Output Feedback Control of Nonlinear Systems Using RBF Neural Networks

Sridhar Seshagiri and Hassan K. Khalil, *Fellow, IEEE*

***Abstract*—An adaptive output feedback control scheme for the output tracking of a class of continuous-time nonlinear plants is presented. An RBF neural network is used to adaptively compensate for the plant nonlinearities. The network weights are adapted using a Lyapunov-based design. The method uses parameter projection, control saturation, and a high-gain observer to achieve semi-global uniform ultimate boundedness. The effectiveness of the proposed method is demonstrated through simulations. The simulations also show that by using adaptive control in conjunction with robust control, it is possible to tolerate larger approximation errors resulting from the use of lower order networks.**

***Index Terms*—Adaptive control, output feedback, RBF networks.**

I. Introduction

In recent years, the analytical study of adaptive nonlinear control systems using universal function approximators has received much attention (see [14] for references). Typically, these methods use neural networks as approximation models for the unknown system nonlinearities [2], [4], [5], [9], [10], [14]–[17]. A key assumption in most of these methods is that all the states of the plant are available for feedback.

In [1], Aloliwi and Khalil developed an adaptive output feedback controller for a class of nonlinear systems and pointed out the potential application of their method to linear-in-the-weight neural networks. In this paper, we investigate the use of a radial basis function (RBF) neural network for the purpose. From a mathematical perspective, RBF networks represent just one family in the class of linear in the weight approximators. This class includes, among others, splines, wavelets, certain fuzzy systems and cerebellar model articulation controller (CMAC) networks (see [5] and [6] for references).

Our design is developed for systems represented by input-output models. Augmenting integrators at the input side, the extended system is represented by a state model. RBF networks are used to approximate the system's nonlinearities and the network reconstruction errors[1] contribute to a matched disturbance. The results of [1] show that, provided the upper bound d on the disturbance is small enough, the mean-square tracking error will be of the order $O(\epsilon + d)$, where ϵ is a design parameter. In order to realize small reconstruction errors, it is often necessary to use high-order networks. In [1], the fact that the disturbance satisfies the matching condition is exploited to design an additional robustifying control component for the case when d is not small. The design guarantees that, provided an upper bound on the disturbance is known, the mean square tracking error can be made of the order $O(\epsilon + \mu)$, where both ϵ and μ are design parameters. Thus, it is possible to tolerate larger approximation errors resulting from the use of lower order networks.

II. Problem Statement

We consider a single-input-single-output nonlinear system represented globally by the nth-order differential equation $y^{(n)} = F(\cdot) + G(\cdot)u^{(m)}$ where u is the control input, y is the measured output, $(\cdot)^{(i)}$ denotes the ith derivative of $(\cdot)$, and $m < n$. The functions F and G are smooth functions of y, $y^{(1)}, \cdots, y^{(n-1)}, u, u^{(1)}, \cdots, u^{(m-1)}$. In particular

$$F(\cdot) = F(y, y^{(1)}, \cdots, y^{(n-1)}, u, u^{(1)}, \cdots, u^{(m-1)})$$

and

$$G(\cdot) = G(y, y^{(1)}, \cdots, y^{(n-1)}, u, u^{(1)}, \cdots, u^{(m-1)}).$$

We augment a series of m integrators at the input side of the system and represent the extended system by a state-space model. The states of these integrators are $z_1 = u$, $z_2 = u^{(1)}$, up to $z_m = u^{(m-1)}$ and we set $v = u^{(m)}$ as the control input of the extended system. Taking $x_1 = y$, $x_2 = y^{(1)}$, up to $x_n = y^{(n-1)}$ yields the extended system model

$$\begin{aligned} \dot{x}_i &= x_{i+1}, \qquad 1 \le i \le n-1 \\ \dot{x}_n &= F(x,z) + G(x,z)v \\ \dot{z}_i &= z_{i+1}, \qquad 1 \le i \le m-1 \\ \dot{z}_m &= v \\ y &= x_1 \end{aligned} \tag{1}$$

where $x = [x_1, \cdots, x_n]^T, z = [z_1, \cdots, z_m]^T$.

Assumption 1: $|G(x,z)| \ge k_1 > 0 \forall x \in R^n$ and $z \in R^m$.

Assumption 1 ensures that (1) is input–output linearizable by full state feedback. Using the results of [3], it can be shown that there exists a global diffeomorphism

$$\begin{bmatrix} x \\ \zeta \end{bmatrix} = \begin{bmatrix} x \\ T_1(x,z) \end{bmatrix} \stackrel{\text{def}}{=} T(x,z)$$

[1]The network reconstruction error is defined in Section III.

Manuscript received May 7, 1998; revised May 25, 1999 and September 23, 1999. This work was supported in part by the National Science Foundation under Grants ECS9402187 and ECS9703742.

S. Seshagiri is with the Scientific Research Lab, Ford Motor Company, Dearborn, MI 48121 USA.

H. K. Khalil is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824-1226 USA.

Publisher Item Identifier S 1045-9227(00)00903-6.

with $T_1(0,0)=0$, which transforms the last m state equations of (1) into

$$\dot{\zeta}=H(\zeta,x). \tag{2}$$

This, together with the first n state equations of (1), defines a global normal form. The objective is to design an output feedback controller which guarantees that the output y and its derivatives up to order $n-1$ track a given reference signal y_r and its corresponding derivatives, while keeping all the states bounded. The reference y_r and its derivatives up to order n are assumed to be bounded and $y_r^{(n)}$ is assumed to be piecewise continuous.

III. Function Approximation Using Gaussian Radial Basis Functions

The control design presented in this paper employs an RBF neural network to approximate the functions $F(\cdot)$ and $G(\cdot)$ over a compact region of the state space. RBF networks are of the general form $\hat{F}(\cdot)=\theta^T f(\cdot)$, where $\theta\in R^p$ is a vector of adjustable weights and $f(\cdot)$ a vector of Gaussian basis functions. Their ability to uniformly approximate smooth functions over compact sets is well documented in the literature (see [16] for references). In particular, it has been shown that given a smooth function $F\colon \Omega\mapsto R$, where Ω is a compact subset of R^{m+n} and $\epsilon>0$, there exists a Gaussian basis function vector $f\colon R^{m+n}\mapsto R^p$ and a weight vector $\theta^*\in R^p$ such that $|F(x)-{\theta^*}^T f(x)|\le\epsilon\,\forall x\in\Omega$. The quantity $F(x)-{\theta^*}^T f(x)\overset{\text{def}}{=}d(x)$ is called the **network reconstruction error**.

The optimal weight vector θ^* defined above is a quantity required only for analytical purposes. Typically θ^* is chosen as the value of θ that minimizes $d(x)$ over Ω, that is

$$\theta^*=\arg\min_{\theta\in R^p}\{\sup_{x\in\Omega}|F(x)-\theta^T f(x)|\}. \tag{3}$$

The choice of the Gaussian network parameters used in our control design is motivated by the discussion in [16]. The basis functions are located on a regular grid that contains the subset of interest of the state space. The update law for the weight vector θ is derived in the next section.

IV. Control Design

In this section, we first design an adaptive output feedback controller under the assumption that the network reconstruction errors are "small." Next, the condition of small reconstruction errors is relaxed by adding a robustifying control component to make the mean-square tracking error arbitrarily small. The design of the output feedback controller is done in two steps: first, a state feedback controller is designed; then, the states are replaced by their estimates provided by a high-gain observer. We start with the following representation for the functions $F(\cdot)$ and $G(\cdot)$, valid for all $x\in Y$ and $z\in Z$, where Y and Z are compact sets defined in Section IV-A1

$$\begin{aligned}F(x,z)&=\theta_f^{*T}f(x,z)+d_F(x,z),\\ G(x,z)&=\theta_g^{*T}g(x,z)+d_G(x,z).\end{aligned} \tag{4}$$

Assumption 2: The vectors θ_f^* and θ_g^* belong to known compact subsets $\Omega_f\subset R^{p_1}$ and $\Omega_g\subset R^{p_2}$. Typically, some off-line training is done to obtain values θ_{f_0} and θ_{g_0} that result in "good" approximations of the functions F and G over $Y\times Z$. This can be accomplished, for example, by the standard backpropogation technique [7]. The sets Ω_f and Ω_g are then chosen judiciously as compact sets that contain θ_{f_0} and θ_{g_0}. If we denote the "optimal" reconstruction errors that result from the use of the vectors θ_f^* and θ_g^* by $d_F^*(\cdot)$ and $d_G^*(\cdot)$ respectively, then, in view of the off-line training, it is reasonable to expect that our choice of the sets Ω_f and Ω_g will result in reconstruction errors $d_F(\cdot)$ and $d_G(\cdot)$ that are comparable to $d_F^*(\cdot)$ and $d_G^*(\cdot)$, respectively. Notice also that it is simply possible to choose the sets Ω_f and Ω_g arbitrarily large. However, this would be undesirable from the viewpoint of using parameter projection. The fixed optimal weights θ_f^* and θ_g^* in (3) are replaced by their time varying estimates $\hat{\theta}_f$ and $\hat{\theta}_g$, that are adapted during learning. The network approximations associated with these weights are denoted by $\hat{F}$ and $\hat{G}$, respectively.

