzzy membership function A^j.

. On the other hand, suppose $\boldsymbol{R}_{\boldsymbol{e}-j}$ is chosen to be of Takagi- Sugeno type and its consequent membership function B^j is singleton with support in the form of synthesis input

$$c_u^j = y_d^{(p)} - \boldsymbol{\alpha}^{j^T} \boldsymbol{e} \tag{24}$$

where $\boldsymbol{\alpha}^j = (\alpha_1^j, \cdots, \alpha_n^j)^T$. This paper chooses membership functions with $a_{LR}^j = 0$ and all credits equal to one for error rule base. By this specification, we have

$$\tilde{c}_u^j = c_u^j$$

.

- *Defuzzification phase:* Let $\boldsymbol{\alpha}^j = \boldsymbol{a}$ for all j and choose $\beta^j = 1/\omega^j$ $\boldsymbol{Q} = Block\ diag(Q_1, \cdots, Q_n)$., i.e. the credits in stage I and II are assigned simultaneously. By the use of (21), (23), (24), the equation (18) resolves into (for details, see Appendix C)

$$\begin{aligned} u_0(t) &= \frac{\sum_{j=1}^m \omega^j \cdot \boldsymbol{A}^j(\boldsymbol{x}) \circ \tilde{c}_u^j + \sum_{j=1}^N \boldsymbol{A}^j(\boldsymbol{e}) \circ \tilde{c}_u^j}{\sum_{j=1}^m \omega^j \cdot \boldsymbol{A}^j(\boldsymbol{x}) + \sum_{j=1}^N \boldsymbol{A}^j(\boldsymbol{e})} \\ &= \frac{-\boldsymbol{\theta}^{(ca)^T} \hat{\boldsymbol{f}}_\theta(\boldsymbol{x}) + y_d^{(p)} - \boldsymbol{a}^{\boldsymbol{e}}}{\boldsymbol{\omega}^T \hat{\boldsymbol{g}}_\omega(\boldsymbol{x}) + 1} \end{aligned} \tag{25}$$

where m is the number of rules in $\boldsymbol{R}_{\boldsymbol{x}}$, $\boldsymbol{\theta}^{(ca)} = (\omega^1 c_u^1, \cdots, \omega^m c_u^m, a_{LR}^1, \cdots, a_{LR}^m)^T$, $\hat{\boldsymbol{f}}_\theta = (-\boldsymbol{A}^1, \cdots, -\boldsymbol{A}^m, \hat{f}_{LR}^1, \cdots, \hat{f}_{LR}^m)^T$, $\hat{f}_{LR}^j = \boldsymbol{A}^j \sqrt{(\boldsymbol{A}^j)^{-1} - 1}$, and $\hat{\boldsymbol{g}}_\omega$, $\boldsymbol{\omega}$ are vectors composed, respectively, of $\boldsymbol{A}^j$ and ω^j.

4.2 The Mapping of Fuzzy Tracking Controller into Layered Network Structure

In this section, the integration of (a) *the decentralized FS-RCAs*, (b) *the interconnections compensating associative memory network*, and (c) *the nonsingularity supervisor* as a fuzzy controller is described.

- *The decentralized FS-RCAs*

  In view of the defuzzification formula (25), the defuzzification of the decentralized FS-RCAs can be defined as

$$\boldsymbol{u}_0(t) = \hat{\boldsymbol{D}}^{-1}(\boldsymbol{x}, \Theta^{(\omega)}) \left(\hat{\boldsymbol{f}}(\boldsymbol{x}, \Theta^{(ca)}) + \boldsymbol{r}(\boldsymbol{e}, t)\right) \tag{26}$$

where

$$\hat{\boldsymbol{D}}(\boldsymbol{x},\Theta^{(\omega)}) = \begin{bmatrix} \boldsymbol{\omega}_{11}^T\hat{\boldsymbol{g}}_{\omega 1}(\boldsymbol{x})+1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{\omega}_{nn}^T\hat{\boldsymbol{g}}_{\omega n}(\boldsymbol{x})+1 \end{bmatrix},$$

$$\hat{\boldsymbol{f}}(\boldsymbol{x},\Theta^{(ca)}) = \begin{bmatrix} -\boldsymbol{\theta}_1^{(ca)^T}\hat{\boldsymbol{f}}_{\theta 1}(\boldsymbol{x}) \\ \vdots \\ -\boldsymbol{\theta}_n^{(ca)^T}\hat{\boldsymbol{f}}_{\theta n}(\boldsymbol{x}) \end{bmatrix}, \boldsymbol{r}(\boldsymbol{e},t) = \begin{bmatrix} y_{1d}^{(p)} - \boldsymbol{\alpha}_1^T e_1 \\ \vdots \\ y_{nd}^{(p)} - \boldsymbol{\alpha}_n^T e_n \end{bmatrix}, \tag{27}$$

with $\Theta^{(\omega)}=(\boldsymbol{\theta}_1^{(\omega)},\cdots,\boldsymbol{\theta}_n^{(\omega)})^T$, $\boldsymbol{\theta}_i^{(\omega)}=(\boldsymbol{\omega}_{i1},\cdots,\boldsymbol{\omega}_{in})^T$, $\boldsymbol{\omega}_{ir}=(\omega_{ij}^1,\cdots,\omega_{ij}^m)^T$, $\hat{\boldsymbol{g}}_{\omega i}=(\hat{g}_{\omega i}^1,\cdots,\hat{g}_{\omega i}^m)^T$, $\Theta^{(ca)}=(\boldsymbol{\theta}_1^{(ca)},\cdots,\boldsymbol{\theta}_n^{(ca)})^T$, $\boldsymbol{\theta}_i^{(ca)}=(\omega_{ii}^1 c_u^1,\cdots,\omega_{ii}^m c_u^m, a_{LR,i}^1,\cdots,a_{LR,i}^m)^T$, $\hat{\boldsymbol{f}}_{\theta i}=(-\boldsymbol{A}_i^1,\cdots,-\boldsymbol{A}_i^m,\hat{f}_{LR,i}^1,\cdots,\hat{f}_{LR,i}^m)^T$, $\hat{f}_{LR,i}^j=\boldsymbol{A}_i^j\sqrt{(\boldsymbol{A}_i^j)^{-1}-1}$.

- *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, $\boldsymbol{u}_0$, into a new vector $\boldsymbol{u}$, the control action, by the operator $\boldsymbol{M}$ defined as

$$\boldsymbol{u}(t) = \boldsymbol{M}(\boldsymbol{u}_0) = (I_n + \boldsymbol{W})(\boldsymbol{u}_0) \tag{28}$$

$$\boldsymbol{W} = -(I_n + \hat{\boldsymbol{C}}^{-1}\hat{\boldsymbol{D}})^{-1} \tag{29}$$

  where

$$\hat{\boldsymbol{C}}(\boldsymbol{x},\Theta^{(\omega)}) = \begin{bmatrix} 0 & \boldsymbol{\omega}_{12}^T\hat{\boldsymbol{g}}_{\omega 1}(\boldsymbol{x}) & \cdots & \boldsymbol{\omega}_{1n}^T\hat{\boldsymbol{g}}_{\omega 1}(\boldsymbol{x}) \\ \boldsymbol{\omega}_{21}^T\hat{\boldsymbol{g}}_{\omega 2}(\boldsymbol{x}) & 0 & \cdots & \boldsymbol{\omega}_{2n}^T\hat{\boldsymbol{g}}_{\omega 2}(\boldsymbol{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\omega}_{n1}^T\hat{\boldsymbol{g}}_{\omega n}(\boldsymbol{x}) & \boldsymbol{\omega}_{n2}^T\hat{\boldsymbol{g}}_{\omega n}(\boldsymbol{x}) & \cdots & 0 \end{bmatrix}. \tag{30}$$

