This is to certify that the

thesis entitled

INDIRECT ADAPTIVE OUTPUT FEEDBACK CONTROL

presented by

Sridhar Seshagiri

has been accepted towards fulfillment
of the requirements for

Master's degree in Electrical Eng

Hassan Khalil
Major professor

Date 7/10/98

INDIRECT ADAPTIVE OUTPUT FEEDBACK CONTROL

By

Sridhar Seshagiri

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Electrical Engineering

1998

Abstract

INDIRECT ADAPTIVE OUTPUT FEEDBACK CONTROL

By

Sridhar Seshagiri

An adaptive output feedback control scheme for the output tracking of a class of continuous-time nonlinear plants is studied. The method uses parameter projection, control saturation, and a high-gain observer to achieve semi-global uniform ultimate boundedness. First, an application to the longitudinal control of a platoon of non-identical vehicles is discussed. A nonlinear model is used to represent the vehicle dynamics of each vehicle within the platoon. The model depends linearly on unknown parameters which belong to a known compact set. In contrast to previous work, the number of measured quantities is kept to a minimum. The efficacy of the proposed method is demonstrated through simulations. Next, an application to the control of unknown nonlinear systems using an RBF neural network is discussed. The RBF network is used to adaptively compensate for the plant nonlinearities. The network's weights are adjusted using a Lyapunov-based scheme. It is shown 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.

To my family

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to the following people for having made the completion of this thesis possible :

To my advisor Dr. Hassan K. Khalil, for his insight, guidance and invaluable assistance, without which this thesis would not have been possible. His dedication to the subject and personal integrity have served as an example and an inspiration to me.

To all my other instructors at MSU, for their skillful instruction in the classroom, but more importantly, for their constant encouragement and friendship.

To my friends, for having made my stay here the memorable experience that it has been.

And to my family, for years of support, patience and understanding.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Adaptive control of nonlinear systems has matured as an exciting area of research over the last few years. Early efforts in the design of adaptive controllers for nonlinear systems focused on the state-feedback problem and resulted in a systematic design procedure called adaptive backstepping [16]. An extension to the more challenging output feedback problem for the case when the system nonlinearities depended only on the output was next made. This problem was first solved under restrictive structural and growth conditions on the nonlinearities [14, 15]. Subsequently, the growth restrictions were removed [17], but the structural restriction remained: the output nonlinearities were not allowed to precede the control input. The removal of this structural restriction by Marino and Tomei in [31] was a breakthrough in adaptive nonlinear output feedback control. Their work addressed the problem of designing a global adaptive output feedback tracking control for single-input single-output

(SISO) nonlinear systems that are linear with respect to the input and an unknown constant parameter vector. This was achieved by merging the filtered transformations of [29] and [30] with the adaptive backstepping scheme of [16] and using a novel compensation of the estimation error effects. The scheme however suffered from the drawbacks of overparametrization inherited from the original adaptive backstepping procedure and restriction to the unnormalized gradient update law. These drawbacks were addressed in the work of Krstic and Kokotovic [27], which proposed three new adaptive schemes to achieve minimal parametrization and to remove the restriction of unnormalized gradient update.

Indirect adaptive control design for systems representable by input-output models was done by Khorasani in [26]. The starting point in [26] was a class of SISO nonlinear systems represented in state-space form. Under the assumption of certain rank conditions, the technique of output prolongation [12] was used to convert the model to an equivalent input-output representation. The scheme however required the derivatives of the output to be available for feedback. In the work of Khalil [24] this requirement was relaxed by the use of a high-gain observer to estimate the derivatives of the output. The system under consideration in [24] was SISO, input-output linearizable, minimum-phase, and modeled by an input-output model of the form of an nth-order differential equation. By combining results from [9, 40, 41] with Lyapunov-based adaptive design [18, 32], he designed a semiglobal controller that achieved asymptotic output tracking for reference signals which were bounded and had bounded derivatives up to the nth order. The design was simpler than traditional ones since it did not use filtering or error augmentation ideas. It was simply a state feedback controller with a linear observer. A similar control design was also presented in [23].

