Fuzzy Sets and Systems 110 (2000) 351–363

FUZZY sets and systems

www.elsevier.com/locate/fss

Adaptive tuning of the fuzzy controller for robots

Sheng-De Wang *, Chuan-Kai Lin

EE Building, Room 441, Department of Electrical Engineering, National Taiwan University, 1 Roosevelt Rd., Sec. 4, Taipei 106, Taiwan

Received July 1997; received in revised form March 1998

Abstract

An adaptive tuning algorithm of the fuzzy controller is developed for a class of serial-link robot arms. The algorithm can on-line tune parameters of premise and consequence parts of fuzzy rules of the fuzzy basis function (FBF) controller. The main part of the fuzzy controller is a fuzzy basis function network to approximate unknown rigid serial-link robot dynamics. Under some mild assumptions, a stability analysis guarantees that both tracking errors and parameter estimate errors are bounded. Moreover, a robust technique is adopted to deal with uncertainties including approximation errors and external disturbances. Simulations of the proposed controller on the PUMA-560 robot arm demonstrate the effectiveness. © 2000 Elsevier Science B.V. All rights reserved.

1. Introduction

Robot manipulators have highly nonlinear dynamics. A strategy, feedback linearization of nonlinear systems, cancels the nonlinearities of robot manipulators and imposes a desired linear model so that linear control techniques can be applied [12, 2]. However, the method is based on the exact knowledge of robot dynamics. Without knowing the exact knowledge of robot dynamics, a nonlinear component is required to approximate and cancel the dynamics. Neural networks and fuzzy systems provide good solutions to this challenging task. In this paper, we design a fuzzy controller for rigid robot manipulators with completely unknown dynamics.

It has been proved that fuzzy basis function (FBF) expansions can be universal approximators with arbitrarily small errors [16]. Therefore, a fuzzy basis function network is used to approximate and cancel the unknown dynamics of robot manipulators. As in [15], the control structure and learning rules are derived from a Lyapunov theory extension that guarantee both tracking errors and parameter estimate errors in the closed-loop system are bounded. By taking the uncertainties including approximation errors and external disturbances into consideration, such a technique can reject the effects.

Tuning parameters of fuzzy systems has been an active research area in the past two decades. Most of the approaches can only tune parameters of consequence part of fuzzy rules [16–14]. Some approaches can tune parameters of premise and consequence parts of fuzzy rules, however, they do not guarantee global stability and tracking performance [4, 7, 10]. The main topic of this paper is to present an algorithm

* Corresponding author. Tel.: 886-2-23635251 ext. 441; fax: 886-2-23671909.

E-mail address: sdwang@hpc.ee.ntu.edu.tw (S.-D. Wang)

0165-0114/00/$ - see front matter © 2000 Elsevier Science B.V. All rights reserved.

PII: S0165-0114(98)00078-5

to tune all parameters of the fuzzy controller under a perturbation environment. The use of an FBF-based controller in direct closed-loop controllers with the algorithm can guarantee global stability and tracking performance.

The arrangement of the rest of this paper is as follows. In Section 2, the dynamics of the rigid serial-link robot manipulators and an FBF network are introduced. Section 3 presents an algorithm for tuning the FBF-based controller. Based on the Lyapunov synthesis technique, a global stable fuzzy controller in a constructive manner too is developed. In Section 4, the FBF controller is used to track desired trajectories for a popular PUMA-560 robot arm successfully. Finally, Section 5 provides the conclusions.

2. Robot arm dynamics and fuzzy basis function networks

2.1. Robot arm dynamics

Consider a rigid robot manipulator with n serial links described by the equations

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{V_m}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{F}\dot{\boldsymbol{\theta}} + \boldsymbol{\tau_d} = \boldsymbol{\tau} \tag{1}$$

with vector $\boldsymbol{\theta} \in \boldsymbol{R}^n$ being the joint position vector; $\boldsymbol{M}(\boldsymbol{\theta}) \in \boldsymbol{R}^{n\times n}$ being a symmetric positive definite inertia matrix; $\boldsymbol{V_m}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}}$ being a vector of Coriolis and centripetal torques; $\boldsymbol{G}(\boldsymbol{\theta}) \in \boldsymbol{R}^n$ representing the gravitational torques; $\boldsymbol{F} = \boldsymbol{K_\omega} + \boldsymbol{V_f} \in \boldsymbol{R}^{n\times n}$ being a diagonal matrix consisting of the back emf coefficient matrix $\boldsymbol{K_\omega}$ and the viscous friction coefficients matrix $\boldsymbol{V_f}$; $\boldsymbol{\tau_d} \in \boldsymbol{R}^{n\times l}$ being the unmodeled disturbances vector; and $\boldsymbol{\tau} \in \boldsymbol{R}^{n\times l}$ being the vector of control input torques. The structural properties of the robot manipulator such as boundedness of $\boldsymbol{M}(\boldsymbol{\theta})$, $\boldsymbol{V_m}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})$ and $\boldsymbol{\tau_d}$ and skew-symmetry of matrix $\dot{\boldsymbol{M}} - 2\boldsymbol{V_m}$ hold for (1).

2.2. Fuzzy basis function networks

Assume that there are r rules in a fuzzy rule base and each of which has the following form:

If x_1 is $\tilde{A}_{1j}$ and x_2 is $\tilde{A}_{2j}$ and … and x_n is $\tilde{A}_{nj}$

then y_1 is $\tilde{B}_{j1}$ and y_2 is $\tilde{B}_{j2}$ and … y_m is $\tilde{B}_{jm}$,

where $j = 1,2,\ldots,r$, the input vector $\boldsymbol{x} = (x_1,\ldots,x_n)^{\mathrm{T}}$ contains the input variables to the fuzzy system, y_k ($k = 1,2,\ldots,m$) are the output variables of the fuzzy system, and $\tilde{A}_{ij}$ and $\tilde{B}_{jk}$ are linguistic terms characterized by their corresponding fuzzy membership functions $\mu_{\tilde{A}_{ij}}(x_i)$'s and $\mu_{\tilde{B}_{jk}}(y_k)$, respectively. For an FBF network, the membership functions $\mu_{\tilde{A}_{ij}}(x_i)$'s are Gaussian functions. As in [14], we consider the FBF network with singleton fuzzification, product inference, and defining the defuzzifier as a weighted sum of each rule's output. The scheme of the FBF network with n inputs, r rules (hidden units) and m outputs is shown in Fig. 1. Such an FBF network implementing the procedures of fuzzification, fuzzy inference and defuzzification performs the m mappings $f_k : \boldsymbol{R}^n \to \boldsymbol{R}$ according to

$$f_k = \sum_{j=1}^{r} w_{jk}\phi(\|\boldsymbol{x} - \boldsymbol{c}_j\|, \boldsymbol{\sigma}_j), \tag{2}$$

where $\boldsymbol{x} \in \boldsymbol{R}^n$ is the input vector, $\boldsymbol{c}_j \in \boldsymbol{R}^n$ is the center vector of the jth rule, $\boldsymbol{\sigma}_j \in \boldsymbol{R}^n$ is the width vector of the fuzzy basis function $\phi(\cdot)$ and hidden-to-output layer interconnections weights are denoted by w_{jk}. The fuzzy basis function can be represented by

$$\phi_j = \mathrm{e}^{-[((x_1-c_{1j})/\sigma_{1j})^2+\cdots+((x_n-c_{nj})/\sigma_{nj})^2]}$$

or

$$\phi_j = \mathrm{e}^{-[\omega_{1j}^2(x_1-c_{1j})^2+\cdots+\omega_{nj}^2(x_n-c_{nj})^2]}. \tag{3}$$

For ease of notation, we define vector $\boldsymbol{c}$ and $\boldsymbol{\omega}$ collecting all centers and inverse radii of fuzzy basis functions as

$$\begin{aligned}\boldsymbol{c} &= [c_{11} \;\cdots\; c_{n1} \;\; c_{12} \;\cdots\; c_{n2} \;\cdots\; c_{1r} \;\cdots\; c_{nr}]^{\mathrm{T}},\\ \boldsymbol{\omega} &= [\omega_{11} \;\cdots\; \omega_{n1} \;\; \omega_{12} \;\cdots\; \omega_{n2} \;\cdots\; \omega_{1r} \;\cdots\; \omega_{nr}]^{\mathrm{T}}.\end{aligned} \tag{4}$$

The output $\boldsymbol{f}$ of FBF network can be represented in a vector form

$$\boldsymbol{f}(\boldsymbol{x}(t),\boldsymbol{c},\boldsymbol{\omega},\boldsymbol{W}) = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}(t),\boldsymbol{c},\boldsymbol{\omega}), \tag{5}$$

where $\boldsymbol{W}^{\mathrm{T}} = [w_{jk}]$ is an $r \times m$ matrix and $\boldsymbol{\phi} = [\phi_1 \;\cdots\; \phi_r]^{\mathrm{T}}$. It has been proven in [16] that for