- *The Nonsingularity Supervisor*

  Since such a weight matrix W in (29) 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 (26), (28), (29) and applying the Matrix Inversion Lemma [39] $(A+BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B+C^{-1})^{-1}DA^{-1}$, the defuzzified output of the fuzzy controller resolves into

$$\begin{aligned} \boldsymbol{u}(t) &= \left(I_n - (I_n + \hat{\boldsymbol{C}}^{-1}\hat{\boldsymbol{D}})^{-1}\right)\hat{\boldsymbol{D}}^{-1}\left(\hat{\boldsymbol{f}}(\boldsymbol{x},\Theta^{(ca)}) + \boldsymbol{r}(\boldsymbol{e},t)\right) && (31) \\ &= \hat{\boldsymbol{G}}^{-1}\left(\hat{\boldsymbol{f}}(\boldsymbol{x},\Theta^{(ca)}) + \boldsymbol{r}(\boldsymbol{e},t)\right) && (32) \end{aligned}$$

where $\hat{\boldsymbol{G}} = \hat{\boldsymbol{C}} + \hat{\boldsymbol{D}}$. The invertibility of $\hat{\boldsymbol{G}}$ in (32) can be guaranteed by proper choices of controller parameters (see Section 6).

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 a combination of credits of a fuzzy rule. The mapped layered network consists of four layers: an input layer, an output layer and two layers of nonlinear nodes. The nodes are interconnected layer by layer. Each layer corresponds, respectively, to a substage shown in the fuzzy system of Fig. 2. This mapping is shown in Fig. 6. This structure allows the input be fuzzified/defuzzified in a parallel way by simultaneously matching membership functions encoded in the nodes. With the network structure, the fuzzy controller has a total of four layers :

- *Layer 1:* Each node denotes the input $\boldsymbol{e}$ or $\boldsymbol{x}$ to the fuzzy system.
- *Layer 2:* Each node calculates the rule matching degree $\hat{g}_i^j(\boldsymbol{s}) = \boldsymbol{A}_i^j(\boldsymbol{s})$.
- *Layer 3:* Each node in this layer obtains the singleton implication fuzzy set and computes its location $\tilde{c}_{u,i}^j$ by (23) and (24).
- *Layer 4:* This layer contains n nodes, which calculate the decoupling control according to (32).

Though the layered structure of the fuzzy controller is fixed, the connection strengths of the layers can be adjustable. Our focus in the following section will be on the network weights learning rules.

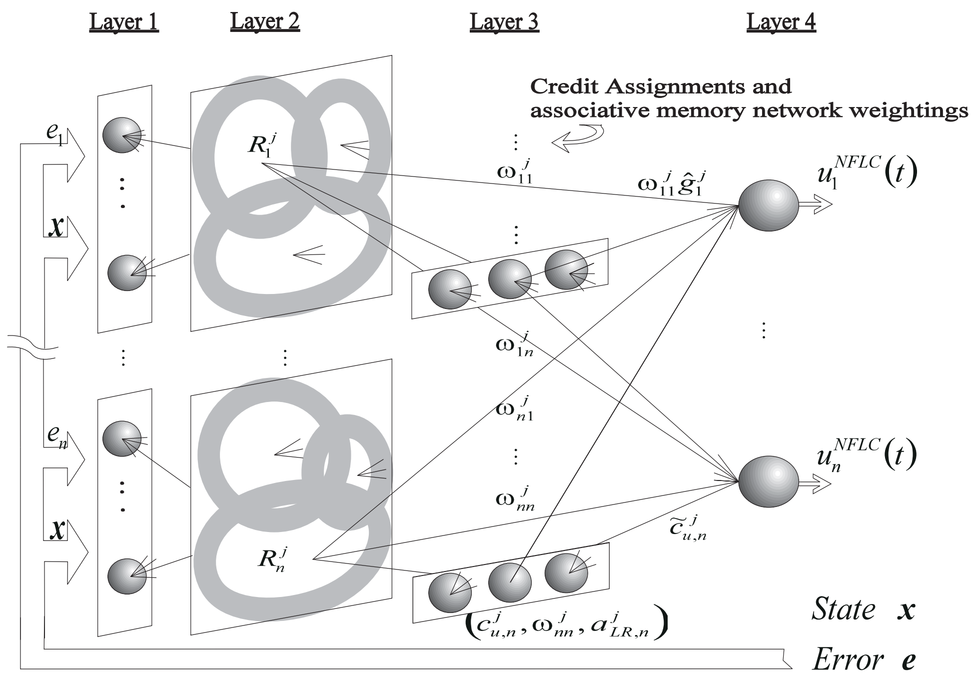

Figure 6: The network structure of the fuzzy controller

5 Parameters Adaptation Algorithm

In this section, we present an adaptation algorithm for parameters of neuro-fuzzy system (Fig. 7). It is assumed that, given any uniform bounds ϵ_f, ϵ_g, there exist parameter vectors $\boldsymbol{\theta}_1^*, \cdots, \boldsymbol{\theta}_n^*$ such that the network approximation errors satisfy

$$\begin{aligned} \max_{\boldsymbol{x}} \left\| \boldsymbol{f}(\boldsymbol{x}, t) - \hat{\boldsymbol{f}}(\boldsymbol{x}, \Theta^{(ca)*}) \right\| &\leq \epsilon_f, \\ \max_{\boldsymbol{x}} \left\| \boldsymbol{G}(\boldsymbol{x}) - \hat{\boldsymbol{G}}(\boldsymbol{x}, \Theta^{(\omega)*}) \right\| &\leq \epsilon_g. \end{aligned} \tag{33}$$

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

$$\dot{\theta}_i = R_i^{-1} w_i b_i^T P_i e_i \tag{34}$$

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

$$P_i A_i + A_i^T P_i + \frac{1}{\rho^2} P_i b_i b_i^T P_i + Q_i = 0 \tag{35}$$

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

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 (34). 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 (34). 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 :

$$\begin{aligned} \dot{\theta}_i &= 0, \qquad\qquad if\ \ e^T P b b^T P e \le d_0^2 \\ &= (I - d_{M_i}(\theta_i)\theta_{i\perp}\theta_{i\perp}^T) R_i^{-1} w_i b_i^T P_i e_i,\ otherwise \end{aligned} \tag{36}$$

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

$$d_{M_i}(\theta_i) := \begin{cases} 0 & if\ \ \theta_i^T (R_i^{-1} w_i b_i^T P_i e_i) \le 0 \\ min\,[1, dist(\theta_i, M_i)/\varepsilon^*], \varepsilon^* > 0, & otherwise \end{cases} \tag{37}$$

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

5.1 Adaptation Algorithm for Systems with Symmetric Decoupling Matrix

The lower component of $\dot{\boldsymbol{\theta}}_i$ can be denoted as

$$\dot{\boldsymbol{\theta}}_i^{(\omega)} = \begin{cases} 0, & if\ \boldsymbol{e}^T \boldsymbol{P} \boldsymbol{b} \boldsymbol{b}^T \boldsymbol{P} \boldsymbol{e} \le d_0^2 \\ (I - d_{M_i} \boldsymbol{\theta}_{i\perp} \boldsymbol{\theta}_{i\perp}^T) {R_i^{(\omega)}}^{-1} \boldsymbol{w}_i^{(\omega)} b_i^T P_i e_i, & otherwise. \end{cases} \tag{38}$$

Suppose $\boldsymbol{G}(\boldsymbol{x})$ in (3) is symmetric. We can let $\boldsymbol{\omega}_{ir} = \boldsymbol{\omega}_{ri}$ so that the number of parameters could be reduced. To meet this need, the adaptation law is modified as follows: The adaptation algorithm $\dot{\boldsymbol{\theta}}_i^{(\omega)}$ is replaced by $\dot{\boldsymbol{\psi}}_i^{(\omega)}$ defined by

$$\dot{\boldsymbol{\psi}}_i^{(\omega)} = ith\ row\ of\ \frac{1}{2}(\dot{\boldsymbol{\theta}}_i^{(\omega)} + {\dot{\boldsymbol{\theta}}_i^{(\omega)}}^T) \tag{39}$$

Remark:
It is interesting to observe that the component matrices ($R_i^{(c)}$, $R_i^{(a)}$) and $R_i^{(\omega)}$ of R_i is closely related

to the inverse of the adaptation gains in adaptation laws of $\boldsymbol{\theta}_i^{(ca)}$ and $\boldsymbol{\theta}_i^{(\omega)}$. This provides a guide for choosing the weighting matrices. Take robotic manipulators as example. For robot dynamics the variations of the components of $\boldsymbol{G}$ are usually smaller than those of $\boldsymbol{f}$, then the adaptation gain of $\boldsymbol{\theta}_i^{(\omega)}$ could be set smaller then that of $\boldsymbol{\theta}_i^{(ca)}$.