Assumption 3: $|\hat{G}(\cdot)|\ge k_2>0\forall x\in Y$, $z\in Z$ and $\hat{\theta}_g\in\hat{\Omega}_g$, where $\hat{\Omega}_g$ is a compact set that contains Ω_g in its interior.

A. Small Reconstruction Error

Under the assumption of small reconstruction errors, we design an adaptive contoller so that the output y tracks the given reference signal y_r. Define

$$e_i=y^{(i-1)}-y_r^{(i-1)}=x_i-y_r^{(i-1)},\qquad 1\le i\le n$$

and

$$e=[e_1,e_2,\cdots,e_n]^T$$

Let

$$\begin{aligned}\mathcal{Y}(t)&=[y(t),y^{(1)}(t),\cdots,y^{(n-1)}(t)]^T\\ \mathcal{Y}_r(t)&=[y_r(t),y_r^{(1)}(t),\cdots,y_r^{(n-1)}(t)]^T\\ \mathcal{Y}_R(t)&=[y_r(t),y_r^{(1)}(t),\cdots,y_r^{(n-1)}(t),y_r^{(n)}(t)]^T\end{aligned}$$

and Y_0 and Y_R be any given compact subsets of R^n and R^{n+1}, respectively, such that $\mathcal{Y}(0)\in Y_0$ and $\mathcal{Y}_R(t)\in Y_R\ \forall t\ge 0$. We rewrite (1) as

$$\begin{aligned}\dot{e}&=A_m e+b\{Ke+\theta_f^{*T}f(e+\mathcal{Y}_r,z)+\theta_g^{*T}g(e+\mathcal{Y}_r,z)v\\ &\quad+d(e+\mathcal{Y}_r,z,\hat{\theta}_f,\hat{\theta}_g)-y_r^{(n)}\}\\ \dot{z}&=A_2z+b_2v\end{aligned} \tag{5}$$

where[2] $d(e+\mathcal{Y}_r,z,\hat{\theta}_f,\hat{\theta}_g)=d_F(\cdot)+d_G(\cdot)v,(A,b)$ and (A_2,b_2) are controllable canonical pairs that represent chains of n and m integrators, respectively, and K is chosen such that $A_m=A-bK$ is Hurwitz.

Assumption 4: The system $\dot{\zeta}=H(\zeta,\mathcal{Y}_r)$ has a unique steady-state solution $\bar{\zeta}$. Moreover, with $\tilde{\zeta}=\zeta-\bar{\zeta}$ the system

$$\begin{aligned}\dot{\tilde{\zeta}}&=H(\bar{\zeta}+\tilde{\zeta},e+\mathcal{Y}_r)-H(\bar{\zeta},\mathcal{Y}_r)\\ &\overset{\text{def}}{=}\tilde{H}(\tilde{\zeta},e,\mathcal{Y}_r,\bar{\zeta})\end{aligned} \tag{6}$$

[2]The dependence on $\hat{\theta}_f$ and $\hat{\theta}_g$ comes through v. See (8).

has a continuously differentiable function $V_1(t, \tilde{\zeta})$ that satisfies[3]

$$\eta_1|\tilde{\zeta}|^2 \leq V_1(t, \tilde{\zeta}) \leq \eta_2|\tilde{\zeta}|^2$$
$$\frac{\partial V_1}{\partial t} + \frac{\partial V_1}{\partial \tilde{\zeta}} \tilde{H}(\tilde{\zeta}, e, \mathcal{Y}_r, \overline{\zeta}) \leq -\eta_3\|\tilde{\zeta}\|^2 + \eta_4\|\tilde{\zeta}\|\|e\|$$

where $\eta_1, \eta_2, \eta_3 >, 0$, and $\eta_4 \geq 0$ are independent of $\mathcal{Y}_r$. The steady-state response of a nonlinear system is introduced in [8]. Basically, it is a particular solution toward which any other solution of the system converges as time increases. The inequalities satisfied by V_1 imply that such convergence is exponential. They also imply that (6), with e as input, is input-to-state stable. Consequently, the zero dynamics of (1) are exponentially stable and (1) is minimum phase.

1) State Feedback: Let $P = P^T > 0$ be the solution of the Lyapunov equation $PA_m + A_m^T P = -Q$ where $Q = Q^T > 0$, and consider the Lyapunov function candidate

$$V = e^T Pe + \tfrac{1}{2}\tilde{\theta}_f^T \Gamma_f^{-1}\tilde{\theta}_f + \tfrac{1}{2}\tilde{\theta}_g^T \Gamma_g^{-1}\tilde{\theta}_g \qquad (7)$$

where $\tilde{\theta}_f = \hat{\theta}_f - \theta_f^*$, $\tilde{\theta}_g = \hat{\theta}_g - \theta_g^*$ and $\Gamma_f = \Gamma_f^T > 0$ and $\Gamma_g = \Gamma_g^T >= 0$ are gains to be specified later. Using (6) the derivative of V along the trajectories of the system is given by

$$\begin{aligned}\dot{V} =& -e^T Qe + \tilde{\theta}_f^T \Gamma_f^{-1}\dot{\hat{\theta}}_f + \tilde{\theta}_g^T \Gamma_g^{-1}\dot{\hat{\theta}}_g \\ &+ 2e^T Pb\{\theta_f^{*T} f(e + \mathcal{Y}_r, z) + \theta_g^{*T} g(e + \mathcal{Y}_r, z)v \\ &+ d(e + \mathcal{Y}_r, z, \hat{\theta}_f, \hat{\theta}_g) + Ke - y_r^{(n)}\}.\end{aligned}$$

Taking

$$\begin{aligned} v &= \frac{-Ke + y_r^{(n)} - \hat{F}(e + \mathcal{Y}_r, z, \hat{\theta}_f)}{\hat{G}(e + \mathcal{Y}_r, z, \hat{\theta}_g)} \\ &\stackrel{\text{def}}{=} \psi(e, z, \mathcal{Y}_R, \hat{\theta}_f, \hat{\theta}_g) \end{aligned} \qquad (8)$$

we can rewrite the expression for $\dot{V}$ as

$$\begin{aligned}\dot{V} =& -e^T Qe + \tilde{\theta}_f^T \Gamma_f^{-1}[\dot{\hat{\theta}}_f - \Gamma_f \phi_f] \\ &+ \tilde{\theta}_g^T \Gamma_g^{-1}[\dot{\hat{\theta}}_g - \Gamma_g \phi_g] + 2e^T Pbd(\cdot)\end{aligned} \qquad (9)$$

where $\phi_f = 2e^T Pbf(e + \mathcal{Y}_r, z)$ and $\phi_g = 2e^T Pbg(e + \mathcal{Y}_r, z)\psi(\cdot)$. Let $\hat{\Omega}_f$ be a compact subset of R^{p_1} that contains Ω_f in its interior. Define

$$\theta = \begin{bmatrix}\theta_f \\ \theta_g\end{bmatrix}, \quad \hat{\theta} = \begin{bmatrix}\hat{\theta}_f \\ \hat{\theta}_g\end{bmatrix}, \quad \tilde{\theta} = \begin{bmatrix}\tilde{\theta}_f \\ \tilde{\theta}_g\end{bmatrix}, \quad \phi = \begin{bmatrix}\phi_f \\ \phi_g\end{bmatrix},$$
$$\Gamma = \text{diag}[\Gamma_f, \Gamma_g], \quad \Omega = \Omega_f \times \Omega_g \quad \text{and} \quad \hat{\Omega} = \hat{\Omega}_f \times \hat{\Omega}_g.$$