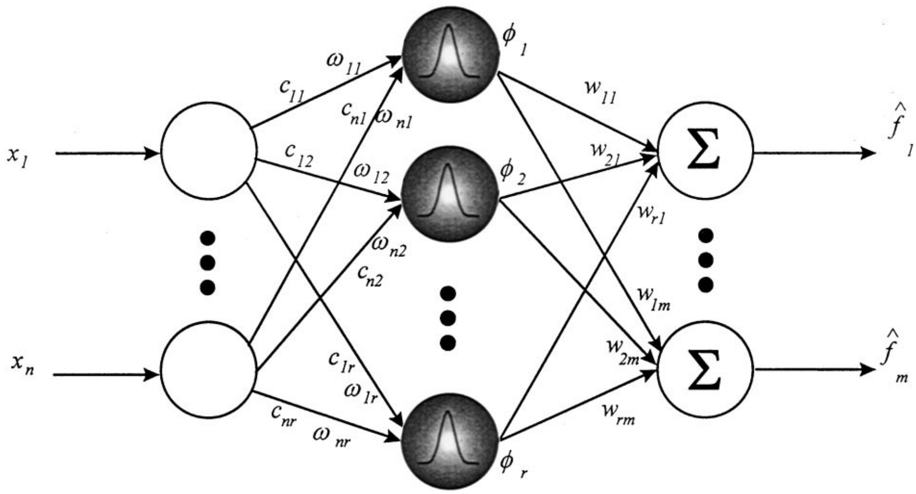


Fig. 1. Network representation of an FBF expansion system.

any given real function $\boldsymbol{f}$ over the input space X, there exists a fuzzy system in the fuzzy basis function expansion form of (5) such that it can uniformly approximate $\boldsymbol{f}$ on the compact set X to arbitrary accuracy. Accordingly, let r be the rule number of the FBF network, there exist an ideal matrix $\boldsymbol{W}^*$, and ideal vectors $\boldsymbol{\omega}^*$ and $\boldsymbol{c}^*$ such that

$$\boldsymbol{f}(\boldsymbol{x}(t)) = \boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}(t), \boldsymbol{c}^*, \boldsymbol{\omega}^*) + \boldsymbol{\varepsilon_r}(\boldsymbol{x}(t)). \tag{6}$$

We employ an FBF network $\hat{\boldsymbol{f}}$ to approximate $\boldsymbol{f}$

$$\hat{\boldsymbol{f}} = \hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}(t), \hat{\boldsymbol{c}}, \hat{\boldsymbol{\omega}}) \tag{7}$$

with $\hat{\boldsymbol{c}}$, $\hat{\boldsymbol{\omega}}$, and $\hat{\boldsymbol{W}}$ of the FBF network estimating $\boldsymbol{\omega}^*, \boldsymbol{c}^*$ and $\boldsymbol{W}^*$. For notational convenience, we denote $\boldsymbol{\phi}^* = \boldsymbol{\phi}(\boldsymbol{x}(t), \boldsymbol{c}^*, \boldsymbol{\omega}^*)$ and $\hat{\boldsymbol{\phi}} = \boldsymbol{\phi}(\boldsymbol{x}(t), \hat{\boldsymbol{c}}, \hat{\boldsymbol{\omega}})$ as

$$\phi_j^* = \exp\{-[\omega_{1j}^{*2}(x_1 - c_{1j}^*)^2 + \cdots + \omega_{nj}^{*2}(x_n - c_{nj}^*)^2]\}$$

and

$$\hat{\phi}_j = \exp\{-[\hat{\omega}_{1j}^2(x_1 - \hat{c}_{1j})^2 + \cdots + \hat{\omega}_{nj}^2(x_n - \hat{c}_{nj})^2]\}.$$

In this paper, the rule's format of an FBF network is represented as [10]

$$\text{If } (c_{1j}, \sigma_{1j}) \text{ and } (c_{2j}, \sigma_{2j}) \text{ and } \ldots \text{ and } (c_{nj}, \sigma_{nj}) \text{ then } (w_{j1}, \ldots, w_{jm}). \tag{8}$$

By defining $\boldsymbol{c}_j = [c_{1j}, \ldots, c_{nj}]^{\mathrm{T}}$ and $\boldsymbol{\sigma}_j = [\sigma_{1j}, \ldots, \sigma_{nj}]^{\mathrm{T}}$ as the center and radius vector of IF part of the jth rule, (8) can be rewritten in the following simpler form:

If $(\boldsymbol{c}_j, \boldsymbol{\sigma}_j)$ then $(w_{j1}, \ldots, w_{jm})$.

An alternative representation of the rule's format is to use the inverse radius vector ω_j instead of the radius vector $\boldsymbol{\sigma}_j$.

In this paper, the parameters of $\mu_{\tilde{A}_{ij}}(x_i)$'s ($c_{ij}$ and σ_{ij}) and w_{jk} are all adjustable and learning rules will be stated in a later section.

3. Robot FBF controller design

3.1. FBF-based controller

In practical robotic systems, the load may vary while performing different tasks, the friction coefficients may change in different configurations and some neglected nolinearities as backlash may appear as disturbances at control inputs, that is, the robot manipulator may receive unpredictable interference from the environment where it resides [2]. Therefore, the control objective is to design a robust FBF-based controller so that the movement of robot arms follow the desired trajectory and all signals in the closed-loop system are bounded even when exogenous and endogenous perturbations are present. Denote the tracking error vector $\boldsymbol{e}(t)$ and error metric $\boldsymbol{s}(t)$ as

$$\boldsymbol{e}(t) = \boldsymbol{\theta}_d(t) - \boldsymbol{\theta}(t), \quad \boldsymbol{s}(t) = \dot{\boldsymbol{e}}(t) + \boldsymbol{\Lambda}\boldsymbol{e}(t), \tag{9}$$

where $\boldsymbol{\theta}_d(t)$, is the desired robot manipulator trajectory vector and $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^{\mathrm{T}} > 0$. Therefore, differentiating $\boldsymbol{s}(t)$ and using (9), the dynamics of robot arms can be rewritten as

$$\boldsymbol{M}\dot{\boldsymbol{s}} = -\boldsymbol{V}_m\boldsymbol{s} + \boldsymbol{f} + \boldsymbol{\tau}_d - \boldsymbol{\tau} \tag{10}$$

where the unknown nonlinear function $\boldsymbol{f}$ as

$$\boldsymbol{f} = \boldsymbol{M}(\boldsymbol{\theta})(\ddot{\boldsymbol{\theta}}_d + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}) + \boldsymbol{V}_m(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})(\boldsymbol{\theta}_d + \boldsymbol{\Lambda}\boldsymbol{e}) + \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{F}\dot{\boldsymbol{\theta}}. \tag{11}$$

Define the control law as

$$\boldsymbol{\tau} = \boldsymbol{K}\boldsymbol{s} + \hat{\boldsymbol{f}} + \hat{\boldsymbol{d}} \tag{12}$$

where $\boldsymbol{K} = \boldsymbol{K}^{\mathrm{T}} > 0$, the output vector of fuzzy basis function networks $\hat{\boldsymbol{f}}$ estimates $\boldsymbol{f}$ and $\hat{\boldsymbol{d}}$ is the robustifying term to attenuate exogenous and endogenous disturbances. The architecture of the closed-loop system is shown in Fig. 2. Using the control in (12), we get closed-loop dynamics as

$$\boldsymbol{M}\dot{\boldsymbol{s}} = -(\boldsymbol{K} + \boldsymbol{V}_m)\boldsymbol{s} + \tilde{\boldsymbol{f}} + \boldsymbol{\tau}_d - \hat{\boldsymbol{d}} \tag{13}$$

where the approximation error $\tilde{\boldsymbol{f}}$ is denoted as

$$\tilde{\boldsymbol{f}} = \boldsymbol{f} - \hat{\boldsymbol{f}} = \boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}(t), \boldsymbol{c}^*, \boldsymbol{\omega}^*) - \hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}(t), \hat{\boldsymbol{c}}, \hat{\boldsymbol{\omega}}) + \boldsymbol{\varepsilon}_r. \tag{14}$$