6 Guaranteed Tracking Performance

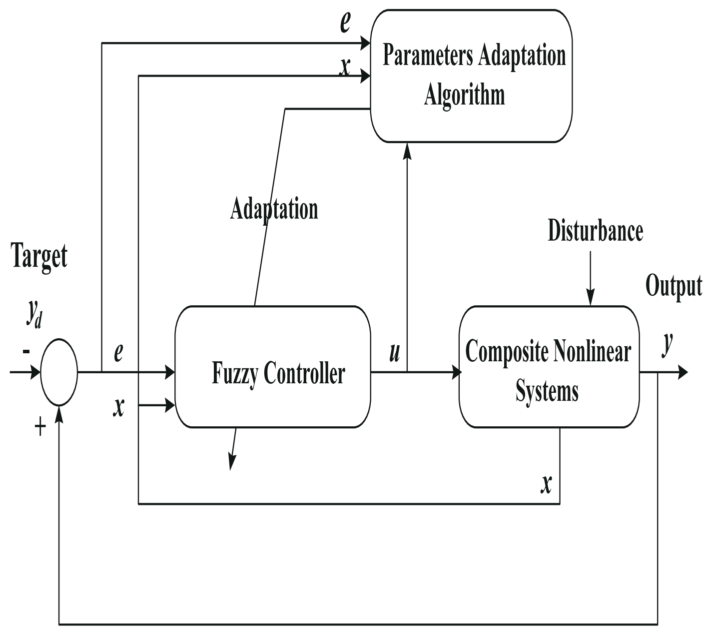

Figure 7: The adaptive fuzzy control system

In this section, stability and performance of the fuzzy tracking system, depicted in Fig. 7, is investigated. Let the parameters estimation error be defined as $\tilde{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_i^*$ where $\boldsymbol{\theta}_i$ represents actual parameter used in the controller and is tuned by deadzone adaptation algorithm. In terms of $\tilde{\theta}_{\mathbf{i}}$ the error equation can be rewritten as

$$\dot{e}_i = A_i e_i - b_i \boldsymbol{w}_i^T \tilde{\boldsymbol{\theta}}_i + b_i \xi_i \tag{40}$$

where

$$\begin{aligned} \xi_i &= \left(f_i(\boldsymbol{x}, t) - \boldsymbol{\theta}_i^{(ca)*^T} \hat{\boldsymbol{f}}_{\theta i} \right) \\ &+ \sum_{j=1}^{n} \left(g_{ij}(\boldsymbol{x}) - \boldsymbol{\omega}_{ij}^{*T} \hat{\boldsymbol{g}}_{\omega i}(\boldsymbol{x}) \ -1 \right) u_j(t) + v_i \end{aligned} \tag{41}$$

represents the lumped disturbance term of *i-th* subsystem due to the network approximation error and external disturbance.

6.1 Feasibility of the Controller

If there exist $\bar{\epsilon}_g$ and $\bar{\delta}_g$ small enough such that $\epsilon_g \le \bar{\epsilon}_g$ and $\|\tilde{\Theta}^{(\omega)}\| \le \bar{\delta}_g$, it can be shown that $\hat{\boldsymbol{G}}^{-1}(\boldsymbol{x}, \Theta^{(\omega)})$ exists, which in turn, guarantees the feasibility of the controller (32). Let $\|\boldsymbol{G}\| \ge g > 0$.

From

$$\begin{aligned}\|\hat{\boldsymbol{G}}(\boldsymbol{x},\Theta^{(\omega)})-\boldsymbol{G}(\boldsymbol{x})\| &\le \|\hat{\boldsymbol{G}}(\boldsymbol{x},\Theta^{(\omega)})-\hat{\boldsymbol{G}}(\boldsymbol{x},\Theta^{(\omega)*})\|+\|\hat{\boldsymbol{G}}(\boldsymbol{x},\Theta^{(\omega)*})-\boldsymbol{G}(\boldsymbol{x})\| \\ &\le \bar{\delta}_g+\bar{\epsilon}_g \end{aligned} \tag{42}$$

we have

$$\begin{aligned}\|\boldsymbol{G}^{-1}(\boldsymbol{x})\hat{\boldsymbol{G}}(\boldsymbol{x},\Theta^{(\omega)})-\boldsymbol{G}(\boldsymbol{x}))\| &\le \frac{1}{g}(\bar{\delta}_g+\bar{\epsilon}_g) \\ &\le 1, \end{aligned} \tag{43}$$

if ϵ_g and δ_g are small enough. Then, we have [39]: $I_n+\boldsymbol{G}^{-1}(\boldsymbol{x})(\hat{\boldsymbol{G}}(\boldsymbol{x},\Theta^{(\omega)})-\boldsymbol{G}(\boldsymbol{x}))$ is invertible. This, in turn, implies that

$$\boldsymbol{G}(\boldsymbol{x})(I_n+\boldsymbol{G}^{-1}(\boldsymbol{x})(\hat{\boldsymbol{G}}(\boldsymbol{x},\Theta^{(\omega)})-\boldsymbol{G}(\boldsymbol{x})))=\hat{\boldsymbol{G}}(\boldsymbol{x},\Theta^{(\omega)})$$

is invertible.

6.2 Initial Weights for Better Tracking Performance

Let the available knowledge about the nonlinear system (3) be encoded in the form of "approximation to the decoupling matrix $\boldsymbol{G}(\boldsymbol{x})$" denoted by $\boldsymbol{G}_0(\boldsymbol{x}|\ \textit{nominal plant parameters})$. The accuracy requirement that ϵ_g is sufficiently small can be satisfied by selecting a set of proper initial weights $\Theta_0^{(\omega)}$ by off-line training using the least squares algorithm [40]. Our approach is based on the element-by-element minimization of the following objective function at a set of training data $\{\boldsymbol{x}^{(k)}\}$:

$$\sum_k \left\|\boldsymbol{G}_0(\boldsymbol{x}^{(k)}|\ \textit{nominal plant parameters})-\hat{\boldsymbol{G}}(\boldsymbol{x}^{(k)},\Theta_0^{(\omega)})\right\|^2$$

where the training data $\{\boldsymbol{x}^{(k)}\}$ are either the sampled points along the desired trajectories or points near them. Thus, a high level of accuracy can be achieved not only at a set of discrete points, but along the points on the desired trajectory.

6.3 Stability and Performance Analysis

The following theorem shows the properties of the fuzzy controller with the deadzone parameters adaptation algorithm.