The parameter adaptation law is chosen as in [12], i.e, $\dot{\hat{\theta}} = \text{Proj}(\hat{\theta}, \phi)$, where $\text{Proj}(\hat{\theta}, \phi) = \Gamma\phi$ for $\hat{\theta} \in \Omega$ and is modified outside Ω to ensure that

$$\tilde{\theta}^T \Gamma^{-1}[\dot{\hat{\theta}} - \Gamma\phi] \leq 0 \qquad (10)$$

and $\hat{\theta}(t)$ belongs to a compact set $\Omega_\delta \forall t \geq 0$, where $\hat{\Omega} \supset \Omega_\delta \supset \Omega$. As an example of parameter projection, consider the case when Ω is the convex hypercube

$$\Omega = \{\theta_i | a_i \leq \theta_i \leq b_i\}, 1 \leq i \leq p = p_1 + p_2\}.$$

Let

$$\Omega_\delta = \{\theta | a_i - \delta \leq \theta_i \leq b_i + \delta\}, 1 \leq i \leq p\}$$

where $\delta > 0$ is chosen such that $\Omega_\delta \subset \hat{\Omega}$, and choose Γ to be a positive diagonal matrix. In this case the projection $\text{Proj}(\hat{\theta}, \phi)$ is taken as

$$[\text{Proj}(\hat{\theta}, \phi)]_i = \begin{cases} \gamma_{ii}\phi_i, & \text{if } a_i \leq \hat{\theta}_i \leq b_i \text{ or} \\ & \text{if } \hat{\theta}_i > b_i \text{ and } \phi_i \leq 0 \text{ or} \\ & \text{if } \hat{\theta}_i < a_i \text{ and } \phi_i \geq 0 \\ \gamma_{ii}\left[1 + \dfrac{b_i - \hat{\theta}_i}{\delta}\right]\phi_i, & \text{if } \hat{\theta}_i > b_i \text{ and } \phi_i > 0 \\ \gamma_{ii}\left[1 + \dfrac{\hat{\theta}_i - a_i}{\delta}\right]\phi_i, & \text{if } \hat{\theta}_i < a_i \text{ and } \phi_i < 0. \end{cases} \qquad (11)$$

By our design of the RBF network, we seek to impose a bound on $d(\cdot)$ over a compact subset of R^{n+m}. With that goal, we first assume that $e(0)$ and $z(0)$ belong to known compact subsets $E_0 \subset R^n$ and $Z_0 \subset R^m$ and let $c_1 = \max_{e \in E_0} e^T Pe$. Choose $c_4 > c_1$ and define $E \stackrel{\text{def}}{=} \{e^T Pe \leq c_4\}$ and $Y \stackrel{\text{def}}{=} \{e + \mathcal{Y}_r | e \in E, \mathcal{Y}_r \in Y_R\}$. Let Z be a compact subset of R^m such that Z_0 is in the interior of Z and

$$z(0) \in Z_0 \quad \text{and} \quad e(t) \in E \forall t \geq 0 \Rightarrow z(t) \in Z \forall t \geq 0.$$

The set Z can be determined using the Lyapunov function V_1 of Assumption 4. The basic idea is to choose c_z large enough that the set $\{V_1 \leq c_z\}$ is positively invariant, and then determine the corresponding set in the z-coordinate. The RBF networks are used to approximate $F(\cdot)$ and $G(\cdot)$ over the compact set $Y \times Z$. Define Ω_{δ_f} and Ω_{δ_g} by $\Omega_\delta = \Omega_{\delta_f} \times \Omega_{\delta_g}$ and let

$$c_2 = \max_{\theta_f^* \in \Omega_f, \hat{\theta}_f \in \Omega_{\delta_f}} \tfrac{1}{2}(\hat{\theta}_f - \theta_f^{*T})\Gamma_f^{-1}(\hat{\theta}_f - \theta_f^*)$$
$$c_3 = \max_{\theta_g^* \in \Omega_g, \hat{\theta}_g \in \Omega_{\delta_g}} \tfrac{1}{2}(\hat{\theta}_g - \theta_g^{*T})\Gamma_g^{-1}(\hat{\theta}_g - \theta_g^*).$$

The adaptation gains Γ_f and Γ_g are chosen large enough to ensure that $c_4 - c_1 > c_2 + c_3$. This is different from [1] where the adaptation gain is not required to be large. This is because, in [1], the parameter vector θ has some physical meaning and the compact set Ω to which it belongs is known *a priori*. In particular, the definition of the set E implicitly involves the set Ω. In the present case however, the compact sets Ω_f and Ω_g to which the optimal weights θ_f^* and θ_g^* of the neural network belong themselves depend on the set E, because the approximation of F and G is done over the set $Y \times Z$. Hence the set Y has to be defined prior to, and consequently, independent of the sets Ω_f and Ω_g. This requires making the adaptation gains large.

Let $d = \max \|d(e + \mathcal{Y}_r, z, \hat{\theta}_f, \hat{\theta}_g)\|$, where the maximization is done over all $e + \mathcal{Y}_r \in Y, z \in Z, \hat{\theta}_f \in \Omega_{\delta_f}$ and $\hat{\theta}_g \in \Omega_{\delta_g}$. Using (7), (9) and (10), $\forall e \in E$ we have

$$\dot{V} \leq -kV + k(c_2 + c_3) + k_d d \qquad (12)$$

[3]Unless otherwise specified, $\|\cdot\|$ denotes the Euclidean norm.

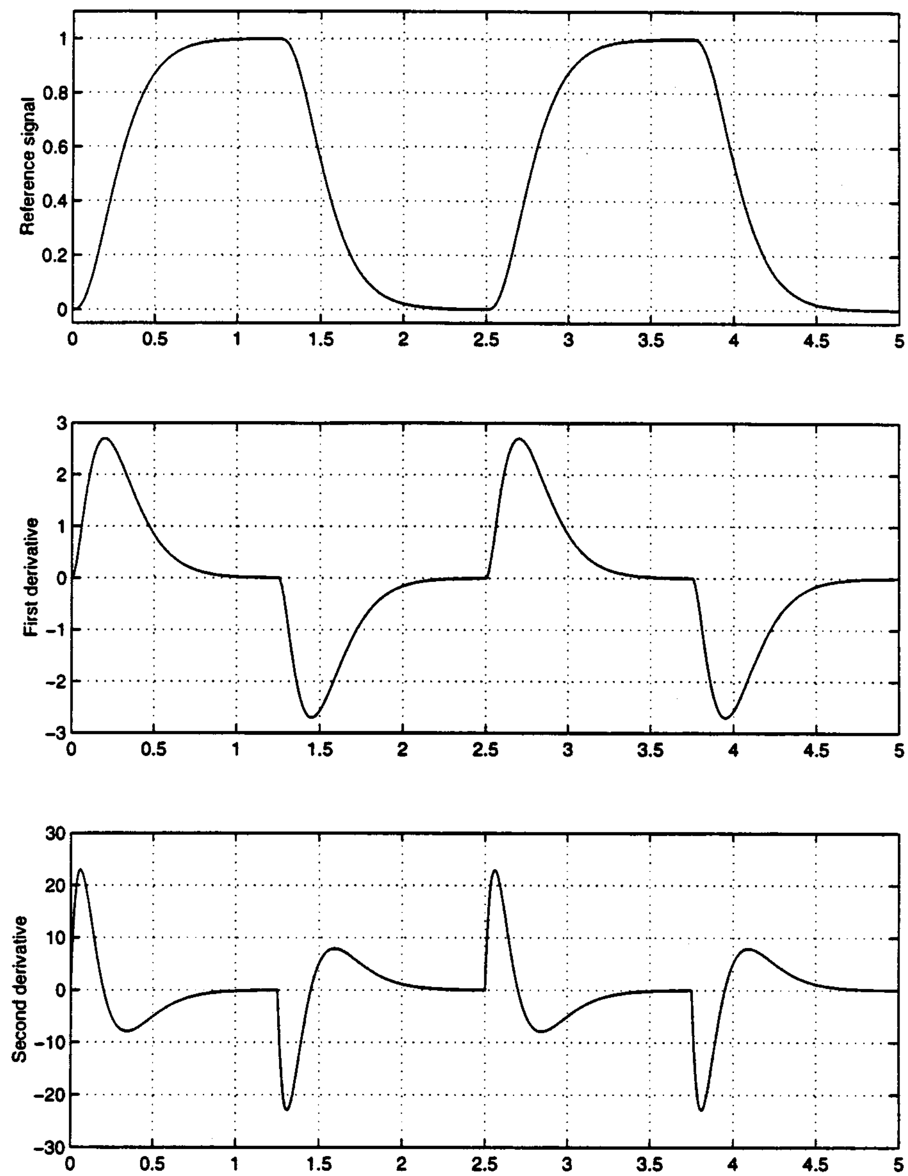


Fig. 1. The reference signal and its derivatives.