An important drawback of the result of [24] was the requirement of persistence of excitation (PE) not only for parameter convergence but even for tracking error convergence. This is unusual in adaptive control results where tracking error convergence is shown without PE. This drawback was addressed in [2]. This improvement over the result of [24] was made possible by changing the analysis approach. In [24], convergence was proved by showing that, under state feedback and the PE condition, the set of zero tracking error and zero parameter error is an exponentially stable invariant set. Then, singular perturbation analysis was used to show that this same property is recovered under output feedback for sufficiently small ϵ. This idea does not work in the lack of PE because the set of zero tracking error and zero parameter (or partial parameter) error is not exponentially stable. In [2], the closed-loop system under output feedback is analayzed directly and various Lyapunov functions are combined to form a composite Lyapunov function that shows tracking error and partial parameter convergence.

1.2 Organization

The rest of this thesis is organized as follows. Chapter 2 is a brief overview of the technique of [2]. In Chapter 3, an application to the longitudinal control of automated vehicles is presented. The controlled vehicle is assumed to be capable of measuring (or estimating) necessary dynamical information from the vehicle immediately in front of it by its onboard sensors. The computer in the vehicle processes the measured data and generates proper throttling and braking actions to follow the vehicle in front at a safe distance. The property that the spacing error for a controlled vehicle can be regulated is referred to as local stability [38]. An important control objective in the longitudinal control problem is that of asymptotic stability of a platoon or string of cars following one another. A platoon is said to be asymptotically stable if there are no slinky-type effects [37] within the platoon, i.e, there are no amplifications in the deviations of vehicle spacings from their steady state values from the front to the end of the platoon. It is well-known that for the case where the vehicle following is not cooperative,i.e, information is not exchanged with other vehicles, velocity dependent spacing rules can guarantee asymptotic platoon stability. In our work, we use the "constant time headway" spacing rule [8, 39]. Simulations are presented for the case of a platoon of four cars following a leader. Good references on the longitudinal vehicle control problem can be found in [1], [6],[20] and [36]. In Chapter 4, we study the application of the technique of [2] to the adaptive control of unknown nonlinear systems using RBF networks. The design is developed for systems represented by input-output models and RBF networks are used to approximate the system's nonlinearities. The weights of the networks are adapted using a Lyapunov-based scheme. Simulations are presented to demonstrate the need for adaptation, the role of the robustifying component and the effect of the network's size (number of Gaussian nodes) on the tracking performance. Discussions and a summary are presented in Chapter 5.

Chapter 2

Robust Adaptive Tracking Control

2.1 Problem Statement

Consider a single-input–single-output nonlinear system represented globally by the nth-order differential equation

$$y^{(n)} = f_0(\cdot) + \textstyle\sum_{i=1}^{p} f_i(\cdot)\theta_i + [g_0(\cdot) + \textstyle\sum_{i=1}^{p} g_i(\cdot)\theta_i]u^{(m)} \tag{2.1}$$

where u is the control input, y is the measured output, $y^{(i)}$ denotes the ith derivative of y, and $m < n$. The functions f_i and g_i are known smooth nonlinearities which may depend on y, $y^{(1)}$, ..., $y^{(n-1)}$, u, $u^{(1)}$, ..., $u^{(m-1)}$; i.e.,

$$f_i(\cdot) = f_i(y, y^{(1)}, \ldots, y^{(n-1)}, u, u^{(1)}, \ldots, u^{(m-1)}),\ 0 \le i \le p \text{ and}$$

$$g_i(\cdot) = g_i(y, y^{(1)}, \ldots, y^{(n-1)}, u, u^{(1)}, \ldots, u^{(m-1)}),\ 0 \le i \le p$$

The constant parameters θ_1 to θ_p are unknown, but the vector $\theta = [\theta_1, \ldots, \theta_p]^T$ belongs to Ω, a known compact convex subset of R^p. By augmenting a series of m integrators at the input side of the system, the extended system can be represented 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 $v = u^{(m)}$ is 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

$$\left.\begin{aligned}
\dot{x}_i &= x_{i+1}, \quad 1 \leq i \leq n-1 \\
\dot{x}_n &= f_0(x,z) + \theta^T f(x,z) \\
&\quad + [g_0(x,z) + \theta^T g(x,z)]v \\
\dot{z}_i &= z_{i+1}, \quad 1 \leq i \leq m-1 \\
\dot{z}_m &= v \\
y &= x_1
\end{aligned}\right\} \tag{2.2}$$

where

$$\begin{aligned}
x &= [x_1, \ \ldots, \ x_n]^T, \quad z = [z_1, \ \ldots, \ z_m]^T \\
f &= [f_1, \ \ldots, \ f_p]^T, \quad g = [g_1, \ \ldots, \ g_p]^T
\end{aligned}$$