Theorem 1 Consider the composite nonlinear systems (3) with unknown but bounded $f_i(\boldsymbol{x},t)$, $v_i(\boldsymbol{x},t)$, $i=1,\cdots,n$ and nonsingular matrix $\boldsymbol{G}$. Assume that the controller (32) is adopted with the adaptation algorithm (36) . Then in the bounded state space $\boldsymbol{x}\in\Omega=\{\boldsymbol{x}:\|\boldsymbol{x}\|\ \textit{is bounded}\}$, we have $\boldsymbol{\theta}_i$ and the control input $\boldsymbol{u}(t)$ are bounded. Let $\boldsymbol{\xi}=(\xi_1(\boldsymbol{x},t),\cdots,\xi_n(\boldsymbol{x},t))^T$. Assume that there exists $\bar{\xi}=\mathrm{Sup}_{\boldsymbol{x},t}\|\xi_i(\boldsymbol{x},t)\|^2$. Then $\boldsymbol{e}$ converges to the residual set $\{\boldsymbol{e}:\boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e}\le\rho^2\bar{\xi}\ \textit{or}\ \boldsymbol{e}^T\boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^T\boldsymbol{P}\boldsymbol{e}\le d_0^2\}$, where $\boldsymbol{Q}=\textit{Block diag}(Q_1,\cdots,Q_n)$. Moreover, for the case that ϵ_f and ϵ_g are small enough such that $\|\boldsymbol{\xi}\|\le\frac{1}{2\rho^2}d_0$, then $\boldsymbol{e}$ converges to the deadzone $\{\boldsymbol{e}:\boldsymbol{e}^T\boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^T\boldsymbol{P}\boldsymbol{e}\le d_0^2\}$.

Proof: Let V_θ be

$$V_\theta=\frac{1}{2}\sum_{i=1}^n\boldsymbol{\theta}_i^T\boldsymbol{\theta}_i \tag{44}$$

then we have $\dot{V}_\theta=\frac{1}{2}\sum_{i=1}^n\boldsymbol{\theta}_i^T\dot{\boldsymbol{\theta}}_i\le 0$ or $\boldsymbol{\theta}_i$ will not be adapted whenever $\boldsymbol{\theta}_i\in\partial\boldsymbol{M}_{\epsilon^*}$, the union of $\boldsymbol{M}_i$ and a boundary layer of thickness ϵ^*. Therefore, we can guarantee the boundedness of $\boldsymbol{\theta}_i$. Thus the arguments $\boldsymbol{x}$, $\boldsymbol{\theta}_i$, in (32) are bounded by the assumption of $\boldsymbol{x}\in\Omega$. Thus $\boldsymbol{u}(t)$ is bounded.

Consider the tracking error equation. Choose the Lyapunov function candidate as

$$\boldsymbol{V} = V_1 + \cdots + V_n \tag{45}$$

where

$$V_i(t) = \begin{cases} \frac{1}{2}d_0^2 + \frac{1}{2}\tilde{\boldsymbol{\theta}}_i^T \boldsymbol{R}_i \tilde{\boldsymbol{\theta}}_i, & if\ \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^T \boldsymbol{P}\boldsymbol{e} \le d_0^2 \\ \frac{1}{2}e_i^T P_i e_i + \frac{1}{2}\tilde{\boldsymbol{\theta}}_i^T \boldsymbol{R}_i \tilde{\boldsymbol{\theta}}_i, & otherwise, \end{cases} \tag{46}$$

Taking the time derivative of V_i, we have $\dot{V}_i = 0$, if $\boldsymbol{e}^T \boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^T \boldsymbol{P}\boldsymbol{e} \le d_0^2$, while for $\boldsymbol{e}^T \boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^T \boldsymbol{P}\boldsymbol{e} > d_0^2$ we have

$$\begin{aligned} \dot{V}_i(t) &= e_i^T P_i \dot{e}_i + \tilde{\boldsymbol{\theta}}_i^T (I - d_M \boldsymbol{\theta}_{i\perp}\boldsymbol{\theta}_{i\perp}^T)\boldsymbol{w}_i b_i^T P_i e_i, \\ &= \frac{1}{2}e_i^T (A_i^T P_i + P_i A_i)e_i + \xi_i b_i^T P_i e_i \\ &\quad -d_{M_i}\tilde{\boldsymbol{\theta}}_i^T \boldsymbol{\theta}_{i\perp}\boldsymbol{\theta}_{i\perp}^T \boldsymbol{w}_i b_i^T P_i e_i, \end{aligned} \tag{47}$$

Suppose the parameter bounding set M_i is appropriately selected such that $\boldsymbol{\theta}_i^*$ is in the interior of $\boldsymbol{M}_i$ and $\theta_i^T R_i \theta_i > \theta_i^{*T} R_i \theta_i^*$ if $\boldsymbol{\theta}_i$ is in the exterior of M_i. Now, we investigate the last term in the the above inequality. By definition of d_{M_i}, for $d_{M_i} \ne 0$, $\boldsymbol{\theta}_i$ is in the exterior of $\boldsymbol{M}_i$ and $\boldsymbol{\theta}_i^T (R_i^{-1}\boldsymbol{w}_i b_i^T P_i e_i) > 0$ which implies $\boldsymbol{\theta}_{i\perp}^T (R_i^{-1}\boldsymbol{w}_i b_i^T P_i e_i) > 0$. Thus, $\theta_i^T (R_i^{-1} w_i b_i^T P_i e_i) > 0$, which implies $\theta_{i\perp}^T (R_i^{-1} w_i b_i^T P_i e_i) > 0$. In addition, by the definition of $\boldsymbol{\theta}_{i\perp}$, we have

$$\tilde{\boldsymbol{\theta}}_i^T R_i \boldsymbol{\theta}_{i\perp} = \boldsymbol{\theta}_i^T R_i \boldsymbol{\theta}_{i\perp} - \boldsymbol{\theta}_i^{*T} R_i \boldsymbol{\theta}_{i\perp} \tag{48}$$

$$= \frac{1}{2\|\boldsymbol{\theta}_i\|}\left((\boldsymbol{\theta}_i^* - \boldsymbol{\theta}_i)^T R_i (\boldsymbol{\theta}_i^* - \boldsymbol{\theta}_i) + \boldsymbol{\theta}_i^T R_i \boldsymbol{\theta}_i - \boldsymbol{\theta}_i^{*T} R_i \boldsymbol{\theta}_i^*\right) > 0 \tag{49}$$

for θ_i in the exterior of M_i. Thus, one can deduce that, outside the deadzone

$$\begin{aligned} \dot{V}_i(t) &\le \frac{1}{2}e_i^T (A_i^T P_i + P_i A_i)e_i + \xi_i b_i^T P_i e_i \\ &\le -\frac{1}{2}e_i^T Q_i e_i + \frac{1}{2}\rho^2 \|\xi_i\|^2 \end{aligned} \tag{50}$$

and

$$\dot{\boldsymbol{V}} \le -\frac{1}{2}\boldsymbol{e}^T \boldsymbol{Q}\boldsymbol{e} + \frac{1}{2}\rho^2 \bar{\xi} \tag{51}$$

Thus $\boldsymbol{e}$ converges to $\{\boldsymbol{e} : \boldsymbol{e}^T \boldsymbol{Q}\boldsymbol{e} \le \rho^2 \bar{\xi}\ or\ \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^T \boldsymbol{P}\boldsymbol{e} \le d_0^2\}$.

Next, consider the case that ϵ_f and ϵ_g are small enough such that $\|\boldsymbol{\xi}\| \le \frac{1}{2\rho^2}d_0$, for $\boldsymbol{e}^T \boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^T \boldsymbol{P}\boldsymbol{e} \le d_0^2$. In this case,

$$\begin{aligned} \dot{\boldsymbol{V}} &= -\frac{1}{2}\boldsymbol{e}^T \boldsymbol{Q}\boldsymbol{e} - \frac{1}{2\rho^2}\boldsymbol{e}^T \boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^T \boldsymbol{P}\boldsymbol{e} + \boldsymbol{\xi}\boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^T \boldsymbol{P}\boldsymbol{e} \\ &\le -\frac{1}{2}\boldsymbol{e}^T \boldsymbol{Q}\boldsymbol{e} + (-\frac{1}{2\rho^2}\|\boldsymbol{e}^T \boldsymbol{P}\boldsymbol{b}\| + \|\boldsymbol{\xi}\|)\|\boldsymbol{b}^T \boldsymbol{P}\boldsymbol{e}\| \\ &\le -\frac{1}{2}\boldsymbol{e}^T \boldsymbol{Q}\boldsymbol{e} \end{aligned} \tag{52}$$

Following the arguments of [41], we can conclude that $\boldsymbol{e}$ converges to the deadzone $\{\boldsymbol{e} : \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^T \boldsymbol{P}\boldsymbol{e} \le d_0^2\}$. Q.E.D.