For simplicity of discussion, we define $\boldsymbol{\phi}^* = \boldsymbol{\phi}(\boldsymbol{x}(t), \boldsymbol{c}^*, \boldsymbol{\omega}^*)$, $\hat{\boldsymbol{\phi}} = \boldsymbol{\phi}(\boldsymbol{x}(t), \hat{\boldsymbol{c}}, \hat{\boldsymbol{\omega}})$ and $\tilde{\boldsymbol{\phi}} = \boldsymbol{\phi}^* - \hat{\boldsymbol{\phi}}$ to obtain a rewritten form of (14)

$$\tilde{\boldsymbol{f}} = \boldsymbol{W}^{*\mathrm{T}}\tilde{\boldsymbol{\phi}} + \tilde{\boldsymbol{W}}^{\mathrm{T}}\hat{\boldsymbol{\phi}} + \boldsymbol{\varepsilon}_r, \tag{15}$$

where $\tilde{\boldsymbol{W}} = \boldsymbol{W}^* - \hat{\boldsymbol{W}}$. In this paper, a method is proposed to guarantee the closed-loop stability and the tracking performance, and on-line tune centers and radii of fuzzy basis functions. For achieving the goal, linearization technique is employed to transform the nonlinear fuzzy basis functions into partially linear form so that Lyapunov theorem extension can be applied. Therefore, take the expansion of $\tilde{\boldsymbol{\phi}}$ in a Taylor series to obtain

$$\tilde{\boldsymbol{\phi}} = \begin{bmatrix} \tilde{\phi}_1 \\ \vdots \\ \tilde{\phi}_r \end{bmatrix} = \begin{bmatrix} \frac{\partial \phi_1}{\partial \boldsymbol{\omega}} \\ \vdots \\ \frac{\partial \phi_r}{\partial \boldsymbol{\omega}} \end{bmatrix}\Bigg|_{\omega=\hat{\omega}} \tilde{\boldsymbol{\omega}} + \begin{bmatrix} \frac{\partial \phi_1}{\partial \boldsymbol{c}} \\ \vdots \\ \frac{\partial \phi_r}{\partial \boldsymbol{c}} \end{bmatrix}\Bigg|_{c=\hat{c}} \tilde{\boldsymbol{c}} + \boldsymbol{h}$$

or

$$\tilde{\boldsymbol{\phi}} = \boldsymbol{A}^{\mathrm{T}}\tilde{\boldsymbol{\omega}} + \boldsymbol{B}^{\mathrm{T}}\tilde{\boldsymbol{c}} + \boldsymbol{h}, \tag{16}$$

where

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega}^* - \hat{\boldsymbol{\omega}}, \quad \tilde{\boldsymbol{c}} = \boldsymbol{c}^* - \hat{\boldsymbol{c}},$$

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial \phi_1}{\partial \boldsymbol{\omega}} & \cdots & \frac{\partial \phi_r}{\partial \boldsymbol{\omega}} \end{bmatrix}\Bigg|_{\omega=\hat{\omega}},$$

$$\boldsymbol{B} = \begin{bmatrix} \frac{\partial \phi_1}{\partial \boldsymbol{c}} & \cdots & \frac{\partial \phi_r}{\partial \boldsymbol{c}} \end{bmatrix}\Bigg|_{c=\hat{c}},$$

$\boldsymbol{h}$ is a vector of higher-order terms and

$$\frac{\partial \phi_j}{\partial \boldsymbol{\omega}} \quad \text{and} \quad \frac{\partial \phi_j}{\partial \boldsymbol{c}}$$

are defined as

$$\left[\frac{\partial \phi_j}{\partial \boldsymbol{\omega}}\right]^{\mathrm{T}} = \Bigg[\underbrace{0 \ \cdots \ 0}_{(j-1)\times n} \ \ \frac{\partial \phi_j}{\partial \omega_{1j}} \ \cdots \ \frac{\partial \phi_j}{\partial \omega_{nj}} \ \ \underbrace{0 \ \cdots \ 0}_{(r-j)\times n}\Bigg]$$

$$\left[\frac{\partial \phi_j}{\partial \boldsymbol{c}}\right]^{\mathrm{T}} = \Bigg[\underbrace{0 \ \cdots \ 0}_{(j-1)\times n} \ \ \frac{\partial \phi_j}{\partial c_{1j}} \ \cdots \ \frac{\partial \phi_j}{\partial c_{nj}} \ \ \underbrace{0 \ \cdots \ 0}_{(r-j)\times n}\Bigg]. \tag{17}$$

Substituting (16) into (15) yields

$$\begin{aligned}\tilde{\boldsymbol{f}} + \boldsymbol{\tau}_d &= \tilde{\boldsymbol{W}}^{\mathrm{T}}\hat{\boldsymbol{\phi}} + \hat{\boldsymbol{W}}^{\mathrm{T}}\tilde{\boldsymbol{\phi}} + \tilde{\boldsymbol{W}}^{\mathrm{T}}\tilde{\boldsymbol{\phi}} + \boldsymbol{\varepsilon}_r + \boldsymbol{\tau}_d \\ &= \tilde{\boldsymbol{W}}^{\mathrm{T}}\hat{\boldsymbol{\phi}} + \hat{\boldsymbol{W}}^{\mathrm{T}}\left[\boldsymbol{A}^{\mathrm{T}}\tilde{\boldsymbol{\omega}} + \boldsymbol{B}^{\mathrm{T}}\tilde{\boldsymbol{c}}\right] + \boldsymbol{d},\end{aligned} \tag{18}$$

where $\boldsymbol{d} = \tilde{\boldsymbol{W}}^{\mathrm{T}}\tilde{\boldsymbol{\phi}} + \hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{h} + \boldsymbol{\varepsilon}_r + \boldsymbol{\tau}_d$. By substituting (18) into (13), the closed-loop system dynamics can be rewritten as

$$\boldsymbol{M}\dot{\boldsymbol{s}} = -(\boldsymbol{K} + \boldsymbol{V}_m)\boldsymbol{s} + \tilde{\boldsymbol{W}}^{\mathrm{T}}\hat{\boldsymbol{\phi}} + \hat{\boldsymbol{W}}^{\mathrm{T}}\left[\boldsymbol{A}^{\mathrm{T}}\tilde{\boldsymbol{\omega}} + \boldsymbol{B}^{\mathrm{T}}\tilde{\boldsymbol{c}}\right] + \tilde{\boldsymbol{d}}, \tag{19}$$

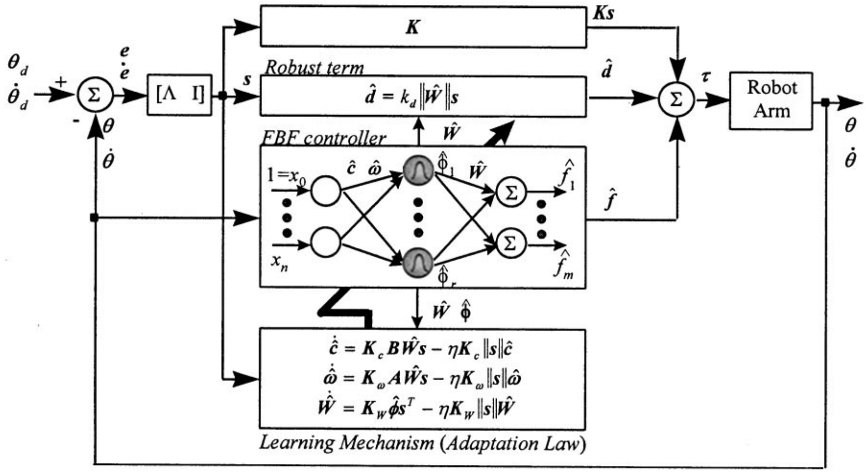


Fig. 2. The diagram of a closed-loop system.

where $\tilde{\boldsymbol{d}} = \boldsymbol{d} - \hat{\boldsymbol{d}}$. Before defining the robustifying term $\hat{\boldsymbol{d}}$, there are some assumptions required to hold in the following discussion.

Assumption 1. The norms of optimal weights, $\|\boldsymbol{W}^*\|$, $\|\boldsymbol{\omega}^*\|$ and $\|\boldsymbol{c}^*\|$, are bounded by known positive real values, i.e., $\|\boldsymbol{W}^*\| \leqslant W_m$, $\|\boldsymbol{\omega}^*\| \leqslant \omega_m$ and $\|\boldsymbol{c}^*\| \leqslant c_m$ with some known W_m, ω_m and c_m (the norm of a vector or matrix in this paper, $\| \bullet \|$, is the Frobenius norm [17]).