Assumption 1 $|g_0(x,z) + \theta^T g(x,z)| \geq k > 0 \ \forall \ x \in R^n, \ z \in R^m$ and $\theta \in \Omega_1$, *where Ω_1 is a compact set that contains Ω in its interior.*

Assumption 1 ensures that (2.2) is input-output linearizable by full state feedback for every $\theta \in \Omega$. Using the results of [5], it can be shown that there exists a global diffeomorphism, possibly dependent on θ,

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

with $T_1(0,0) = 0$, which transforms the last m state equations of (2.2) into

$$\dot{\zeta} = F(\zeta, x, \theta)$$

This, together with the first n state equations of (2.2), defines a global normal form. The objective is to design an adaptive output feedback controller which guarantees boundedness of all state variables and tracking of a given reference signal y_r, where y_r is bounded, has bounded derivatives up to the nth-order, and $y_r^{(n)}$ is piecewise continuous.

2.2 Control Design

The controller design is done in two steps. First a state feedback controller that ensures boundedness of all signals and yields zero steady-state tracking error is designed. This same controller is used in the output feedback case with the states replaced by estimates provided by a high-gain observer (HGO). The control is saturated outside a compact region of interest to protect the system from peaking induced by the HGO.

2.2.1 State Feedback

We design an adaptive state feedback controller so that the output y tracks the given reference signal y_r. Define

$$
\begin{aligned}
e_i &= y^{(i-1)} - y_r^{(i-1)}, \quad 1 \le i \le n \\
e &= [e_1,\ e_2,\ \ldots,\ e_n]^T \\
\mathcal{Y}(t) &= [y(t),\ y^{(1)}(t),\ \ldots,\ y^{(n-1)}(t)]^T \\
\mathcal{Y}_r(t) &= [y_r(t),\ y_r^{(1)}(t),\ \ldots,\ y_r^{(n-1)}(t)]^T \\
\mathcal{Y}_R(t) &= [y_r(t),\ y_r^{(1)}(t),\ \ldots,\ y_r^{(n-1)}(t),\ y_r^{(n)}(t)]^T
\end{aligned}
$$

and let Y and Y_R be any given compact subsets of R^n and R^{n+1}, respectively, such that $\mathcal{Y}(0) \in Y$ and $\mathcal{Y}_R(t) \in Y_R \ \forall \ t \ge 0$. We rewrite (2.2) as

$$
\begin{aligned}
\dot{e} &= A_m e + b\{Ke + f_0(e + \mathcal{Y}_r, z) + \theta^T f(e + \mathcal{Y}_r, z) \\
&\quad + [g_0(e + \mathcal{Y}_r, z) + \theta^T g(e + \mathcal{Y}_r, z)]v - y_r^{(n)}\} \qquad (2.3) \\
\dot{z} &= A_2 z + b_2 v
\end{aligned}
$$

where (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 2 *The system $\dot{\zeta} = F(\zeta, \mathcal{Y}_r, \theta)$ has a unique steady-state solution $\bar{\zeta}$.*

Moreover, with $\tilde{\zeta} = \zeta - \bar{\zeta}$ *the system*

$$\dot{\tilde{\zeta}} = F(\bar{\zeta} + \tilde{\zeta}, e + \mathcal{Y}_r, \theta) - F(\bar{\zeta}, \mathcal{Y}_r, \theta) \stackrel{\text{def}}{=} F_2(\tilde{\zeta}, e, \mathcal{Y}_r, \bar{\zeta}, \theta) \tag{2.4}$$

has a continuously differentiable function $V_1(t, \tilde{\zeta})$*, possibly dependent on* θ*, that satisfies* [1]

$$\eta_1 \|\tilde{\zeta}\|^2 \leq V_1(t, \tilde{\zeta}) \leq \eta_2 \|\tilde{\zeta}\|^2$$

$$\frac{\partial V_1}{\partial t} + \frac{\partial V_1}{\partial \tilde{\zeta}} F_2(\tilde{\zeta}, e, \mathcal{Y}_r, \bar{\zeta}, \theta) \leq -\eta_3 \|\tilde{\zeta}\|^2 + \eta_4 \|\tilde{\zeta}\| \|e\|$$

where $\eta_1, \eta_2, \eta_3 > 0$*, and* $\eta_4 \geq 0$ *are independent of* $\mathcal{Y}_r$ *and* θ*.*