where $k = (\lambda_{\min}(Q)/\lambda_{\max}(P))$ and $k_d = \max_{e\in E} 2\|e\|\,\|Pb\|$. If $d < d^* = k(c_4 - c_3 - c_2)k_d$, then $\dot{V} < 0$ on $\{V = c_4\} \cap \Omega_\delta$. Thus the set $\{V \le c_4\} \cap \Omega_\delta$ is positively invariant for all $d < d^*$. Inside this set, $e \in E$. As long as $e \in E$, z will remain in Z. Thus the trajectory $(e, z, \hat{\theta})$ is trapped inside the set $R_s = \{e \in E\} \times \{z \in Z\} \times \{\hat{\theta} \in \Omega_\delta\}$. Hence all the states are bounded and from (12), there exists a $\tau > 0$ such that for all $t \ge \tau$, the tracking error is of the order $O(d + (1/\gamma_f) + (1/\gamma_g))$, where $\gamma_f = \lambda_{\min}(\Gamma_f)$ and $\gamma_g = \lambda_{\min}(\Gamma_g)$.

2) Output Feedback: To implement the controller developed in the previous section using output feedback, we replace the states e by their estimates $\hat{e}$ provided by a high gain observer (HGO). The control is saturated outside a compact region of interest to prevent the peaking induced by the HGO [12]. We assume that $\hat{\theta}_f(0) \in \Omega_f$ and $\hat{\theta}_g(0) \in \Omega_g$. Let $S \ge \max|\psi(e, z, \mathcal{Y}_R, \hat{\theta}_f, \hat{\theta}_g)|$ where the maximization is taken over all $e \in E_1 \stackrel{\text{def}}{=} \{e^T Pe \le c_5\}$, $z \in Z$, $\mathcal{Y}_R \in Y_R$, $\hat{\theta}_f \in \Omega_{\delta_f}$, $\hat{\theta}_g \in \Omega_{\delta_g}$ where $c_5 > c_4$. Define the saturated function ψ^s by

$$\psi^s(e, z, \mathcal{Y}_R, \hat{\theta}_f, \hat{\theta}_g) = S \,\text{sat}\left(\frac{\psi(e, z, \mathcal{Y}_R, \hat{\theta}_f, \hat{\theta}_g)}{S}\right)$$

where $\text{sat}(\cdot)$ is the saturation function. The output feedback controller is taken as $v = \psi^s(\hat{e}, z, \mathcal{Y}_R, \hat{\theta}_f, \hat{\theta}_g)$. The HGO used to estimate the states is the same one used in [12] and is described by the following equations:

$$\begin{aligned} \dot{\hat{e}}_i &= \hat{e}_{i+1} + \frac{\alpha_i}{\epsilon^i}(e_1 - \hat{e}_1), \qquad 1 \le i \le n-1 \\ \dot{\hat{e}}_n &= \frac{\alpha_n}{\epsilon^n}(e_1 - \hat{e}_1) \end{aligned} \tag{13}$$

where $\epsilon > 0$ is a design parameter that will be specified shortly. The positive constants α_i are chosen such that the roots of $s^n +$

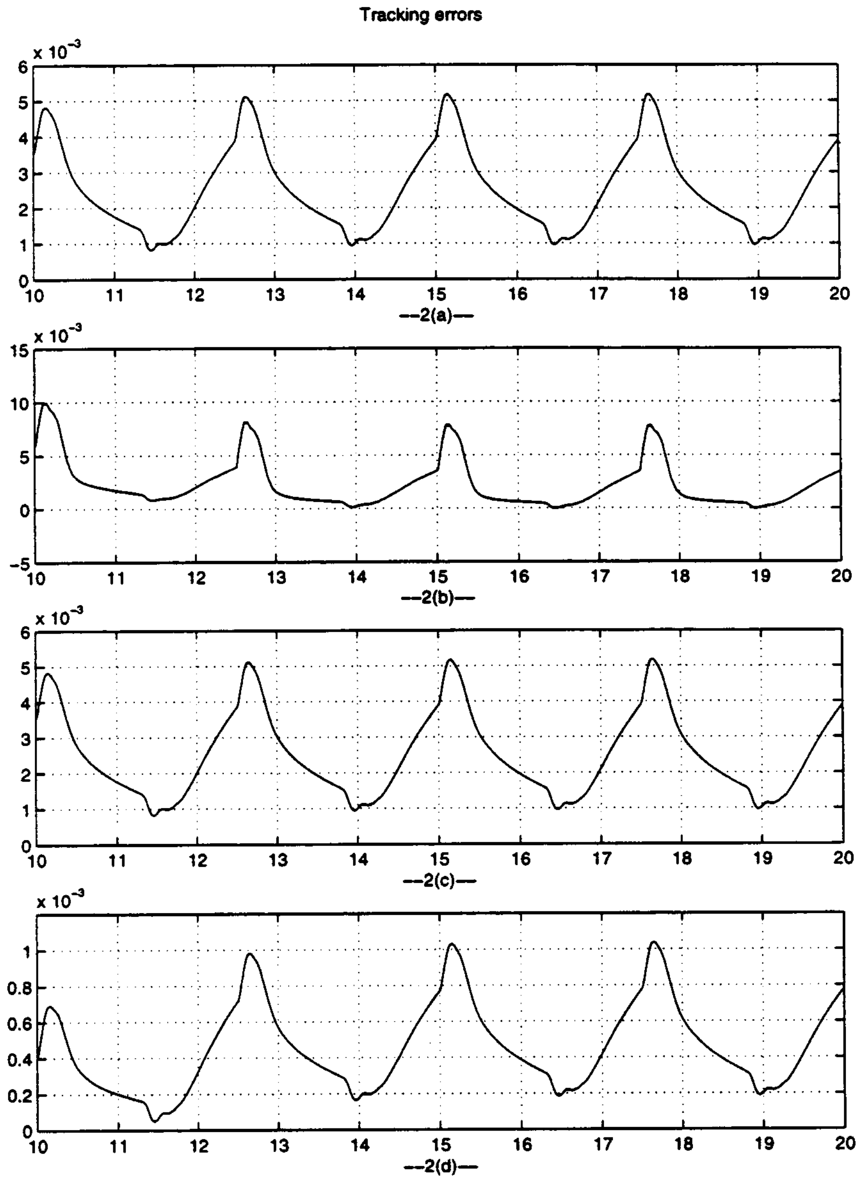


Fig. 2. (a) State feedback, $\mu = 0.5$. (b) Output feedback, $\epsilon = 10^{-3}$, $\mu = 0.5$. (c) Output feedback $\epsilon = 10^{-4}$, $\mu = 0.5$. (d) Output feedback $\epsilon = 10^{-4}$, $\mu = 0.1$..