For simplicity, we define an optimal matrix $\boldsymbol{\Theta}^*$ including all optimal weights and an estimating matrix $\hat{\boldsymbol{\Theta}}$ as

$$\boldsymbol{\Theta}^* = \begin{bmatrix} \boldsymbol{W}^* & 0 & 0 \\ 0 & \boldsymbol{\omega}^* & 0 \\ 0 & 0 & \boldsymbol{c}^* \end{bmatrix} \text{ and } \hat{\boldsymbol{\Theta}} = \begin{bmatrix} \hat{\boldsymbol{W}} & 0 & 0 \\ 0 & \hat{\boldsymbol{\omega}} & 0 \\ 0 & 0 & \hat{\boldsymbol{c}} \end{bmatrix}.$$

From the above assumption, $\boldsymbol{\Theta}^*$ is bounded by a known positive real value $\Theta_m (\|\boldsymbol{\Theta}^*\| \leqslant \Theta_m)$. Obviously, $\boldsymbol{W}^*, \boldsymbol{\omega}^*$, and $\boldsymbol{c}^*$ are bounded by $\|\boldsymbol{W}^*\| \leqslant \Theta_m$, $\|\boldsymbol{\omega}^*\| \leqslant \Theta_m$ and $\|\boldsymbol{c}^*\| \leqslant \Theta_m$, respectively.

Assumption 2. The approximation errors and disturbances are bounded, i.e., specified b_ε and b_τ satisfying $\|\boldsymbol{\varepsilon}_r\| \leqslant b_\varepsilon$ and $\|\boldsymbol{\tau}_d\| \leqslant b_\tau$, respectively.

Assumption 3. The vector of higher order terms in (16), $\boldsymbol{h}$, is bounded by $\|\boldsymbol{h}\| \leqslant k_d$.

Assumption 4. Since the values of fuzzy basis functions are positive and not greater than one, $\tilde{\boldsymbol{\phi}}$ is bounded by $\|\tilde{\boldsymbol{\phi}}\| \leqslant 1$. Therefore, $\tilde{\boldsymbol{W}}^{\mathrm{T}} \tilde{\boldsymbol{\phi}}$ is bounded by $\|\tilde{\boldsymbol{W}}^{\mathrm{T}} \tilde{\boldsymbol{\phi}}\| \leqslant \|\tilde{\boldsymbol{W}}^{\mathrm{T}}\| \|\tilde{\boldsymbol{\phi}}\| = \|\tilde{\boldsymbol{W}}\| \leqslant \|\tilde{\boldsymbol{\Theta}}\|$.

Based on the above assumptions, we can find the bound of $\boldsymbol{s}^{\mathrm{T}} \boldsymbol{d}$ as

$$\begin{aligned} \|\boldsymbol{s}^{\mathrm{T}} \boldsymbol{d}\| &\leqslant \|\boldsymbol{s}\| \|\tilde{\boldsymbol{W}}^{\mathrm{T}} \tilde{\boldsymbol{\phi}} + \hat{\boldsymbol{W}}^{\mathrm{T}} \boldsymbol{h} + \boldsymbol{\varepsilon}_r + \boldsymbol{\tau}_d\| \\ &\leqslant \|\boldsymbol{s}\| \|\tilde{\boldsymbol{\Theta}}\| + \|\boldsymbol{s}\| \|\hat{\boldsymbol{W}}\| \|\boldsymbol{h}\| + \|\boldsymbol{s}\| (b_\tau + b_\varepsilon). \end{aligned} \tag{20}$$

The robustifying term $\hat{\boldsymbol{d}}$ eliminating the partial bound of $\boldsymbol{d}$ is denoted as

$$\hat{\boldsymbol{d}} = k_d \|\hat{\boldsymbol{W}}\| \boldsymbol{s} \tag{21}$$

where $k_d \geqslant \|\boldsymbol{h}\|$ is assumed to be satisfied. The parameters are updated by the following learning rules:

$$\begin{aligned} \dot{\hat{\boldsymbol{W}}} &= \boldsymbol{K}_W \hat{\boldsymbol{\phi}} \boldsymbol{s}^{\mathrm{T}} - \eta \boldsymbol{K}_W \|\boldsymbol{s}\| \hat{\boldsymbol{W}}, \\ \dot{\hat{\boldsymbol{\omega}}} &= \boldsymbol{K}_\omega \boldsymbol{A} \hat{\boldsymbol{W}} \boldsymbol{s} - \eta \boldsymbol{K}_\omega \|\boldsymbol{s}\| \hat{\boldsymbol{\omega}}, \\ \dot{\hat{\boldsymbol{c}}} &= \boldsymbol{K}_c \boldsymbol{B} \hat{\boldsymbol{W}} \boldsymbol{s} - \eta \boldsymbol{K}_c \|\boldsymbol{s}\| \hat{\boldsymbol{c}}, \end{aligned} \tag{22}$$

where $\boldsymbol{K}_W, \boldsymbol{K}_\omega$, and $\boldsymbol{K}_c$ are diagonal positive square matrices and η is a positive real value. The first terms of (22) are similar to the modified back-propagation algorithm that can tune weights and gains of nodes (neurons) [5]. As to the last terms of (22), they are similar to the e-modification of adaptive control theory [9]. The stability proof will be stated later.

3.2. Stability analysis

In adaptive control, the phenomenon of the possible unboundedness of weight estimates will occur when the persistency of excitation (PE) condition fails to hold. There are some techniques as σ-modification and e-modification can overcome this problem [9]. In [6], a weight tuning rule for neural networks is proposed to guarantee the boundedness of weight estimates even though PE does not hold. A proof being similar to the proof of [6] is to show that the control scheme with learning rules (22) can guarantee the boundedness of all signals generated in the closed-loop system without making any assumptions of PE conditions.

Theorem 1. *Suppose that the vector $\boldsymbol{\theta_d}(t)$ is bounded and Assumptions* 1 *and* 2 *hold. Consider the dynamic equations* (1) *with the control law* (12) *and learning rules* (22). *Make no assumptions of any sort of PE conditions on $\hat{\boldsymbol{\phi}}$, $\boldsymbol{A}\hat{\boldsymbol{W}}$ and $\boldsymbol{B}\hat{\boldsymbol{W}}$. Then*
(1) *the error metric $\boldsymbol{s}(t)$ and weights $\hat{\boldsymbol{c}}$, $\hat{\boldsymbol{\omega}}$ and $\hat{\boldsymbol{W}}$* (or $\hat{\boldsymbol{\Theta}}$) *will remain uniformly ultimately bounded* (*UUB*) *and*
(2) *the tracking errors will be kept as small as desired by increasing $\boldsymbol{K}$.*

Proof. Let the Lyapunov-like function candidate be

$$V(t) = \tfrac{1}{2}(\boldsymbol{s}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{s} + \mathrm{tr}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{K}_W^{-1}\tilde{\boldsymbol{W}}) + \mathrm{tr}(\tilde{\boldsymbol{\omega}}^{\mathrm{T}}\boldsymbol{K}_\omega^{-1}\tilde{\boldsymbol{\omega}}) + \mathrm{tr}(\tilde{\boldsymbol{c}}^{\mathrm{T}}\boldsymbol{K}_c^{-1}\tilde{\boldsymbol{c}})). \tag{23}$$

By the property of skew-symmetry of $\dot{\boldsymbol{M}} - 2\boldsymbol{V_m}$ and (18), the time derivative of $V(t)$ along the trajectories of learning rules (19) and (22) is evaluated by

$$\begin{aligned}\dot{V} &= -\boldsymbol{s}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{s} + \tfrac{1}{2}\boldsymbol{s}^{\mathrm{T}}(\dot{\boldsymbol{M}} - 2\boldsymbol{V_m})\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}(\tilde{\boldsymbol{f}} + \boldsymbol{\tau_d}) \\ &\quad - \boldsymbol{s}^{\mathrm{T}}\hat{\boldsymbol{d}} + \mathrm{tr}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{K}_W^{-1}\dot{\tilde{\boldsymbol{W}}}) \\ &\quad + \mathrm{tr}(\tilde{\boldsymbol{\omega}}^{\mathrm{T}}\boldsymbol{K}_\omega^{-1}\dot{\tilde{\boldsymbol{\omega}}}) + \mathrm{tr}(\tilde{\boldsymbol{c}}^{\mathrm{T}}\boldsymbol{K}_c^{-1}\dot{\tilde{\boldsymbol{c}}}) \\ &= -\boldsymbol{s}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}\tilde{\boldsymbol{W}}^{\mathrm{T}}\hat{\boldsymbol{\phi}} + \boldsymbol{s}^{\mathrm{T}}\hat{\boldsymbol{W}}^{\mathrm{T}}\left[\boldsymbol{A}^{\mathrm{T}}\tilde{\boldsymbol{\omega}} + \boldsymbol{B}^{\mathrm{T}}\tilde{\boldsymbol{c}}\right] \\ &\quad + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{d} - \boldsymbol{s}^{\mathrm{T}}\hat{\boldsymbol{d}} - \mathrm{tr}(\tilde{\boldsymbol{W}}^{\mathrm{T}}(\hat{\boldsymbol{\phi}}\boldsymbol{s}^{\mathrm{T}} - \eta\|\boldsymbol{s}\|\hat{\boldsymbol{W}})) \\ &\quad - \mathrm{tr}(\tilde{\boldsymbol{\omega}}^{\mathrm{T}}(\boldsymbol{A}\hat{\boldsymbol{W}}\boldsymbol{s} - \eta\|\boldsymbol{s}\|\hat{\boldsymbol{\omega}})) \\ &\quad - \mathrm{tr}(\tilde{\boldsymbol{c}}^{\mathrm{T}}(\boldsymbol{B}\hat{\boldsymbol{W}}\boldsymbol{s} - \eta\|\boldsymbol{s}\|\hat{\boldsymbol{c}})).\end{aligned} \tag{24}$$