The steady-state response of a nonlinear system is introduced in [19, Section 8.1]. Basically, it is a particular solution towards 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 (2.4), with e as input, is input-to-state stable. Consequently, the zero dynamics of (2.2) are exponentially stable and (2.2) is minimum phase.

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

$$V = e^T P e + \tfrac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta} \tag{2.5}$$

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

where $\Gamma = \Gamma^T > 0$, $\tilde{\theta} = \hat{\theta} - \theta$, and $\hat{\theta}$ is an estimate of θ to be determined by the parameter adaptation law. The derivative of V along the trajectories of the system is given by

$$\begin{aligned}\dot{V} = -e^T Q e &+ \tilde{\theta}^T \Gamma^{-1} \dot{\hat{\theta}} + 2e^T P b\{f_0(e + \mathcal{Y}_r, z) + \theta^T f(e + \mathcal{Y}_r, z) \\ &+ Ke - y_r^{(n)} + [g_0(e + \mathcal{Y}_r, z) + \theta^T g(e + \mathcal{Y}_r, z)]v\}\end{aligned}$$

Taking

$$\begin{aligned} v &= \frac{-Ke + y_r^{(n)} - f_0(e + \mathcal{Y}_r, z) - \hat{\theta}^T f(e + \mathcal{Y}_r, z)}{g_0(e + \mathcal{Y}_r, z) + \hat{\theta}^T g(e + \mathcal{Y}_r, z)} \\ &\stackrel{\text{def}}{=} \psi(e, z, \mathcal{Y}_R, \hat{\theta}) \end{aligned} \tag{2.6}$$

we can rewrite the expression for $\dot{V}$ as $\dot{V} = -e^T Q e + \tilde{\theta}^T \Gamma^{-1} [\dot{\hat{\theta}} - \Gamma \phi]$ where

$$\phi = 2e^T P b[f(e + \mathcal{Y}_r, z) + g(e + \mathcal{Y}_r, z)\psi(e, z, \mathcal{Y}_R, \hat{\theta})] = \phi(e, z, \mathcal{Y}_R, \hat{\theta})$$

The parameter adaptation law is chosen as in [24], 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

$$\tilde{\theta}^T \Gamma^{-1} [\dot{\hat{\theta}} - \Gamma\phi] \le 0 \tag{2.7}$$

and $\hat{\theta}(t)$ belongs to a compact set Ω_δ for all $t \geq 0$, where $\Omega_1 \supset \Omega_\delta \supset \Omega$. This can be achieved by standard adaptation laws with smoothed parameter projection to ensure that $\text{Proj}(\hat{\theta}, \phi)$ is locally Lipschitz. As an example, consider the case when Ω is the convex hypercube $\Omega = \{\theta \mid a_i \leq \theta_i \leq b_i\},\ 1 \leq i \leq p\}$. Let

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

where $\delta > 0$ is chosen such that $\Omega_\delta \subset \Omega_1$, and choose Γ to be a positive diagonal matrix. In this case the projection $\text{Proj}(\hat{\theta}, \phi)$ is taken as

$$[\text{Proj}(\hat{\theta}, \phi)]_i = \begin{cases} \gamma_{ii}\phi_i, & \text{if } a_i \leq \hat{\theta}_i \leq b_i \text{ or} \\ & \text{if } \hat{\theta}_i > b_i \text{ and } \phi_i \leq 0 \text{ or} \\ & \text{if } \hat{\theta}_i < a_i \text{ and } \phi_i \geq 0 \\ \gamma_{ii}\left[1 + (b_i - \hat{\theta}_i)/\delta\right]\phi_i, & \text{if } \hat{\theta}_i > b_i \text{ and } \phi_i > 0 \\ \gamma_{ii}\left[1 + (\hat{\theta}_i - a_i)/\delta\right]\phi_i, & \text{if } \hat{\theta}_i < a_i \text{ and } \phi_i < 0 \end{cases} \tag{2.8}$$