$\alpha_1 s^{n-1} + \cdots + \alpha_{n-1}s + \alpha_n = 0$ have negative real parts. Let $\xi_i = (e_i - \hat{e}_i/\epsilon^{n-i}), 1 \le i \le n$, $\xi = [\xi_1, \cdots, \xi_n]^T$ and $V_\xi = \xi^T \overline{P} \xi$, where $\overline{P} = \overline{P}^T > 0$ is the solution of the Lyapunov equation $\overline{P}(A - HC) + (A - HC)^T \overline{P} = -I$. Boundedness of all signals of the closed-loop system can be proved by an argument similar to the one in Section IV-A1. First, it is not difficult to show that for all $(e, \hat{\theta}) \in \{V \le c_4\} \cap \Omega_\delta$, there exist constants $c_6, c_7 > 0$ such that the sets $\{V_1 \le c_6\}$ and $\{V_\xi \le c_7\epsilon^2\}$ are positively invariant. Next, using the results of [1], for all $(e, \hat{\theta}, \tilde{\zeta}, \xi)$ belonging to the set

$$R = \{\{V \le c_4\} \cap \Omega_\delta\} \times \{V_1 \le c_6\} \times \{V_\xi \le c_7\epsilon^2\}$$

the derivative of V satisfies

$$\dot{V} \le -kV + k(c_2 + c_3) + k_\epsilon \epsilon + k_d d \tag{14}$$

where $k_\epsilon > 0$. Hence for all

$$d < d^* = \frac{k(c_4 - c_3 - c_2)}{2k_d}$$

and

$$\epsilon < \epsilon^* = \frac{k(c_4 - c_3 - c_2)}{2k_\epsilon} \tag{15}$$

$\dot{V} < 0$ on $\{V = c_4\} \cap \Omega_\delta$ and the set R is positively invariant. Using the difference in speeds between the slow and fast variables and the fact that $\dot{V}_\xi \le -(1/2\epsilon)||\xi||^2$ outside $\{V_\xi \le c_7\epsilon^2\}$ it can be shown that the trajectory enters the set R during the time interval $[0, T(\epsilon)]$, where $T(\epsilon) \to 0$ as $\epsilon \to 0$. Hence, as in the previous case, for sufficiently small d and ϵ, all the states are bounded and (14) is satisfied for all $t \ge T(\epsilon)$. Hence the "approximate tracking error" is of the order $O(\epsilon + d + (1/\gamma_f) + (1/\gamma_g))$. Adaptive output feedback control that uses parameter projection, high adaptation gain, and control saturation has also

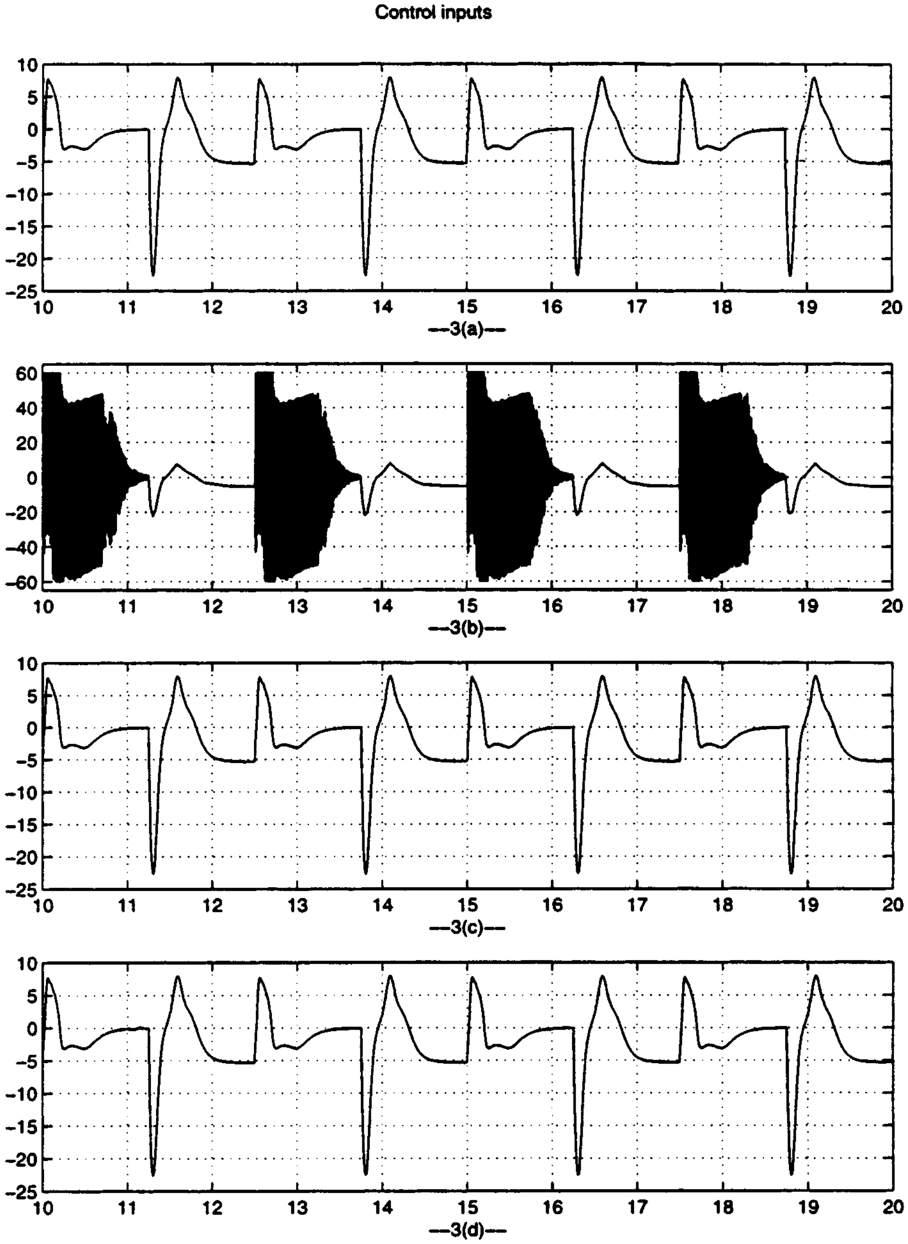


Fig. 3. (a) State feedback, $\mu = 0.5$. (b) Output feedback, $\epsilon = 10^{-3}$, $\mu = 0.5$. (c) Output feedback $\epsilon = 10^{-4}$, $\mu = 0.5$. (d) Output feedback $\epsilon = 10^{-4}$, $\mu = 0.1$..

been considered in a similar setting by [11]. In particular, [11] contains a result similar to the one in this section, namely that, the tracking error can be made as small as desired by increasing the observer and parameter adpatation gains.

B. Reconstruction Error with a Known Bound

We design an additional robustifying control component to make the mean-square tracking error arbitrarily small, irrespective of the bound on the disturbance d, provided this bound is known. Let

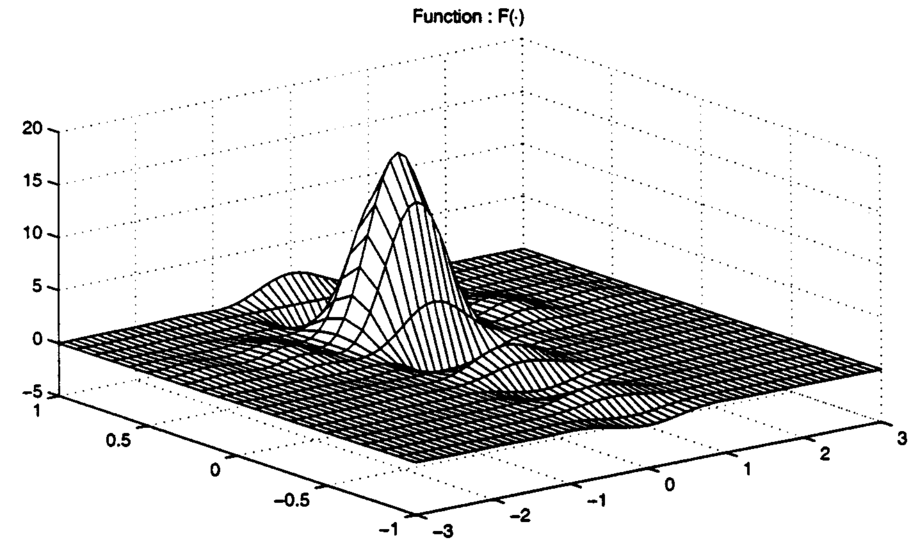
$$v = \frac{-Ke + y_r^{(n)} - \hat{F}(e + \mathcal{Y}_r, z, \hat{\theta}_f) + v_1}{\hat{G}(e + \mathcal{Y}_r, z, \hat{\theta}_g)}. \qquad (16)$$

We will choose the robustifying component v_1 using the Lyapunov redesign technique, e.g., [13]. Assume that