Using the facts $\mathrm{tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{B}) \leqslant \|\boldsymbol{A}\|\|\boldsymbol{B}\|$ and $\mathrm{tr}(\tilde{\boldsymbol{\Theta}}^{\mathrm{T}}(\boldsymbol{\Theta}^* - \tilde{\boldsymbol{\Theta}})) \leqslant \|\tilde{\boldsymbol{\Theta}}\|\|\boldsymbol{\Theta}^*\| - \|\tilde{\boldsymbol{\Theta}}\|^2$ [2], and applying (20) and (21) results in

$$\begin{aligned}\dot{V} &\leqslant -\boldsymbol{s}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{s} + \eta\|\boldsymbol{s}\|\mathrm{tr}(\tilde{\boldsymbol{\Theta}}^{\mathrm{T}}(\boldsymbol{\Theta}^* - \tilde{\boldsymbol{\Theta}})) \\ &\quad + \|\boldsymbol{s}\|\|\tilde{\boldsymbol{\Theta}}\| + \|\boldsymbol{s}\|\|\hat{\boldsymbol{W}}\|\|\boldsymbol{h}\| + \|\boldsymbol{s}\|(b_\tau + b_\varepsilon) - \boldsymbol{s}^{\mathrm{T}}\hat{\boldsymbol{d}} \\ &\leqslant -\boldsymbol{s}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{s} + \eta\|\boldsymbol{s}\|(\|\tilde{\boldsymbol{\Theta}}\|\|\boldsymbol{\Theta}^*\| - \|\tilde{\boldsymbol{\Theta}}\|^2) \\ &\quad + \|\boldsymbol{s}\|\|\tilde{\boldsymbol{\Theta}}\| + \|\boldsymbol{s}\|\|\hat{\boldsymbol{W}}\|\|\boldsymbol{h}\| \\ &\quad + \|\boldsymbol{s}\|(b_\tau + b_\varepsilon) - k_{\boldsymbol{d}}\|\hat{\boldsymbol{W}}\|\|\boldsymbol{s}\| \\ &\leqslant -\|\boldsymbol{s}\|\{K_{\min}\|\boldsymbol{s}\| + \eta(\|\tilde{\boldsymbol{\Theta}}\| - \boldsymbol{c}_\theta)^2 - D\},\end{aligned} \tag{25}$$

where $K_{\min}$ is the minimum singular value of $\boldsymbol{K}$, $c_\theta = \boldsymbol{\Theta}_m/2 + 1/2\eta$ and $D = b_\tau + b_\varepsilon + \eta c_\theta^2$. Therefore, if $\|\boldsymbol{s}\| > \delta_s$ or $\|\tilde{\boldsymbol{\Theta}}\| > \delta_\theta$, where

$$\delta_s = \frac{D}{K_{\min}} \quad \text{and} \quad \delta_\theta = c_\theta + \frac{D}{\eta}, \tag{26}$$

then $\dot{V} \leqslant 0$. This implies that the Lyapunov derivative $\dot{V}$ is negative outside the compact set ($\|\boldsymbol{s}\| < \delta_s$ or $\|\tilde{\boldsymbol{\Theta}}\| < \delta_\theta$). In other words, outside the compact set given by (26) the tracking errors and parameter errors will decrease. As for inside the compact region around the origin, the tracking errors and parameter errors are bounded. Therefore, according to a standard Lyapunov theorem extension [6], we can prove that $\boldsymbol{s}(t)$ and $\tilde{\boldsymbol{\Theta}}$ are UUB. Since $\boldsymbol{\Theta}^*$ is bounded (Assumption 1), $\hat{\boldsymbol{\Theta}}$ is also UUB. The explicit bound of $\boldsymbol{s}(t)$ is derived in (26) and the bound can be kept as small as desired by increasing $K_{\min}$. □

Remark 1. Without the last terms of (22), the $\hat{\boldsymbol{\phi}}$, $\boldsymbol{A}\hat{\boldsymbol{W}}$ and $\boldsymbol{B}\hat{\boldsymbol{W}}$ should be persistently exciting signals. In other words, positive numbers $T_i, \delta_i, \varepsilon_i$ $(i = 1, 2, 3)$ exist such that given $t \geqslant t_0$, there exists $t_i \in [t, t + T_i]$ such that $[t_i, t_i + \delta_i] \subset [t, t + T_i]$ and

$$\frac{1}{T_i}\int_{t_i}^{t_i+\delta_i} \boldsymbol{g}_i(\tau)\boldsymbol{g}_i(\tau)^{\mathrm{T}}\,\mathrm{d}\tau \geqslant \boldsymbol{\varepsilon}_i\boldsymbol{I} \quad \forall t \geqslant t_0,$$

where $\boldsymbol{g}_1 = \hat{\boldsymbol{\phi}}$, $\boldsymbol{g}_2 = \boldsymbol{A}\hat{\boldsymbol{W}}$ and $\boldsymbol{g}_3 = \boldsymbol{B}\hat{\boldsymbol{W}}$.

Remark 2. It can be found that an implicit parameter η in (26) determines the magnitudes of $\|\boldsymbol{s}\|$ and $\|\tilde{\boldsymbol{\Theta}}\|$. A smaller η will result in a smaller $\|\boldsymbol{s}\|$ and a larger $\|\tilde{\boldsymbol{\Theta}}\|$, and vice versa.

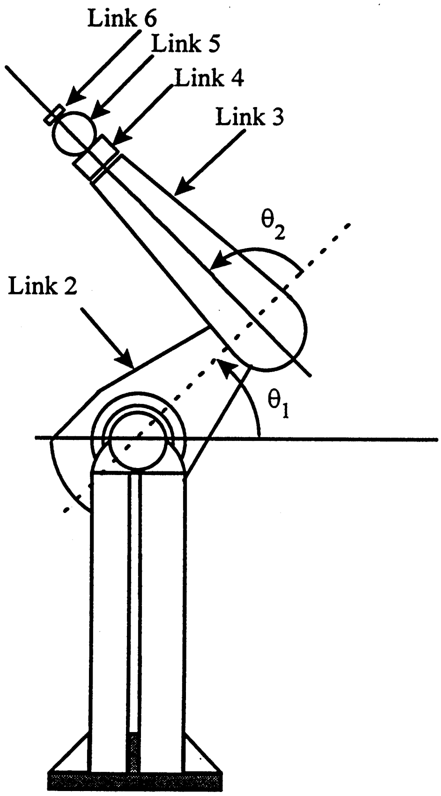


Fig. 3. A two-link robot manipulator with links 4, 5 and 6 fixed.

4. Simulation results

Computer simulations were conducted on the PUMA-560 robot manipulator to verify the availability and performance of the proposed controller. Fig. 3 depicts a 6-link planar robot arm with the fourth, fifth and sixth links fixed to be a two-link robot manipulator. Therefore, the angles of the second and third links were considered to be θ_1 and θ_2, respectively. The numerical values of parameters of the robot model were specified as that in [3]. For demonstrating the tracking performance of our proposed controller, the desired trajectories for θ_1 and θ_2 were set as

$$\theta_{d1} = 0.5 + 0.2(\sin t + \sin 2t) \quad \text{(rad) for } \theta_1$$

and

$$\theta_{d2} = 1.3 - 0.1(\sin t + \sin 2t) \quad \text{(rad) for } \theta_2,$$

respectively.