Inequality (2.7) ensures that $\dot{V} \leq 0$. Therefore, $e(t)$ and $\hat{\theta}$ are bounded for all $t \geq 0$. Since $\mathcal{Y}_r$ is bounded, we conclude that $x(t)$ is bounded, which implies, in view of Assumption 2, that $z(t)$ is bounded. With all signals bounded, we conclude that $e(t)$ $\to 0$ as $t \to \infty$.

2.2.2 High-gain Observer

To implement the state feedback adaptive controller using output feedback, we need to estimate e; there is no need for estimating z as it is already available (the state of the integrators at the input side). With the goal of recovering the performance achieved under state feedback, we use the same high-gain observer used in [24], namely,

$$\left.\begin{aligned} \dot{\hat{e}}_i &= \hat{e}_{i+1} + \alpha_i(e_1 - \hat{e}_1)/\epsilon^i, \quad 1 \le i \le n-1 \\ \\ \dot{\hat{e}}_n &= \alpha_n(e_1 - \hat{e}_1)/\epsilon^n \end{aligned}\right\} \tag{2.9}$$

where ϵ is a small positive parameter to be specified. The positive constants α_i are chosen such that the roots of $s^n+\alpha_1 s^{n-1}+\cdots+\alpha_{n-1}s+\alpha_n = 0$ have negative real parts. To implement the control using output feedback the state e in ψ^s and ϕ is replaced by its estimate $\hat{e}$. We assume that all initial conditions are in a given compact set; in particular, $\hat{\theta}(0) \in \Omega$, $e(0) \in E_0$, and $z(0) \in Z_0$, where E_0 and Z_0 are compact sets. The sets E_0 and Z_0 can be chosen large enough to cover any given bounded initial conditions, but once they are chosen we cannot allow initial conditions outside them. Let $c_1 = \max_{e \in E_0} e^T P e, c_2 = \max_{\theta \in \Omega, \hat{\theta} \in \Omega_1} \frac{1}{2}(\hat{\theta} - \theta)^T \Gamma^{-1} (\hat{\theta} - \theta)$ and $c_3 > c_1 + c_2$. Then $e(t) \in E \stackrel{\text{def}}{=} \{e^T P e \le c_3\}$ for all $t \ge 0$. Let Z be a compact subset of R^m such that Z_0 is in the interior of Z and [2]

$$z(0) \in Z_0 \text{ and } e(t) \in E \ \forall\, t \ge 0 \Rightarrow z(t) \in Z \ \forall\, t \ge 0. \tag{2.10}$$

[2]The set Z can be determined using the Lyapunov function V_1 of Assumption 2. The basic idea is to choose c_z large enough that the set $\{V_1 \le c_z\}$ is positively invariant, and then determine the corresponding set in the z-coordinates.

Let $S \geq \max|\psi(e, z, \mathcal{Y}_R, \hat{\theta})|$ where the maximization is taken over all

$$e \in E_1 \stackrel{\text{def}}{=} \{e^T P e \leq c_4\}, z \in Z, \mathcal{Y}_R \in Y_R, \hat{\theta} \in \Omega_\delta, \text{ where } c_4 > c_3.$$

Define the saturated function ψ^s by

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

where sat$(\cdot)$ is the saturation function.