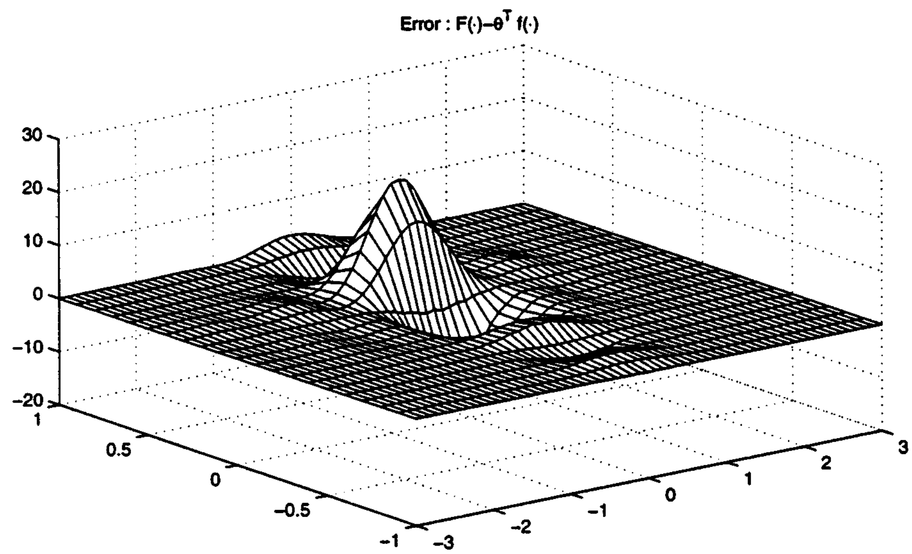
$$||d(\cdot)|| \le \rho(e,z) + k_v ||v_1||, \qquad 0 \le k_v < 1$$

where ρ and k_v are known. Take $\eta(e,z) \ge \rho(e,z)$ and define $s = 2e^T Pb$

$$\psi_r(e,z) = \begin{cases} -\dfrac{\eta(e,z)}{(1-k_v)} \cdot \dfrac{s}{|s|}, & \text{for } \eta(e,z)|s| \ge \mu \\ -\dfrac{\eta^2(e,z)}{(1-k_v)} \cdot \dfrac{s}{\mu}, & \text{for } \eta(e,z)|s| < \mu \end{cases} \qquad (17)$$



(a)



(b)

Fig. 4. (a) $F(y, \dot{y})$. (b) $F(y, \dot{y}) - theta_f^T f(y, \dot{y})$.

and

$$\psi(e, z, \mathcal{Y}_R, \hat{\theta}_f, \hat{\theta}_g) = \frac{-Ke + y_r^{(n)} - \hat{F}(e + \mathcal{Y}_r, z, \hat{\theta}_f) + \psi_r(e, z)}{\hat{G}(e + \mathcal{Y}_r, z, \hat{\theta}_g)}. \quad (18)$$

The adaptive controller is taken as $v = \psi^s(\hat{e}, z, \mathcal{Y}_R, \hat{\theta}_f, \hat{\theta}_g)$. The arguments of the preceeding section can be extended to show that $\exists \epsilon^* > 0$, $\mu^* > 0$ and $\tau > 0$, such that $\forall 0 < \epsilon < \epsilon^*$ and $0 < \mu < \mu^*$, all signals are bounded and that for all $t \geq \tau$, the tracking error is of the order $O(\epsilon + \mu + (1/\gamma_f) + (1/\gamma_g))$, and can be made arbitrarily small by appropriate choice of the design parameters ϵ, μ, Γ_f, and Γ_g.

V. Simulations

In this section, three simulations are presented to illustrate the points made in the earlier sections. The programs for the simulations are written in Matlab, using the neural-network toolbox. In the first simulation, we show the effect of changing various design parameters on the tracking error. In the second, we attempt to justify the need to adapt for the network's weights. Last, we demonstrate the effect of the network's size on the controller's performance. The plant used in all these simulations is the same one used in [16] and [17], namely $\ddot{y} = F(y, \dot{y}) + G(y)u$ where

$$F(y, \dot{y}) = 16 \frac{\sin(4\pi y)}{4\pi y} \left(\frac{\sin(\pi \dot{y})}{\pi \dot{y}} \right)^2$$

and

$$G(y) = 2 + \sin(3\pi(y - 0.5)).$$

A. Simulation 1

The plant output is required to track a reference signal y_r that is the output of a low-pass filter with transfer function $(1 +$

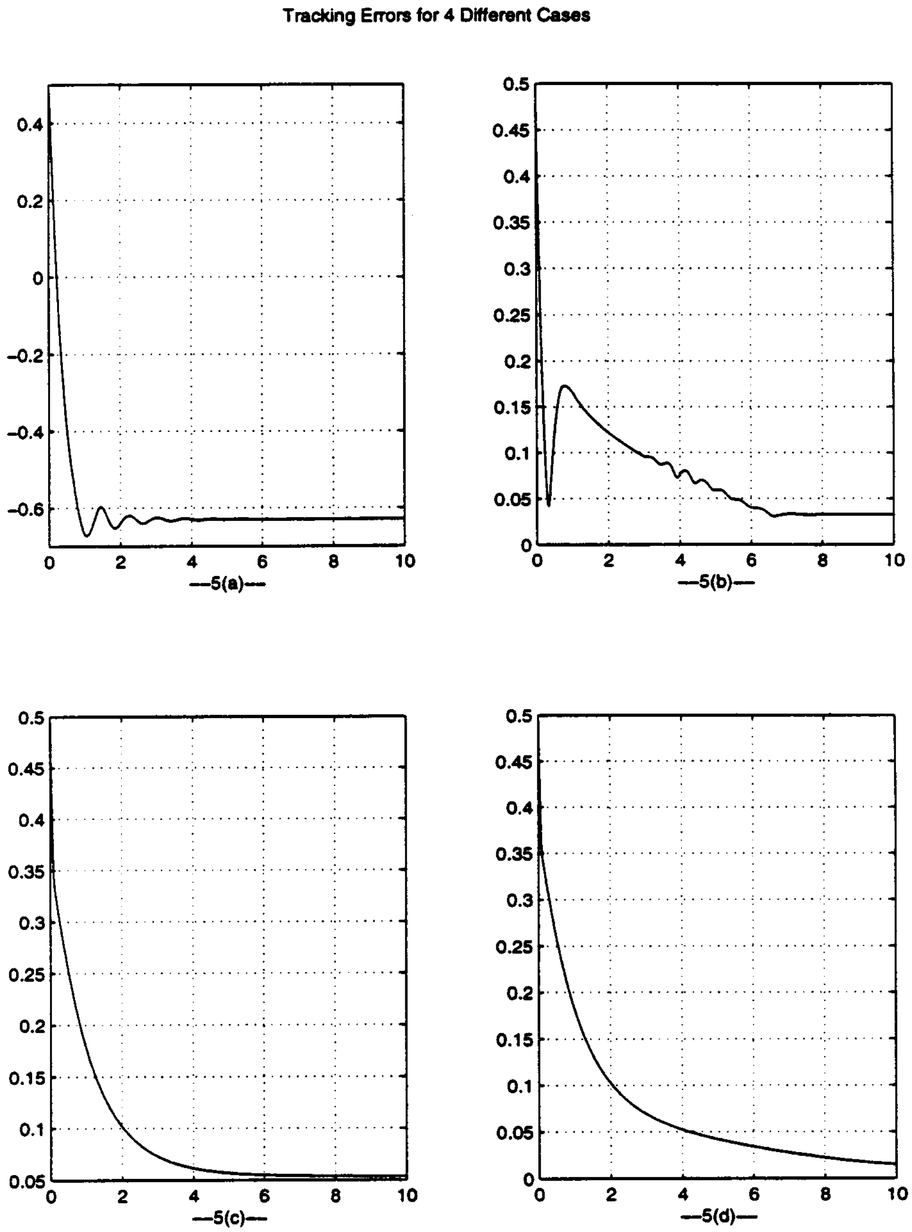


Fig. 5. (a) No adaptation for weights, no robust control. (b) Only adpatation for weights. (c) Only robust control. (d) Adaptation for weights and robust control.