In applications to the class of composite nonlinear systems whose $\boldsymbol{G}$ matrix is symmetric (e.g. robot manipulators), we have the following.

Corollary 1 *Consider the affine nonlinear system (3) with symmetric* $\boldsymbol{G}$ *matrix. Then for the fuzzy controller (32) with the adaptation algorithm (39), all properties described in Theorem 1 hold.*

Proof: For $\boldsymbol{e}^T\boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^T\boldsymbol{P}\boldsymbol{e} \leq d_0^2$, we have

$$\dot{V}_i(t) = e_i^T P_i \dot{e}_i + \boldsymbol{\theta}_i^{(ca)^T} R_i^{(ca)} \dot{\boldsymbol{\theta}}_i^{(ca)} + \boldsymbol{\theta}_i^{(\omega)^T} R_i^{(\omega)} \dot{\boldsymbol{\theta}}_i^{(\omega)} \tag{53}$$

where $R_i^{(ca)} = Block\ diag(R_i^{(c)}, R_i^{(a)})$. Substituting (39), the above can be rewritten as

$$\dot{V}_i(t) = e_i^T P_i \dot{e}_i + \boldsymbol{\theta}_i^{(ca)^T} R_i^{(ca)} \dot{\boldsymbol{\theta}}_i^{(ca)} + \boldsymbol{\theta}_i^{(\omega)^T} R_i^{(\omega)} \dot{\boldsymbol{\psi}}_i^{(\omega)}. \tag{54}$$

In view of the following properties:

(i) $\tilde{\boldsymbol{\omega}}_{ij} = \tilde{\boldsymbol{\omega}}_{ji}$, since $\boldsymbol{\omega}_{ij} = \boldsymbol{\omega}_{ji}$,

(ii) $R_i^{(j)} = R_j^{(i)}$,

we have the identity

$$\sum_{i=1}^{n} \tilde{\boldsymbol{\theta}}_i^{(\omega)^T} R_i^{(\omega)} \dot{\boldsymbol{\theta}}_i^{(\omega)} = \sum_{i=1}^{n} \tilde{\boldsymbol{\theta}}_i^{(\omega)^T} R_i^{(\omega)} \dot{\boldsymbol{\psi}}_i^{(\omega)}. \tag{55}$$

Thus,

$$\sum_{i=1}^{n} \dot{V}_i|_{along\ (36),(40)} = \sum_{i=1}^{n} \dot{V}_i|_{along\ (39),(40)} \tag{56}$$

so that the results of Theorem 1 hold. Q.E.D.

7 Simulation

In this section, two simulations are presented to demonstrate the effectiveness of the adaptive fuzzy controller.

1) *Inverted Pendulum*: Consider the inverted pendulum depicted in Fig. 8. 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:

$$\begin{aligned} \dot{x_1} &= x_2, \\ \dot{x_2} &= \frac{gsinx_1 - \frac{mlx_2^2 cosx_1 sinx_1}{m_c+m}}{l(\frac{4}{3} - \frac{mcos^2x_1}{m_c+m})} + \frac{\frac{cosx_1}{m_c+m}}{l(\frac{4}{3} - \frac{mcos^2x_1}{m_c+m})}u + v, \\ y &= x_1, \end{aligned} \tag{57}$$

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. In simulation, we choose $|v(t)| \leq 0.5$. This system is unstable if the control u is set to be 0. Fig. 9 shows the membership functions of x_1, x_2 and u. Let $\boldsymbol{M} = \{\boldsymbol{\theta} : |c_u^j| \leq 15, |a_{LR}^j| \leq 6, |\omega^j| \leq 2\}$ and $\alpha_1 = 10$, $\alpha_2 = 100$. Initially, c_{u0}^j are chosen randomly in the interval $[-12, 12]$, and $a_{L0}^j = a_{R0}^j = 2$, $\omega_0^j = 1$. Simulations are performed for $\rho = 0.02$, $Q = 10I_{2\times2}$, $\epsilon* = 0.05$ and $R = Block\ diag[0.02I_{25\times25}, 0.01I_{25\times25}, 1.0I_{25\times25}]$. Fig. 10 shows the result of tracking a periodic trajectory $y_d = \frac{\pi}{15}sint + \frac{\pi}{30}cost$. The tracking performance of the neuro-fuzzy controller is quite good.

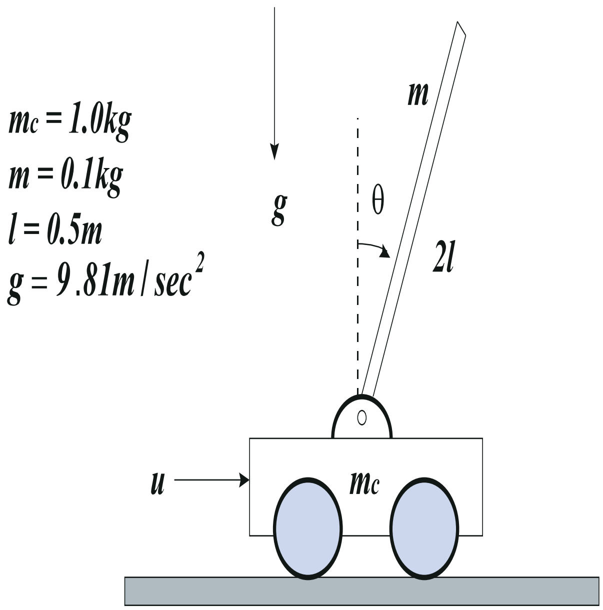

Figure 8: The inverted pendulum system.

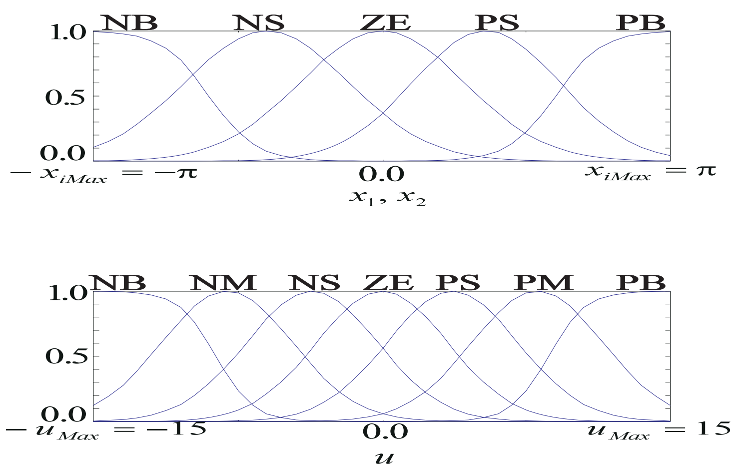

Figure 9: Membership functions for x_1, x_2 and u. N represents negative, P positive, ZE approximately zero, S small, M medium, B big.

2) *Planar Robot*: Consider a two-link robot manipulator depicted in Fig. 11.