The proposed FBF controller was compared with the well-known Slotine-Li's adaptive controller. In the well-known Slotine-Li's approach, there are nine parameters to be estimated. As for the proposed method, the task of approximating two nonlinear functions (11) were carried out. The derivative gains of Slotine-Li's method were $K_D = \text{diag}(250, 250)$. For our FBF network, there were 20 rules in the rule base and the parameters of the FBF network were tuning by (22). Each rule has four inputs ($\theta_1, \dot{\theta}_1, \theta_2$ and $\dot{\theta}_2$) and two outputs ($\hat{f}_1$ and $\hat{f}_2$), i.e., there are 200 weights to be tuned. The adaptation rates were specified as $\boldsymbol{K}_W = 100.0\boldsymbol{I}_{r\times r}$, $\boldsymbol{K}_c = 50.0\boldsymbol{I}_{r\times r}$ and $\boldsymbol{K}_\omega = 50.0\boldsymbol{I}_{n\times n}$ ($\boldsymbol{I}_{p\times p}$ is a $p \times p$ identity matrix) and $\eta = 0.01$, and the coefficient of robustifying term $k_d = 100.0$. The initial values of centers $\hat{\boldsymbol{c}}(0)$ and $\hat{\boldsymbol{W}}(0)$ were set to be small random numbers and the inverse radii $\hat{\boldsymbol{\omega}}(0)$ was specified to be 1. Fig. 4 shows the desired trajectories and trajectories obtained from FBF and Slotine-Li's controller. The maximum tracking errors of θ_1 and θ_2 after the first two seconds of movement of the robot arm using the Slotine-Li's method were 0.72° and 0.60°. Using the proposed FBF controller, the maximum errors were found to be 0.42° and 0.06°, respectively. This comparison shows that the proposed controller can obtain more accurate tracking performance due to the good approximation capability of the FBF network as shown in Fig. 5. Fig. 6 shows control inputs with smooth curves. Fig. 7 shows the process of tuning centers and inverse radii of some FBFs. After the tuning process, we found that all 20 rules are located in a reasonable input range with suitable radii. Take one rule as an example:

$$\begin{aligned} &\text{If } (\boldsymbol{c}_1 = (0.2570120, 0.1702516, 0.2400224, \\ &\quad -0.1067455),\ \boldsymbol{\omega}_1 = (0.7214551, 0.7807048, \\ &\quad 0.9510864, 0.9238370)) \\ &\quad \text{then } (w_{11}, w_{12}) = (0.02314668, 0.0031). \end{aligned} \tag{27}$$

Finally, Fig. 8 shows the simulation results with bounded disturbances $\boldsymbol{\tau}_d$ that are 2 Hz square waves with 10 Nm magnitudes. The errors are only slightly larger than that without disturbances. These results imply the robustness of the proposed FBF controller. All these simulations were carried out using C programs on pentium-120 PC and the running time is about 3 min.

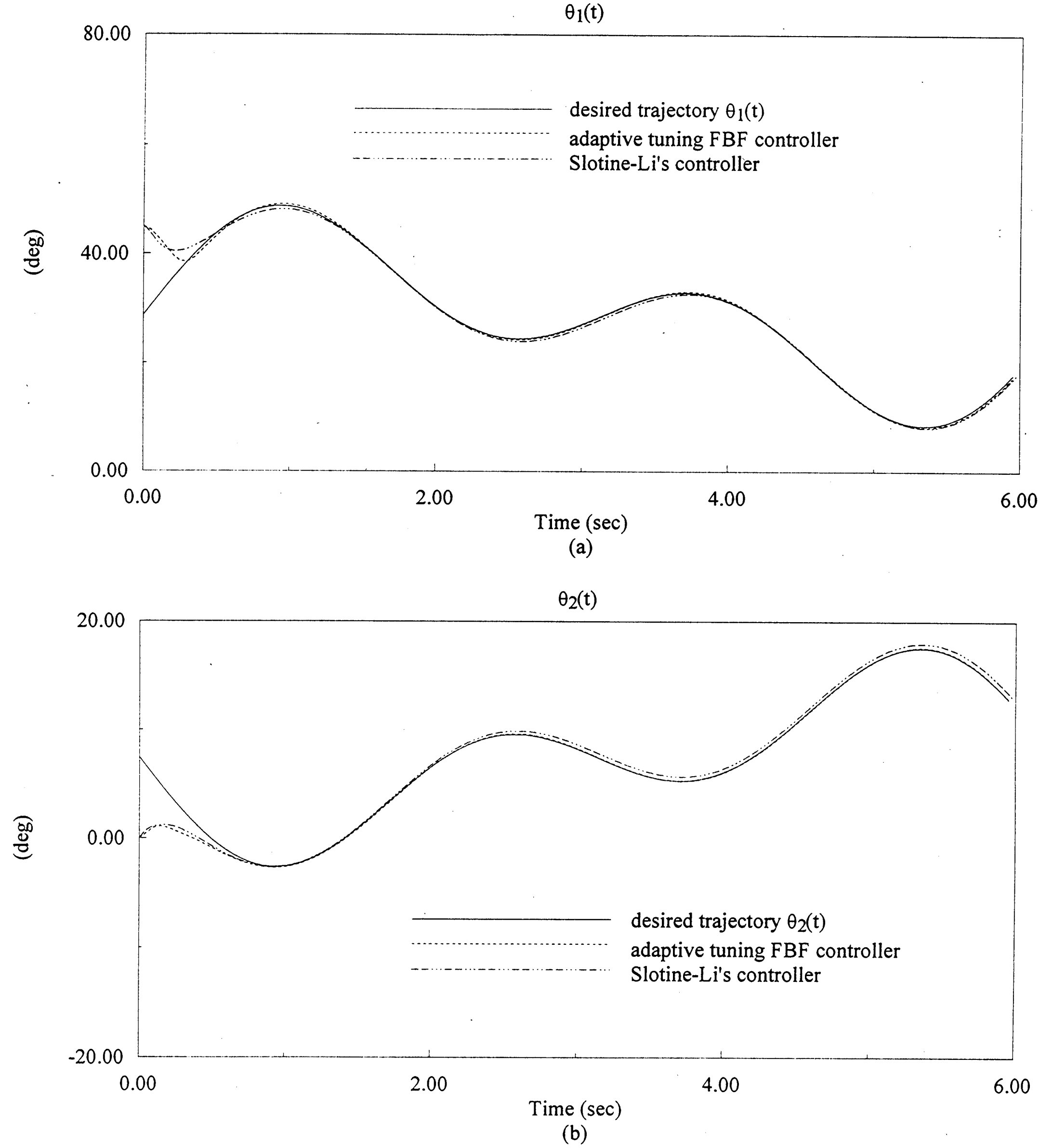


Fig. 4. Simulations for (a) $\theta_1(t)$ and (b) $\theta_2(t)$ using Slotine-Li's and FBF controller.