$s/10)^{-3}$, driven by a unity amplitude square wave input with frequency 0.4 Hz and a time average of 0.5. The reference and its derivatives are shown in Fig. 1. As can be seen, the set $\mathcal{Y}_R$ can be taken as $[00, 1] \times [-3, 3] \times [-25, 25]$. Since $m = 0$, there is no need to augment integrators at the system's input. Let $\tilde{Y} =^{\text{def}} [-1, 1] \times [-3, 3]$. We use two RBF networks to approximate the functions $F(y, \dot{y})$ and $G(y)$ over $\tilde{Y}$. The networks have 48 Gaussian nodes with variance[4] $\sigma^2 = 4\pi$ spread over a regular grid that covers $\tilde{Y}$. Off-line training is done to obtain weights θ_{f_0} and θ_{g_0} that result in "optimal" approximations of the functions F and G. However, the reconstruction errors are still quite large in this case, at some points being comparable to the value of the function itself. Based on the values of θ_{f_0} and θ_{g_0}, the sets Ω_f and Ω_g in Assumption 1 are taken as $[\theta_{f_0} - 0.1, \theta_{f_0} + 0.1]$ and $[\theta_{g_0} - 0.1, \theta_{g_0} + 0.1]$, where the addition and subtraction are done componentwise. The adaptation gains Γ_f and Γ_g are taken for simplicity as $10^3 I$. The value δ in (11) is taken as 0.001. The values of the other design parameters are $\eta = 40$ and $k_v = 0.7$. The initial condition $x(0)$ is taken as $(-0.5, 2.0)$. Fig. 2(a) shows the tracking error for the state feedback case with $\mu = 0.5$, Fig. 2(b) is the output feedback case with $\epsilon = 10^{-3}$, Fig. 2(c) with ϵ reduced to 10^{-4}, and Fig. 2(d) with μ reduced to 0.1. Fig. 3 shows the corresponding control inputs. The simulation illustrates several points: 1) by using a robustifying component, it is possible to obtain reasonable performance even with networks that give large reconstruction errors; 2) as ϵ is decreased, we recover the performance obtained under state feedback; and 3) an n-fold decrease in μ results in

[4]See [16] for a definition of this term in relation to RBF networks.

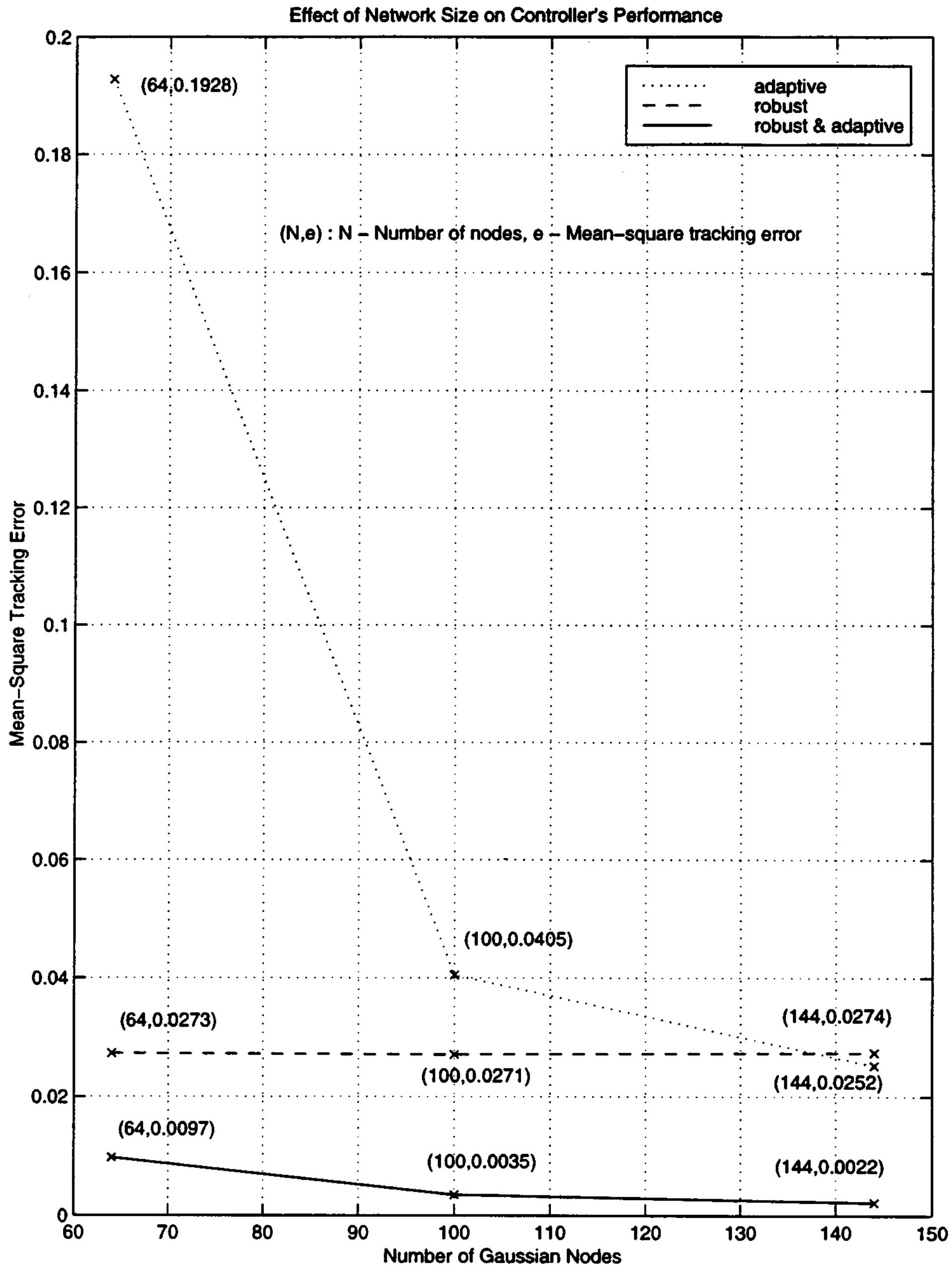


Fig. 6. Effect of network size on the controller's performance.

approximately an n-fold decrease in the tracking error. Thus, by decreasing μ, we can meet more stringent requirements on the tracking error.

B. Simulation 2

The initial weights obtained by off-line training may not be close to their optimal vlaues. This might, for example, be the case when the off-line training is done (based) on a nominal model that differs considerably from the actual one. For definiteness, suppose the function $F(y, \dot{y})$ is any one of the functions

$$F(y,\dot{y}) = k_1 \frac{\sin(4\pi(y-k_2))}{4\pi(y-k_2)} \left(\frac{\sin(\pi(\dot{y}-k_3))}{\pi(\dot{y}-k_3)}\right)^2$$

where $k_1 \in 15, 17$, $k_2 \in [-0.5, 0.5]$ and $k_3 \in [-1, 1]$ and that a nominal model is the one used before, that is,

$$F_{nom}(y,\dot{y}) = 16 \frac{\sin(4\pi y)}{4\pi y} \left(\frac{\sin(\pi \dot{y})}{\pi \dot{y}}\right)^2.$$

For simplicity, we take $G(y) = 1$. Further, the reference signal is taken as $y_r = 0.4$. This time, we use an RBF network with 192 Guassian nodes to "construct" the function $F(\cdot)$, with the parameters of the network chosen as before. Based on the nominal model, we do off-line training to obtain initial estimates θ_{f_0} and θ_{g_0}. The choice of the set Ω_f is crucial. It is chosen in a way which gaurantees that any of the functions $F(\cdot)$ mentioned above can be reasonably approximated[5] by some θ_f in Ω_f. By this, we mean that, for every possible choice of k_1, k_2, and k_3, there exists $\theta_f \in \Omega_f$ such that $|F(\cdot) - \theta_f^T f(\cdot)| \sim |F_{nom}(\cdot) - \theta_{f_0}^T f(\cdot)| \forall y, \dot{y} \in \tilde{Y}$. For the purpose of simulation, the values of k_1, k_2, and k_3 are taken to be 17, 0.4, and 0, respectively. This choice ensures that with the "nominal" weights θ_{f_0}, the reconstruction error is quite large in the region of the state space where the reference lies. Fig. 4 shows the function $F(y, \dot{y})$ and the error $F(\cdot) - \theta_{f_0}^T f(\cdot)$ that results from using the nominal weights. The values of the parameters used in the design are $\Gamma_f = 10^3 I$, $\delta = 0.001$, $\epsilon = 10^{-4}$, $\eta = 20$, $k_v = 0$ and $\mu = 0.2$. The initial condition $x(0)$ is taken as $(0.9, -2.75)$. Fig. 5(a) shows the tracking error for the case when there is no adaptation for the weights, that is, $\Gamma_f = 0$ and no robust control component, Fig. 5(b) for the case when the weights are adapted but there is no robust component, Fig. 5(c) for the case when the weights are not adapted but there is a robust component, and Fig. 5(d) for the case when the weights are adapted and a robust component is used.

The following points are noteworthy. In the first case the tracking error is quite large because we simply do a crude cancellation of the network nonlinearity based on a nominal model. When we start adapting for the weights, the difference between the function F and its estimate $\hat{F}$ provided by the network decreases and hence the tracking error also decreases. However, even with the network providing its "best" approximation, there is a residual error. In the case where we simply use robust control, the performance shows an improvement over the first case and is almost comparable to the error in the second case. Finally, in the case where we do both adaptation and robust control, the network reconstruction error decreases and the robust component handles this smaller error better. Thus the tracking error is the smallest in this case.