The equations of motion for the planar robot are as follows.

$$\begin{bmatrix} (m_1+m_2)r_1^2 + m_2 r_2^2 + 2m_2 r_1 r_2 c_2 + J_1 & m_2 r_2^2 + m_2 r_1 r_2 c_2 \\ m_2 r_2^2 + m_2 r_1 r_2 c_2 & m_2 r_2^2 + J_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -m_2 r_1 r_2 s_2 \dot{q}_1 (\dot{q}_1 + \dot{q}_2) \\ m_2 r_1 r_2 s_2 \dot{q}_2^2 \end{bmatrix} + \begin{bmatrix} ((m_1+m_2) l_1 c_2 + m_2 l_2 c_{12}) g \\ (m_2 l_2 c_{12}) g \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \tag{58}$$

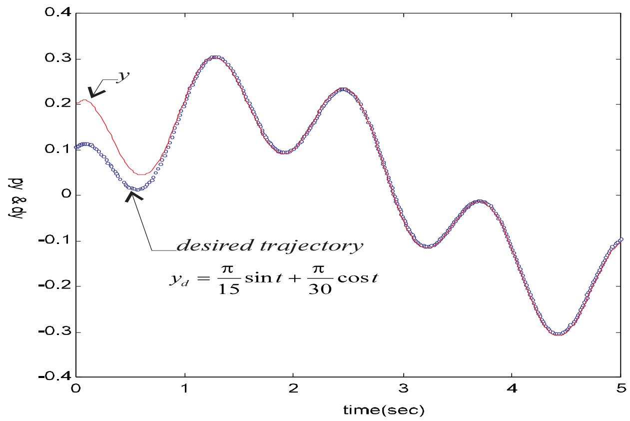

Figure 10: Simulation result of the inverted pendulum.

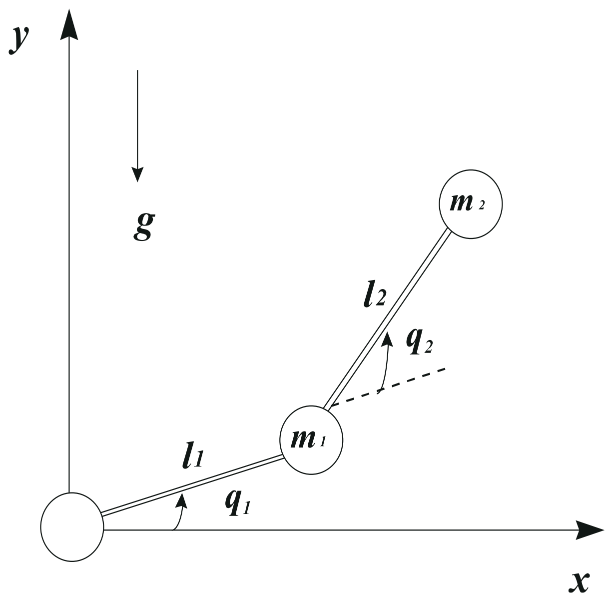

Figure 11: A two-link robot manipulator

where $c_1 \equiv cos(q_1)$, $s_{12} \equiv sin(q_1 + q_2)$, *etc.*. Let the combined effects of friction and the external torque disturbance be assumed as

$$\begin{aligned} d_1 &= 2.0sin(\dot{q}_1) + 2.5sin(\dot{q}_2) + 0.5sin(t) \\ d_2 &= 5.0sin(\dot{q}_1) + 4.0sin(\dot{q}_2) + 0.4sin(t) \end{aligned} \tag{59}$$

The kinematics and inertia parameters of the robot are given by $l_1 = 2.0m$, $l_2 = 1.6m$, $r_1 = 0.5l_1$, $r_2 = 0.5l_2$, $J_1 = J_2 = 5kg \cdot m$, $m_1 = 0.5kg$, $m_2 = 6.35kg$. The excessive ratio between m_1 and m_2 is to emphasize the load effect.

The desired trajectory for the robot to track is given by :

$$q_{1d} = \pi/12 sin(0.5\pi t)$$
$$q_{2d} = 2.5\pi/12 cos(0.5\pi t) + 2.5\pi/24 cos(0.25\pi t) \tag{60}$$

with the initial state $q_1(0) = -1.5rad$, $q_2(0) = -1.2rad$, $\dot{q}_1(0) = 0rad/sec$, $\dot{q}_2(0) = 0rad/sec$. To reduce the number of adaptation parameters, only four linguistic labels for the qualitative statements defined as $\{NB, NS, PS, PB\}$ are used, where $NB : f(x_i) = 1/(1+exp(5(x_i+1)))$, $NS : f(x_i) = exp(-(x_i + 0.6)^2)$, $PS : f(x_i) = exp(-(x_i - 0.6)^2)$, $PB : f(x_i) = 1/(1 + exp(-5(x_i - 1)))$. Moreover, let $a^j_{LR1} = a^j_{LR2} = 0$, i.e., the left and right spreads of the consequent membership function are equal in the adaptation period.

The initial parameters $c^j_{u1}(0)$ and $c^j_{u2}(0)$ are chosen randomly in the interval $(-50, 50)$. To obtain a set of appropriate initial parameters $\Theta^{(\omega)}_0$, thirty-two testing points uniformly distributed along the desired trajectory are used in the least squares minimization process with the following nominal inertia parameters: $J^0_1 = 4.8kg \cdot m$, $J^0_2 = 5.1kg \cdot m$, $m^0_1 = 0.48kg$, $m^0_2 = 6.30kg$. The design parameters in (40) and (58) are $Q_1 = Q_2 = 10I_{2\times2}$, $\epsilon* = 0.05$, $\alpha_{1,1} = \alpha_{2,1} = 100$, $\alpha_{1,2} = \alpha_{2,2} = 65$, $R_1 = Block\ diag[0.005I_{256\times256},\ 3000I_{256\times256},\ 2000I_{256\times256}]$, $R_2 = Block\ diag[0.005I_{256\times256},\ 2000I_{256\times256},\ 3000I_{256\times256}]$.

Simulated responses for different sizes of deadzone are shown in Fig.12 where solid line and dotted line denote $\rho_1 = \rho_2 = 0.01$ and $\rho_1 = \rho_2 = 0.5$, respectively. Only the results of joint 1 are shown since those of joint 2 are relatively good, compared to those of joint 1. It verifies that the smaller the deadzone the higher the tracking accuracy.