By taking $\xi_i = (e_i - \hat{e}_i)/\epsilon^{n-i}, \quad 1 \leq i \leq n$ and $\xi = [\xi_1, \ldots, \xi_n]^T$, the closed-loop system is represented by the standard singularly perturbed form

$$\left.\begin{aligned}
\dot{e} &= A_m e + b\{Ke + f_0(\cdot) + \theta^T f(\cdot) \\
&\quad + (g_0(\cdot) + \theta^T g(\cdot))\psi^s(\cdot) - y_r^{(n)}\} \\
\dot{z} &= A_2 z + b_2 \psi^s(\cdot) \\
\dot{\hat{\theta}} &= \text{Proj}(\hat{\theta}, \phi(\cdot)) \\
\epsilon\dot{\xi} &= (A - HC)\xi + \epsilon b\{f_0(\cdot) + \theta^T f(\cdot) \\
&\quad + (g_0(\cdot) + \theta^T g(\cdot))\psi^s(\cdot) - y_r^{(n)}\}
\end{aligned}\right\} \tag{2.11}$$

where $C = [1, 0, \ldots, 0]$, $H = [\alpha_1, \ldots, \alpha_n]^T$, $(A - HC)$ is Hurwitz, and $\hat{e} = e - D\xi$ where D is a diagonal matrix with ϵ^{n-i} as the ith diagonal element.

2.3 Tracking Error Convergence

Tracking error convergence is proved in [2]. For the sake of completeness we outline the proof. The first step in showing tracking error convergence is to confirm that for any initial conditions in the given compact set, all signals of the closed-loop system (under output feedback) are bounded. First it is shown that there exist constants c_5, $c_6 > 0$ such that the set [3]

$$R_s = \{\{V \leq c_3\} \cap \{\hat{\theta} \in \Omega_\delta\}\} \times \{V_1 \leq c_5\} \times \{V_\xi \leq c_6\epsilon^2\}$$

is positively invariant for sufficiently small ϵ, where $V_\xi = \xi^T \bar{P} \xi$ and $\bar{P} = \bar{P}^T > 0$ is the solution of the Lyapunov equation $\bar{P}(A - HC) + (A - HC)^T \bar{P} = -I$. Then, using the difference in speeds between the slow and fast variables and the fact that $\dot{V}_\xi \leq -(1/2\epsilon)||\xi||^2$ outside $\{V_\xi \leq c_6\}$ it is shown that the trajectory enters the set R_s during the time interval $[0, T(\epsilon)]$ and remains thereafter, where $T(\epsilon) \to 0$ as $\epsilon \to 0$. From that time on, the control saturation is not effective and the closed-loop system is given by

$$\left.\begin{array}{rcl} \dot{e} & = & A_m e - b\tilde{\theta}^T \hat{w}(t) + \Lambda(\cdot) \\ \dot{\hat{\theta}} & = & \Gamma_p(\hat{\theta}, \phi) \\ \dot{\tilde{\zeta}} & = & F_2(\tilde{\zeta}, e, \mathcal{Y}_r, \bar{\zeta}, \theta) \\ \epsilon\dot{\xi} & = & (A - HC)\xi - \epsilon b[\tilde{\theta}^T \hat{w}(t) + Ke] + \epsilon\Lambda(\cdot) \end{array}\right\} \quad (2.12)$$

[3]Note that the set $\{V_1 \leq c_5\}$ could be time-dependent. See [25, Section 3.4] for the use of time-dependent sets in the analysis of nonautonomous systems.

where

$$\begin{aligned}
\Gamma_p(\hat{\theta}, \phi) &= \mathrm{Proj}(\hat{\theta}, \phi(\hat{e}, z, \mathcal{Y}_R, \hat{\theta})) \\
\hat{w}(t) &= \hat{f}(\cdot) + \hat{g}(\cdot)\psi(\hat{e}(t), z(t), \mathcal{Y}_R(t), \hat{\theta}(t)) \\
\Lambda(\cdot) &= b\{K(e-\hat{e}) + (f_0 - \hat{f}_0) \\
&\quad + \theta^T(f-\hat{f}) + (g_0 - \hat{g}_0)v + \theta^T(g-\hat{g})v\} \\
\hat{f}(\cdot) &= f(\hat{e} + \mathcal{Y}_r, z) \\
\hat{g}(\cdot) &= g(\hat{e} + \mathcal{Y}_r, z)
\end{aligned}$$

Define w_r by $w_r(t) \stackrel{\text{def}}{=} f(\mathcal{Y}_r, \bar{z}) + g(\mathcal{Y}_r, \bar{z})\psi(0, \bar{z}, \mathcal{Y}_R, \theta)$ where $\bar{z}$ is the steady state solution of the zero dynamics, determined uniquely from $\bar{\zeta} = T_1(\mathcal{Y}_r, \bar{z})$.