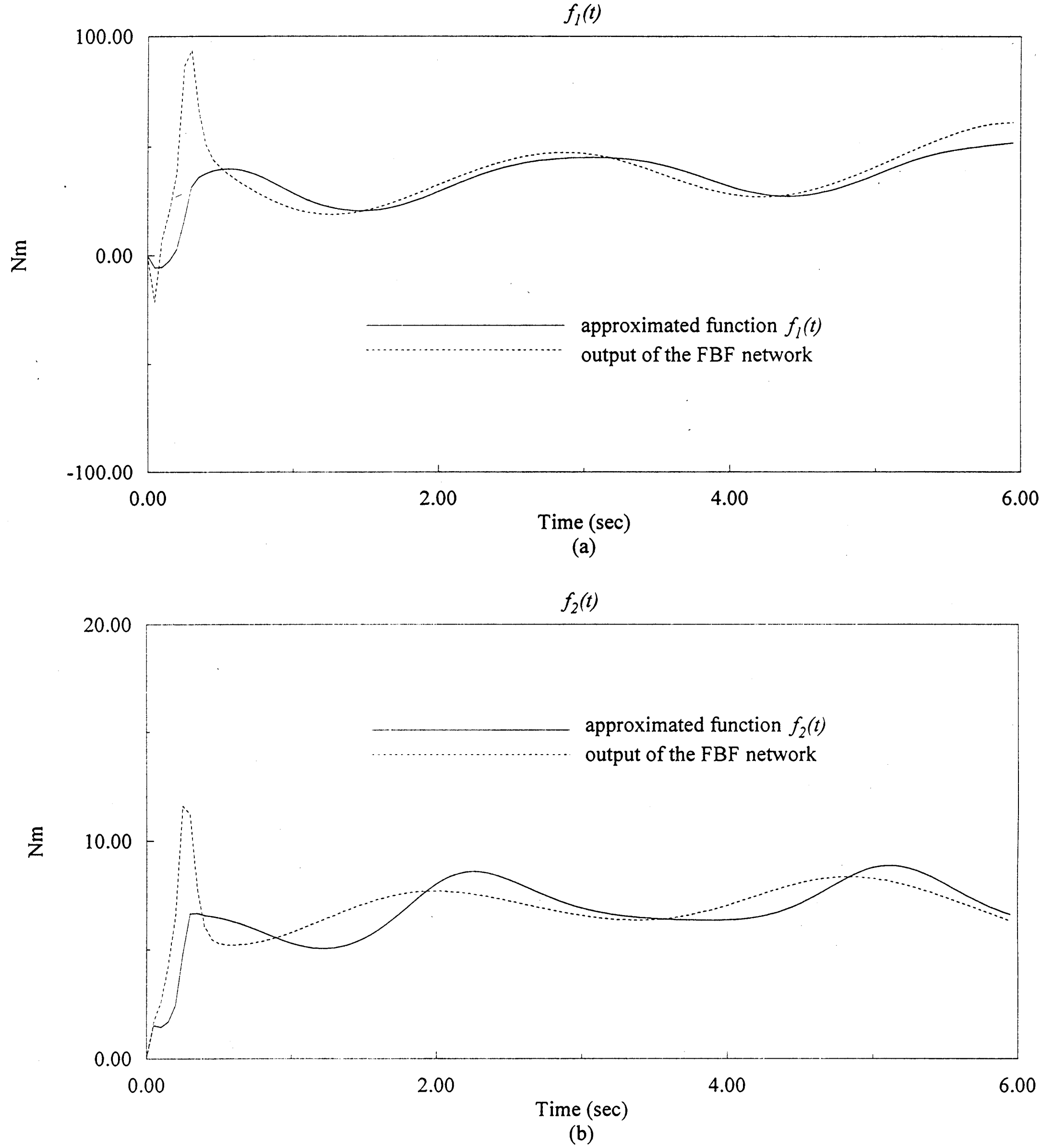
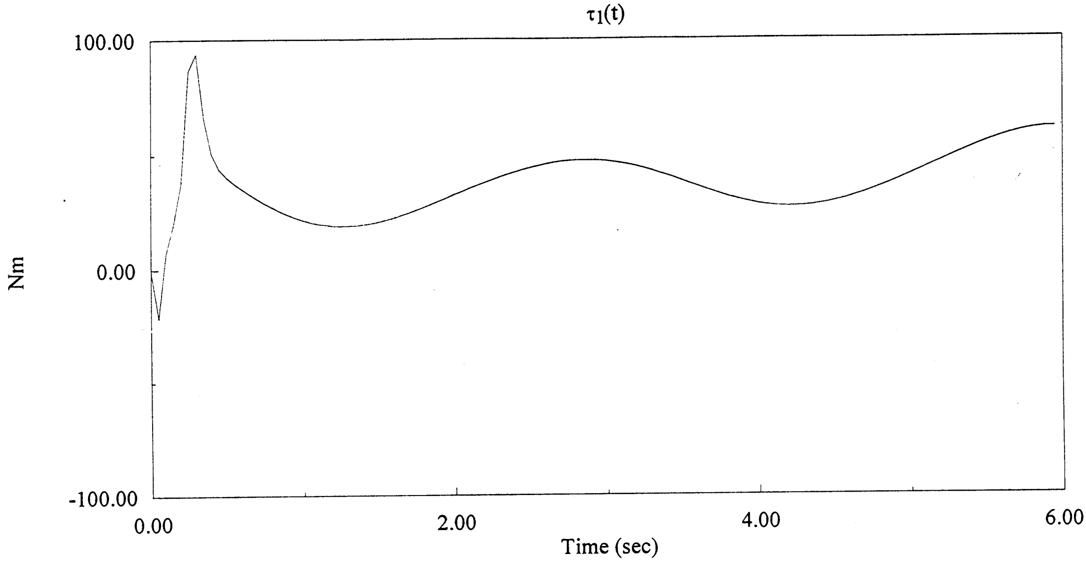
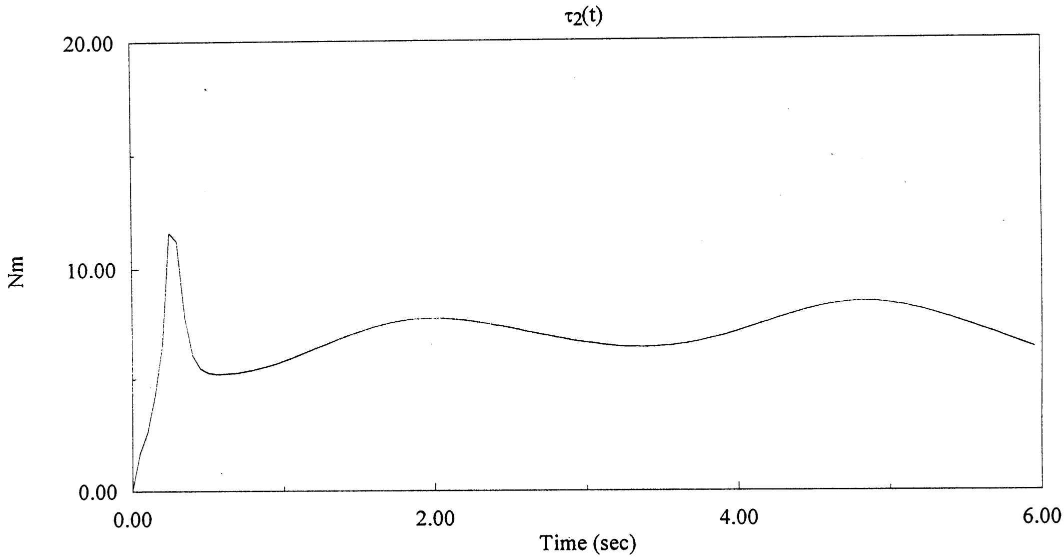


Fig. 5. Function approximation of (a) $f_1(t)$ and (b) $f_2(t)$.



(a)



(b)

Fig. 6. Control inputs: (a) $\tau_1(t)$ and (b) $\tau_2(t)$.