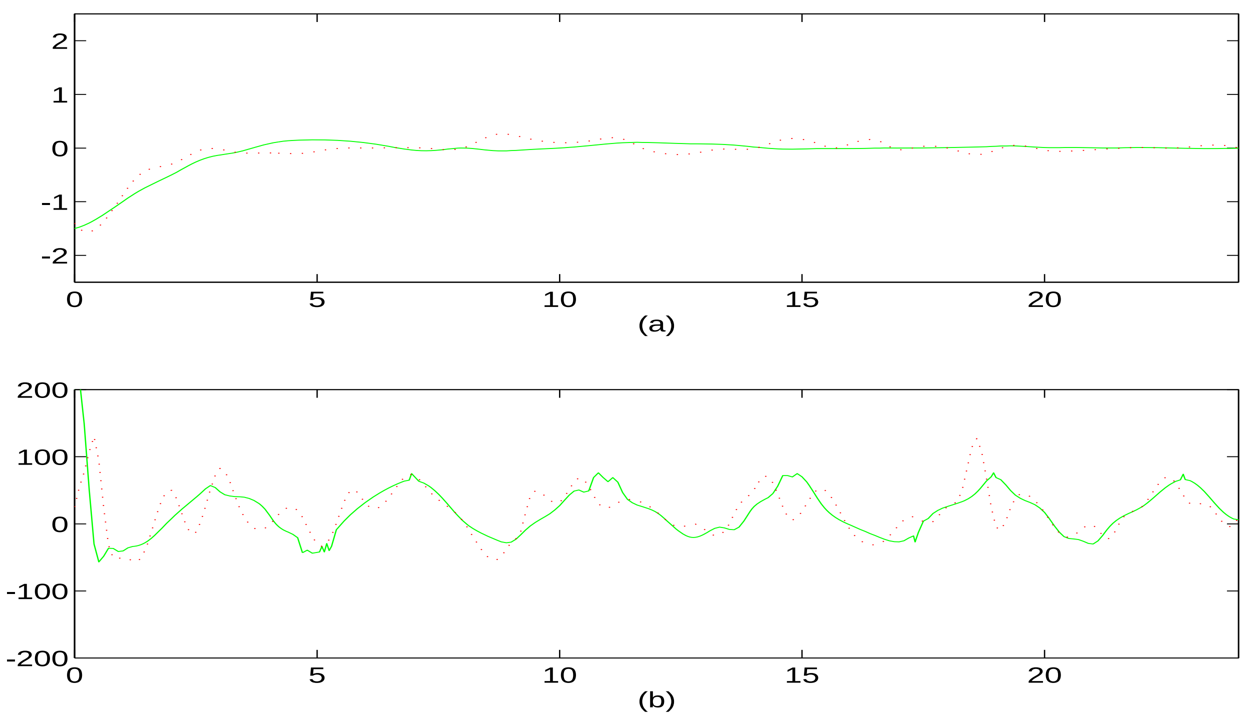

Figure 12: Simulation results for joint 1 (a)position error vs. time (b)applied torque vs. time.

8 Conclusion

For output tracking in composite affine nonlinear systems, this paper proposes a fuzzy controller composed of decentralized approximate reasoning fuzzy system with adjustable rule credit assignment mechanism 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. In this mapping, the weights of network represent a combination of the credits of rules, shapes and locations of membership functions. The weights are tuned via a deadzone adaptation algorithm to compensate for the uncertainties. The incorporation of deadzone in controller achieves on-line computation of decoupling control while guaranteeing the boundedness of network weights and stability and the attenuation of tracking error to a prescribed level in the presence of disturbance and approximation errors. In particular, it is assumed that that the functions $f_i(\boldsymbol{x}, t)$, $v_i(\boldsymbol{x}, t)$ and $g_{ir}(\boldsymbol{x})$ in the nonlinear plant model (1) describing the systems under investigation are bounded and furthermore, their bounds are not explicitly used in controller design. Simulations of the inverted pendulum and a two-link planar robot demonstrate the effectiveness and robustness of the neuro-fuzzy controller in output tracking.

A Appendix

Appendix A

Proof of Fact 1 by Induction:

Suppose there are N rules $R^j_{\boldsymbol{e},i}$ in the rule base $\boldsymbol{R}_{\boldsymbol{e},i}$. For notational simplicity, consider the ith rule base and drop the subscript i later.

In the rule matching phase, suppose that the error $e_1(t)$ has two positive grades of membership, $A_1^p(e_1)$ and $A_1^{p+1}(e_1)$ for some p (see Fig. 13) where the superscript of A_1 denotes some membership function of fuzzy partition and A_1^p, A_1^{p+1} denote two neighboring membership functions in the universe U_1. Likewise, $e_2(t)$ belongs to A_2^q and A_2^{q+1} for some $q, \cdots$, and $e_n(t)$ belongs to A_n^s and A_n^{s+1} for some s.

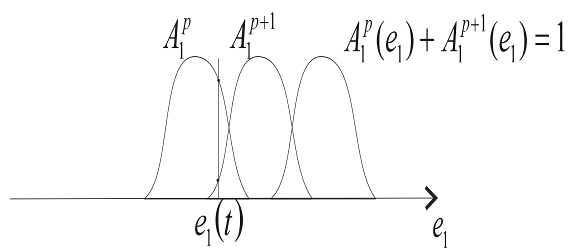

Figure 13: Collections of fuzzy subsets partition.

In this situation, the n-input-one-output system uses 2^n fuzzy rules since each linguistic variable has two linguistic values. Let the following 2^n ($2 \le 2^n \le N$) rules be fired:

$$
\begin{array}{ll}
R_{\boldsymbol{e}}^{(p,q,\cdots,s)}: & IF\ e_1\ is\ A_1^p\ \ AND\ e_2\ is\ A_2^q\ \ AND\cdots AND\ e_n\ is\ A_n^s \\
& THEN\ u\ is\ B^{(p,q,\cdots,s)} \\
R_{\boldsymbol{e}}^{(p+1,q,\cdots,s)}: & IF\ e_1\ is\ A_1^{p+1}\ AND\ e_2\ is\ A_2^q\ \ AND\cdots AND\ e_n\ is\ A_n^s \\
& THEN\ u\ is\ B^{(p+1,q,\cdots,s)} \\
R_{\boldsymbol{e}}^{(p,q+1,\cdots,s)}: & IF\ e_1\ is\ A_1^p\ \ AND\ e_2\ is\ A_2^{q+1}\ AND\cdots AND\ e_n\ is\ A_n^s \\
& THEN\ u\ is\ B^{(p,q+1,\cdots,s)} \\
& \vdots \\
R_{\boldsymbol{e}}^{(p+1,q+1,\cdots,s+1)}: & IF\ e_1\ is\ A_1^{p+1}\ AND\ e_2\ is\ A_2^{q+1}\ AND\cdots AND\ e_n\ is\ A_n^{s+1} \\
& THEN\ u\ is\ B^{(p+1,q+1,\cdots,s+1)}
\end{array}
$$

where the superscript of $R_{\boldsymbol{e}}$ indicates which rule in the rule base $\boldsymbol{R_e}$ is fired.

Now, we prove Fact 1 by induction. Let $n = 1$, we have 2^1 fired rules and $\sum_{j=1}^{N} \boldsymbol{A}^j(\boldsymbol{e}) = A_1^p(e_1)+A_1^{p+1}(e_1)$. Since LR parametrization is chosen such that (20) holds, one has $\sum_{j=1}^{N} \boldsymbol{A}^j(\boldsymbol{e}) = 1$. Suppose for $n =$ some integer k, we have

$$
\begin{aligned}
&\sum_{j=1}^{N} \boldsymbol{A}^j(\boldsymbol{e}) \\
&= A_1^p(e_1)A_2^q(e_2)\cdots A_k^s(e_k) + \\
&\quad A_1^{p+1}(e_1)A_2^q(e_2)\cdots A_k^s(e_k) + \\
&\quad A_1^p(e_1)A_2^{q+1}(e_2)\cdots A_k^s(e_k) + \\
&\quad\quad \vdots \\
&\quad +A_1^{p+1}(e_1)A_2^{q+1}(e_2)\cdots A_k^{s+1}(e_k) \\
&= A_1^p(e_1)A_2^q(e_2)\cdots A_k^s(e_k) + \\
&\quad (1-A_1^p(e_1))A_2^q(e_2)\cdots A_k^s(e_k) + \\
&\quad A_1^p(e_1)(1-A_2^q(e_2))\cdots A_k^s(e_k) + \\
&\quad\quad \vdots \\
&\quad +(1-A_1^p(e_1))(1-A_2^q(e_2))\cdots(1-A_k^s(e_k)) \\
&= 1.
\end{aligned}
\tag{61}
$$

For $n = k+1$,

$$
\begin{aligned}
&\sum_{j=1}^{N} \boldsymbol{A}^j(\boldsymbol{e}) \\
&= [A_1^p(e_1)A_2^q(e_2)\cdots A_k^s(e_k) + \\
&\quad A_1^{p+1}(e_1)A_2^q(e_2)\cdots A_k^s(e_k) + \\
&\quad A_1^p(e_1)A_2^{q+1}(e_2)\cdots A_k^s(e_k) + \\
&\quad\quad \vdots \\
&\quad +A_1^{p+1}(e_1)A_2^{q+1}(e_2)\cdots A_k^{s+1}(e_k))]A_{k+1}^t(e_{k+1}) \\
&\quad +[A_1^p(e_1)A_2^q(e_2)\cdots A_k^s(e_k) + \\
&\quad A_1^{p+1}(e_1)A_2^q(e_2)\cdots A_k^s(e_k) + \\
&\quad A_1^p(e_1)A_2^{q+1}(e_2)\cdots A_k^s(e_k) + \\
&\quad\quad \vdots \\
&\quad +A_1^{p+1}(e_1)A_2^{q+1}(e_2)\cdots A_k^{s+1}(e_k))](1-A_{k+1}^t(e_{k+1}))
\end{aligned}
\tag{62}
$$

In view of (83), we deduce that $\sum_{j=1}^{N} \boldsymbol{A}^j(\boldsymbol{e}) = 1$.