Assumption 3 *There exists a constant nonsingular matrix S, possibly dependent on θ, such that $Sw_r(t) = [w_{r1}(t)\ 0]^T$ where w_{r1} is persistently exciting.*

The possibility that w_r is persistently exciting or that $w_r = 0$ is not excluded. Using the transformation S^{-1} to transform $\tilde{\theta}$ into $\tilde{\theta}^T S^{-1} = [\tilde{\theta}_1^T, \tilde{\theta}_2^T]$, the equation for $\dot{e}$ and $\dot{\tilde{\theta}}$ can be rewritten as

$$\begin{aligned}
\dot{e} &= A_m e - b\tilde{\theta}^T S^{-1} S w_r + b\tilde{\theta}^T(w_r - \hat{w}) + \Lambda(\cdot), \\
\begin{bmatrix} \dot{\tilde{\theta}}_1 \\ \dot{\tilde{\theta}}_2 \end{bmatrix} &= \begin{bmatrix} \Gamma_{1p} \\ \Gamma_{2p} \end{bmatrix}
\end{aligned}$$

Define

$$\bar{f}(\cdot) = f(\mathcal{Y}_r, \bar{z}), \qquad \bar{g}(\cdot) = g(\mathcal{Y}_r, \bar{z}), \qquad \bar{g}_0(\cdot) = g_0(\mathcal{Y}_r, \bar{z})$$

$$\bar{\psi}(\cdot) = \psi(0, \bar{z}, \mathcal{Y}_R, \theta), \quad \tilde{\psi}(\cdot) = \psi(0, \bar{z}, \mathcal{Y}_R, \hat{\theta}), \quad \hat{\psi}(\cdot) = \psi(\hat{e}, z, \mathcal{Y}_R, \hat{\theta})$$

It can be shown that $\dot{e}$ and $\dot{\tilde{\theta}}_1$ satisfy

$$\begin{bmatrix} \dot{e} \\ \dot{\tilde{\theta}}_1 \end{bmatrix} = \begin{bmatrix} A_m & -b\mathcal{G}w_{r1}^T \\ 2\Gamma_1 \mathcal{G} w_{r1} b^T P & 0 \end{bmatrix} \begin{bmatrix} e \\ \tilde{\theta}_1 \end{bmatrix} + \begin{bmatrix} \Lambda_s(\cdot) \\ \Lambda_e(\cdot) \end{bmatrix} \tag{2.13}$$

where

$$S^{-T}\Gamma S^{-1} = \begin{bmatrix} \Gamma_1 & \Gamma_2 \\ \Gamma_3 & \Gamma_4 \end{bmatrix}, \; \Lambda_s(\cdot) = \Lambda(\cdot) + b\tilde{\theta}^T[(\bar{f} - \hat{f}) + (\bar{g}\tilde{\psi} - \hat{g}\hat{\psi})]$$

$$\Lambda_e(\cdot) = [\Gamma_{1p} - 2\Gamma_1 \mathcal{G} w_{r1} b^T P e] \text{ and } K_{\mathcal{G}1} > \mathcal{G}(\cdot) = (\bar{g}_0 + \theta^T \bar{g})/(\bar{g}_0 + \hat{\theta}^T \bar{g}) > K_{\mathcal{G}2}$$

for some positive constants $K_{\mathcal{G}1}$ and $K_{\mathcal{G}2}$ independent of ϵ. Since f, g_0, g and ψ are Lipschitz functions in their arguments, we have

$$||\Lambda_s(\cdot)|| \leq \delta_1||e|| + \delta_2||\xi|| + \delta_3||\tilde{\zeta}||, \; ||\Lambda_e(\cdot)|| \leq \delta_4||e|| \tag{2.14}$$

for some $\delta_i \geq 0$, $i = 1, .., 4$. Consider the system

$$\begin{bmatrix} \dot{e} \\ \dot{\tilde{\theta}}_1 \end{bmatrix} = \begin{bmatrix} A_m & -b\mathcal{G}w_{r1}^T \\ 2\Gamma_1 \mathcal{G} w_{r1} b^T P & 0 \end{bmatrix} \begin{bmatrix} e \\ \tilde{\theta}_1 \end{bmatrix} \tag{2.15}$$