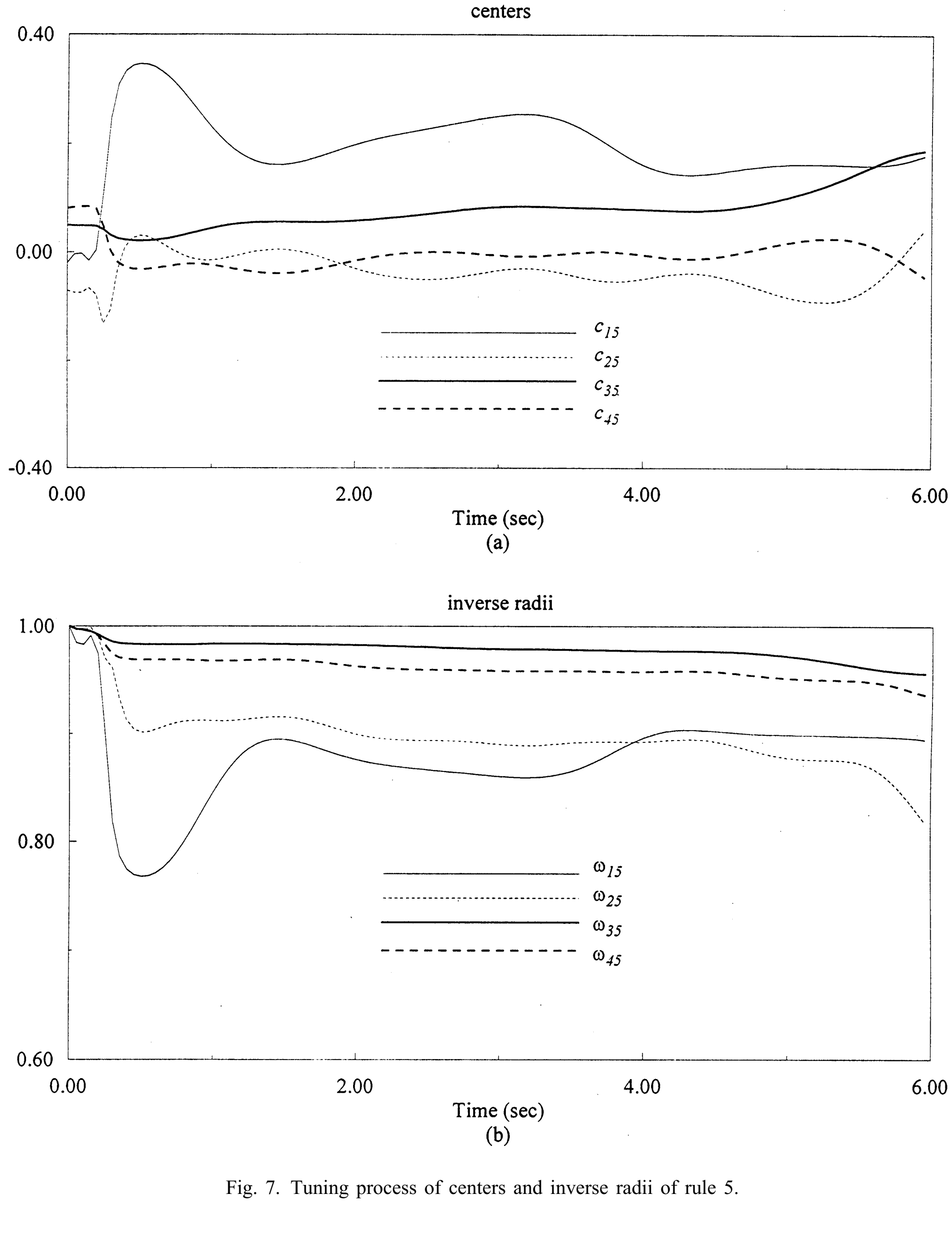
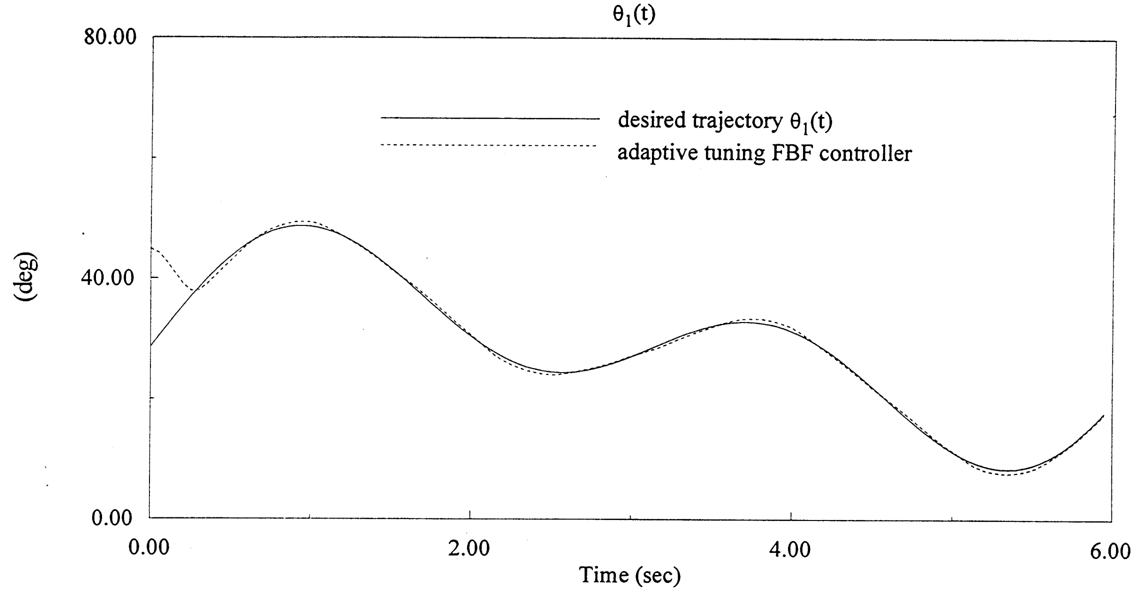
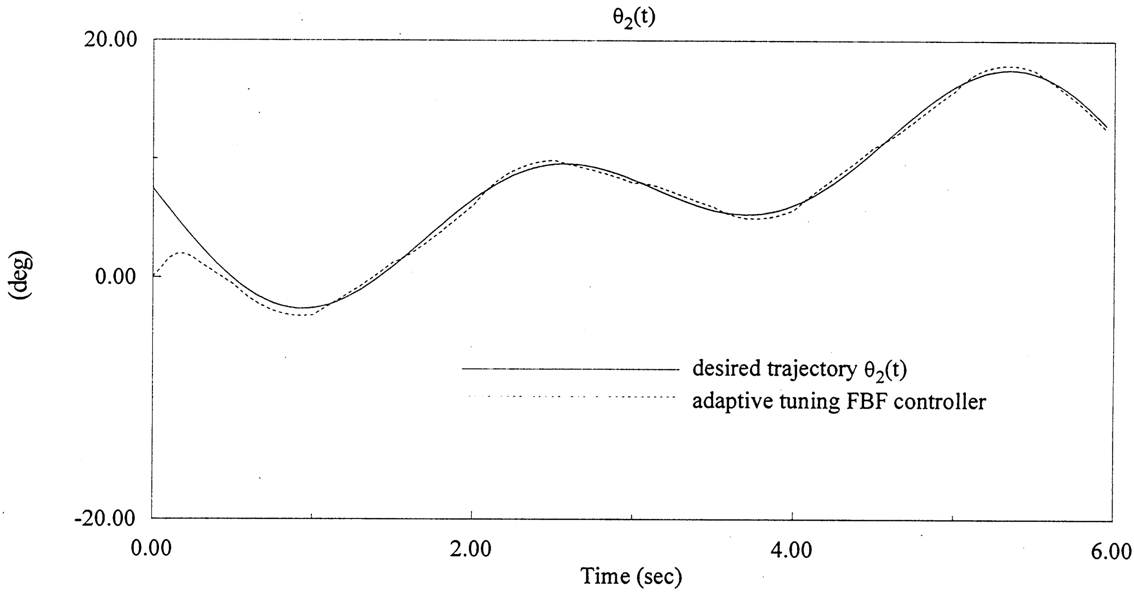


Fig. 7. Tuning process of centers and inverse radii of rule 5.



(a)



(b)

Fig. 8. Simulations for (a) $\theta_1(t)$ and (b) $\theta_2(t)$ with 2 Hz square wave noises.

5. Conclusions

The controller design is based on the FBF network, which is employed to approximate nonlinear functions, and a robust technique. The proposed adaptive tuning FBF-based control system can achieve desired performance as shown in simulation results. The controller is flexible because all parameters of the FBF network can be tuned by weight updating rules once the rule number is determined. By the weight updating rules, we can show all signals in the closed-loop system are bounded without any assumptions of PE conditions to make the controller robust even in the presence of approximation errors and external disturbances.

References

[1] R.G. Berstecher, R. Palm, H. Unbehauen, Construction of a linguistic adaptation law for a fuzzy sliding-mode controller, in: Proc. FUZZ-IEEE'96, 1996, pp. 1794–1799.

[2] B.-S. Chen, T.-S. Lee, J.-H. Feng, A nonlinear H_∞ control design in robotic systems under parameter perturbation and external disturbance, Int. J. Control 59 (1994) 439–461.

[3] M. Erlic, W.-S. Lu, A reduced-order adaptive velocity observer for manipulator control, IEEE Trans. Robotics Automat. 11 (1995) 293–303.

[4] J.-S.R. Jang, Self-learning fuzzy controllers based on temporal back propagation, IEEE Trans. Neural Networks 3 (1992) 714–723.

[5] J.K. Kruschke, J.R. Movellan, Benefits of gain: speeded learning and minimal hidden layers in back-propagation networks, IEEE Trans. Systems Man Cybernet. 21 (1991) 273–280.

[6] F.L. Lewis, K. Liu, A. Yesildirek, Neural net robot controller with guaranteed tracking performance, IEEE Trans. Neural Networks 6 (1995) 703–715.

[7] C.T. Lin, C.S.G. Lee, Reinforcement structure/parameter learning for neural-network-based fuzzy logic control systems, IEEE Trans. Fuzzy Systems 2 (1995) 46–63.

[8] C.-C. Liu, F.-C. Chen, Adaptive control of non-linear continuous-time systems using neural networks – general relative degree and MIMO cases, Int. J. Control 58 (1993) 317–335.

[9] K.S. Narendra, A.M. Annaswamy, Stable Adaptive Systems, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[10] J. Nie, D.A. Linkens, Learning control using fuzzified self-organizing radial basis function network, IEEE Trans. Fuzzy Systems 1 (1993) 280–287.

[11] R.M. Sanner, J.-J.E. Slotine, Gaussian networks for direct adaptive control, IEEE Trans. Neural Networks 3 (1992) 837–863.

[12] J.-J.E. Slotine, W. Li, Adaptive manipulator control: a case study, IEEE Trans. Automat. Control 33 (1988) 995–1003.

[13] J.-J.E. Slotine, W. Li, Applied Nonlinear Control, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[14] C.-Y. Su, Y. Stepanenko, Adaptive fuzzy control of a class of nonlinear systems with fuzzy logic, IEEE Trans. Fuzzy Systems 2 (1994) 285–294.

[15] L.X. Wang, Stable adaptive fuzzy control of nonlinear systems, IEEE Trans. Fuzzy Systems 1 (1993) 146–155.

[16] L.X. Wang, J.M. Mendel, Fuzzy basis functions, universal approximation, and orthogonal least square learning, IEEE Trans. Neural Networks 3 (1992) 807–814.

[17] A. Yesildire, F.L. Lewis, A neural net controller for robots with hebbian tuning and guaranteed tracking, in: Proc. American Control Conf., 1995, pp. 2784–2789.