Q.E.D.

Appendix B

The derivation of (23) is provided in this appendix.

Let $b^j_{L,i} = b^j_{R,i} = 2$. Now, refer to Fig. 5. For input matching degree $\hat{g}^j_i$, at the left intersection point u_A, by (22) we have

$$\hat{g}^j_i = \left(1 + (\frac{c^j_{u,i} - u_A}{\beta^j_{ii} a^j_{L,i}})^2\right)^{-1}$$

or

$$u_A = c^j_{u,i} - \beta^j_{ii} a^j_{L,i} \sqrt{(\hat{g}^j_i)^{-1} - 1}$$

While at the right intersection point u_B, we have similarly the relation

$$\hat{g}^j_i = \left(1 + (\frac{c^j_{u,i} - u_B}{\beta^j_{ii} a^j_{R,i}})^2\right)^{-1}$$

or

$$u_B = c^j_{u,i} + \beta^j_{ii} a^j_{R,i} \sqrt{(\hat{g}^j_i)^{-1} - 1}$$

Thus, by definition,

$$\tilde{c}^j_{u,i} = (u_A + u_B)/2$$

which derives the relation (23).

Appendix C

This appendix derives the weighted center-average defuzzification of single-input single-output fuzzy system.

The weighted center-average defuzzification method considers contributions from all possible control actions and evaluates the final control in proportion to the matching degree of each fuzzy output. Thus, for input s, the output of the fuzzy system is calculated as

$$u = \frac{\sum_j \omega^j \cdot A'(s) \circ R^j_s(s, \bar{u}^j) \cdot \bar{u}^j}{\sum_j \omega^j \cdot A'(s) \circ R^j_s(s, \bar{u}^j)}$$

where $\bar{u}^j$ is the point in U at which $A'(s) \circ R^j_s(s, u)$ achieves its maximum value. Then, using LMOM implication, the above equation reduces to

$$u = \frac{\sum_j \omega^j \cdot A^j(s) \cdot \tilde{c}^j_u}{\sum_j \omega^j \cdot A^j(s)}$$

By the use of (21), (23), (24), the equation is rearranged as

$$\begin{aligned} u &= \frac{\sum_{j=1}^{m} \omega^j \cdot A^j(x) \cdot \tilde{c}^j_u + \sum_{j=1}^{N} A^j(e) \cdot \tilde{c}^j_u}{\sum_{j=1}^{m} \omega^j \cdot A^j(x) + \sum_{j=1}^{N} A^j(e)} \\ &= \frac{-\theta^{(ca)^T} \hat{f}(x) + \ddot{q}_d - a^T e}{\omega^T \hat{g}(x) + 1} \end{aligned}$$

This is the equation (25) for single-input single-output case.

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] J. A. Bernard, "Use of rule-based system for process control," *IEEE Contr. Syst. Mag.*, vol. 8, no. 5, pp. 3–13, 1988.

[5] O. Yagishita, O. Ito, and M. Sugeno, "Application of fuzzy reasoning to the water purification process," in *Industrial Application of Fuzzy Control*, pp. 19–40, M. Sugeno Ed., Amsterdam: North-Holland, 1985.

[6] C. M. Lim and T. Hiyama, "Application of fuzzy logic control to a manipulator," *IEEE Trans. Robotics and Automation*, vol. 7, no. 5, pp. 688–691, 1991.

[7] Y. Kasai and Y. Morimoto, "Electronically controlled continuously variable transmission," in *Proc. Int. Congress on Transportation Electronics*, pp. 33–42, 1988.

[8] 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.

[9] 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.

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

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

[12] 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.

[13] 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.

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

[15] Y.-Q. Zhang and A. Kandel, "Compensatory neurofuzzy systems with fast learning algorithms," *IEEE Trans. Neural Network*, vol. 9, no. 1, pp. 83–105, 1998.

[16] 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.

[17] J. Nie and D. A. Linkens, "Learning control using fuzzified self-organizing radial basis function network," *IEEE Trans. Fuzzy Systems*, vol. 1, no. 4, pp. 280–287, 1993.

[18] G. Feng, "Improved tracking control for robots using neural networks," in *Proc. of the American Control Conference*, vol. 1, pp. 69–73, 1993.

[19] K. S. Ray and D. D. Majumder, "Application of circle criteria for stability analysis of linear siso and mimo systems associated with fuzzy controller," *IEEE Trans. Systems, Man and Cybernetics*, vol. 14, no. 2, pp. 345–349, 1984.

[20] J. B. Kiszka, M. M. Gupta, and P. N. Nikfourk, "Energetistic stability of fuzzy dynamic systems," *IEEE Trans. Systems, Man and Cybernetics*, vol. 15, no. 6, pp. 783–792, 1985.

[21] Y. Y. Chen, "Stability analysis of fuzzy control - a lyapunov approach," in *Proc. Int. Conference on Systems, Man and Cybernetics*, pp. 1027–1031, 1987.

[22] C. Batur and T. Leephakpreeda, "Stability of fuzzy control systems," in *Proc. IEEE Conf. Neural Networks*, vol. 4, pp. 2605–2610, 1994.

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

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

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

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

[27] E. Tzirkel-Hancock and F. Fallside, "Stable adaptive control of nonlinear systems using neural networks," *Int. J. Robust and Nonlinear Control*, vol. 2, pp. 63–86, 1992.

[28] S. Fabri and V. Kadirkamanathan, "Dynamic structure neural networks for stable adaptive control of nonlinear systems," *IEEE Trans. Neural Network*, vol. 7, no. 5, pp. 1151–1167, 1996.

[29] 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.

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

[31] 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.

[32] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic control - part ii," *IEEE Trans. Systems, Man and Cybernetics*, vol. 20, pp. 419–435, 1990.

[33] H. R. Berenji, "Fuzzy logic controllers," in *An Introduction to Fuzzy Logic Applications in Intelligent System* (R. R. Yager and L. A. Zadeh, eds.), (Boston, Mass.), pp. 69–96, Kluwer Academic Publisher, 1991.

[34] M. Mizumoto and H. Zimmermann, "Comparison of fuzzy reasoning methods," *Fuzzy Sets and Systems*, vol. 8, pp. 253–283, 1982.

[35] J. L. Castro, "Fuzzy logic controllers are universal approximators," *IEEE Trans. Systems, Man and Cybernetics*, vol. 25, no. 4, pp. 629–635, 1995.

[36] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic control - part i," *IEEE Trans. Systems, Man and Cybernetics*, vol. 20, pp. 404–418, 1990.

[37] T. Yamakawa, "A fuzzy inference engine in nonlinear analog mode and its application to a fuzzy logic control," *IEEE Trans. Neural Network*, vol. 4, no. 3, pp. 496–522, 1993.

[38] H. Anna, *Knowledge Acquisition for Expert Systems.* New York: McGraw Hill, 1986.

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

[40] D. G. Luenberger, *Linear and Nonlinear Programming.* Reading, Mass.: Addison-Wesley, 1984.

[41] F. C. Chen and C. C. Liu, "Adaptively controlling nonlinear continuous-time systems using multilayer neural network," *IEEE Trans. Automatic Control*, vol. 39, no. 6, pp. 1306–1310, 1994.