C. Simulation 3

In Section IV-B we saw that decreasing μ results in a decrease in the mean-square tracking error. While theoretically μ can be made as small as we want, it is not always possible in practice to do so. This is because, in many practical applications, the system contains high-frequency unmodeled dynamics. Decreasing μ implies a "high-gain like" feedback inside the layer $|s| < (\mu/\eta)$ which might result in the excitation of the unmodeled dynamics. In this section, we assume that μ cannot be made smaller than 0.1, fix it at this value and examine the controller's performance as the network size is varied. To be able to do this, we first need to define a "suitable" measure of the network's performance. For a given network, let $e_1, e_2, \cdots, e_n$ denote ultimate bounds on the tracking error[6] corresponding to initial conditions $x_{01}, x_{02}, \cdots, x_{0n}$. We take the mean-square error $\sqrt{\sum e_i^2/n)}$ to be a measure of the network's performance. We use the same plant used in the previous simulation, with initial estimates for the weights based on the nominal model. The reference y_r is chosen as $0.4 + 0.1\sin(t)$. We compare the performance of three networks, having 64, 100 and 144 Gaussian nodes, respectively. The networks are used to construct F on $\tilde{Y} = [-1, 1] \times [-1, 1]$. For each network, four sets of initial conditions for the state x are used, $(-0.9, -0.9)$, $(-0.9, 0.9)$, $(0.9, -0.9)$ and $(0.9, 0.9)$ and the mean-square error is evaluated. Fig. 6 summarizes the results of the simulation. The dashed line shows the mean-suare error for the case when only robust control is used, the dotted line for the case when only adaptation is used and the solid line for the case when both adaptation and robust control are used. As can be seen, the error is almost constant in the first case. Thus, by using only robust control, we cannot hope to decrease the error beyond a certain point. In the second case, the mean-square error decreases as the network size increases. This is not surprising because increasing the network's size increases its approximation capabilities. Since the error is of the order $O(\epsilon + d + (1/\gamma_f) + (1/\gamma_g))$, it decreases as the size of the network increases. This suggests that as requirements on the tracking error become more stringent, it becomes necessary to increase the size of the network. Last, the performance in the last case is the best of the three cases.

[5]This might require making Ω_f larger than what it would have been if we had assumed that the actual and the nominal models are identical.

[6]As observed by simulation.

VI. CONCLUSIONS

An adaptive output feedback scheme that uses RBF neural networks has been studied for the control of a class of nonlinear systems represented by input–output models. The objective of the design is to achieve good tracking performance in the absence of known system dynamics. The method is based on the results of [1] and uses RBF networks to approximately construct the system nonlinearities. The reconstruction errors of the networks are not required to be small, thus allowing for the use of lower order networks. This is made possible by combining robust control tools with those of adaptive control. Another merit of the scheme is the use of the HGO to robustly estimate the output derivatives, thus dispensing with the requirement of availability of all the states. A key difference from the work of [1] is the requirement that the parameter adaptation gains be sufficiently high. The results are similar to those of [11], which also considered adaptive output feedback control in a similar setting. The effectiveness of the scheme is demonstrated through simulations. The simulations also illustrate the effect of changing various design parameters and of the network size on the controller's performance.

REFERENCES

[1] B. Aloliwi and H. K. Khalil, "Robust adaptive output feedback control of nonlinear systems without persistence of excitation," *Automatica*, vol. 33, no. 11, pp. 2025–2032, Nov. 1997.

[2] G. A. Rovithakis and M. A. Christodoulou, "A robust neural adaptive control scheme," in *Proc. 1995 Conf. Decision Contr.*, 1995, pp. 1831–1832.

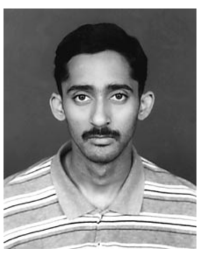
[3] C. I. Byrnes and A. Isidori, "Asmptotic stabilization of minimum phase nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 36, no. 10, pp. 1122–1137, Oct. 1991.

[4] F. C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Trans. Automat. Contr.*, vol. 40, no. 5, pp. 791–801, May 1992.

[5] S. Fabri and V. Kadirkamanathan, "Dynamic structure neural networks for stable adaptive control of nonlinear systems," *IEEE Trans. Neural Networks*, vol. 7, no. 5, pp. 1151–1166, Sept. 1996.

[6] J. A. Farrell, "Persistence of excitation conditions in passive learning control," *Automatica*, vol. 33, no. 4, pp. 699–703, 1997.

[7] S. S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.
[8] A. Isidori, *Nonlinear Control Systems*. New York: Springer-Verlag, 1995.
[9] R. Ordonez, J. T. Spooner, and K. M. Passino, "Stable direct adaptive control of a class of discrete-time nonlinear systems," in *Proc. 13th World Congr. IFAC*, 1996, pp. 343–348.
[10] S. Jagannathan and F. L. Lewis, "Discrete-time neural-net controller with guaranteed performance," *IEEE Trans. Automat. Contr.*, vol. 41, no. 11, pp. 1693–1699, Nov. 1996.
[11] M. Jankovic, "Adaptive output feedback control of nonlinear feedback linearizable systems," *Int. J. Adaptive Contr. Sign. Proc.*, vol. 10, pp. 1–18, 1996.
[12] H. K. Khalil, "Adaptive output feedback control of nonlinear systems represented by input–output models," *IEEE Trans. Automat. Contr.*, vol. 41, no. 2, pp. 177–188, Feb. 1996.
[13] ——, *Nonlinear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[14] M. Polycarpou, "Stable adaptive neural control scheme for nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 41, no. 3, pp. 447–451, Mar. 1996.
[15] M. Polycarpou and P. Ioannou, "Modeling, identification, and stble adaptive control of continuous-time nonlinear dynamical systems using neural networks," in *Proc. 1992 Amer. Contr. Conf.*, June 1992, pp. 36–40.
[16] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol. 3, no. 6, pp. 837–863, Nov. 1992.
[17] E. Tzirkel-Hancock and F. Fallside, "Stable control of nonlinear systems using neural networks," *Int. J. Robust Nonlinear Contr.*, vol. 2, no. 1, pp. 63–86, Apr. 1992.



Sridhar Seshagiri received the B.Tech degree from the Indian Institute of Technology, Madras, and the M.S. degree from Michigan State University, East Lansing, in 1995 and 1998, respectively, both in electrical engineering.

He is currently working with the Hardware-in-the-Loop group at the Scientific Research Labs (SRL), Ford Motor Company, Dearborn. His current research interests include automotive control and the stability and control of hybrid systems.

Hassan K. Khalil (S'77–M'78–SM'85–F'89) received the B.S. and M.S. degrees from Cairo University, Cairo, Egypt, and the Ph.D. degree from the University of Illinois, Urbana-Champaign, in 1973, 1975, and 1978, respectively, all in electrical engineering.

Since 1978, he has been with Michigan State University, East Lansing, where he is currently Professor of Electrical and Computer Engineering. He has consulted for General Motors and Delco Products. He has published several papers on singular perturbation methods, decentralized control, robustness, nonlinear control, and adaptive control. He is author of the book *Nonlinear Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1996), coauthor of the book *Singular Perturbation Methods in Control: Analysis and Design* (New York: Academic, 1986), and coeditor of the book *Singular Perturbation in Systems and Control* (New York: IEEE, 1986).

Dr. Khalil served as Associate Editor of IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 1984 to 1985; Registration Chairman of the IEEE-CDC Conference in 1984; Finance Chairman of the 1987 American Control Conference (ACC); Program Chairman of the 1988 ACC; General Chair of the 1994 ACC; Associate Editor of *Automatica* from 1992 to 1999, and Action Editor of *Neural Networks* from 1998 to 1999. He is currently serving as Editor of *Automatica* for nonlinear systems and control, and is a member of the IEEE Control Systems Society Board of Governors. He was the recipient of the 1983 Michigan State University Teacher Scholar Award, the 1989 George S. Axelby Outstanding Paper Award of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the 1994 Michigan State University Withrow Distinguished Scholar Award, and the 1995 Michigan State University Distinguished Faculty Award.