Using well known results from adaptive control theory (see for example [25, Section 13.4]) and the fact that w_{r1} is persistently exciting and $\mathcal{G}(\cdot)$ is bounded from below, it can be shown that (2.14) has an exponentially stable equilibrium point at the origin. Then, from the converse Lyapunov theorem, there exists a Lyapunov function $V_2(t, e, \tilde{\theta}_1)$ whose derivative along (2.13) satisfies

$$\begin{aligned} \dot{V}_2 \leq\ & -\delta_5\|e\|^2 - \delta_6\|\tilde{\theta}_1\|^2 + \delta_7\|e\|\|\xi\| + \delta_8\|\tilde{\theta}_1\|\|\xi\| \\ & + \delta_9\|e\|^2 + \delta_{10}\|e\|\|\tilde{\theta}_1\| + \delta_{11}\|e\|\|\tilde{\zeta}\| + \delta_{12}\|\tilde{\theta}_1\|\|\tilde{\zeta}\| \end{aligned} \tag{2.16}$$

for some positive constants δ_5 and δ_6 and some non-negative constants δ_i , $7 \leq i \leq 12$. The derivative of V_ξ with respect to

$$\epsilon\dot{\xi} = (A - HC)\xi - \epsilon b\mathcal{G}\tilde{\theta}_1^T w_{r1} - \epsilon K e + \epsilon\Lambda_s \tag{2.17}$$

satisfies

$$\dot{V}_\xi \leq \frac{1}{\epsilon}\|\xi\|^2 + \gamma_3\|\tilde{\theta}_1\|\|\xi\| + \gamma_4\|e\|\|\xi\| + \gamma_5\|\tilde{\zeta}\|\|\xi\| + \gamma_6\|\xi\|^2 \tag{2.18}$$

for some non-negative constants γ_i , $3 \leq i \leq 6$. Construct the Lyapunov function candidate

$$W = \alpha V + \beta V_1 + V_2 + V_\xi \tag{2.19}$$

where $\alpha > 0$ and $\beta > 0$ will be chosen later. Using the inequality

$$\dot{V} \leq -k_1\|e\|^2 + k_2\|e\|\|\xi\|$$

together with Assumption 2, (2.16), and (2.18), it can be shown that the derivative of W with respect to (2.12) satisfies

$$\dot{W} \leq - \begin{bmatrix} ||e|| \\ ||\tilde{\theta}_1|| \\ ||\tilde{\zeta}|| \\ ||\xi|| \end{bmatrix}^T M \begin{bmatrix} ||e|| \\ ||\tilde{\theta}_1|| \\ ||\tilde{\zeta}|| \\ ||\xi|| \end{bmatrix} \tag{2.20}$$

where M is given by

$$M = \begin{bmatrix} \alpha k_1 + \delta_5 - \delta_9 & -\delta_{10}/2 & (-\beta\eta_4 - \delta_{11})/2 & (-\alpha k_2 - \delta_7 - \gamma_4)/2 \\ -\delta_{10}/2 & \delta_6 & -\delta_{12}/2 & (-\delta_8 - \gamma_3)/2 \\ (-\beta\eta_4 - \delta_{11})/2 & -\delta_{12}/2 & \beta\eta_3 & -\gamma_5/2 \\ (-\alpha k_2 - \delta_7 - \gamma_4)/2 & (-\delta_8 - \gamma_3)/2 & -\gamma_5/2 & (1/\epsilon) - \gamma_6 \end{bmatrix}$$

By choosing α and β large enough and ϵ small enough, M can be made positive definite. Hence, by [25, Theorem 4.4], we conclude that

$$[||e|| \ ||\tilde{\theta}_1|| \ ||\tilde{\zeta}|| \ ||\xi||]^T \to 0, \quad \text{as } t \to \infty$$

2.4 A Robustness Property

In this section we recall from [2] robustness of the adaptive controller to unknown bounded disturbance. Consider a perturbation of (2.1):

$$y^{(n)} = f_0(\cdot) + \sum_{i=1}^{p} f_i(\cdot)\theta_i + [g_0(\cdot) + \sum_{i=1}^{p} g_i(\cdot)\theta_i]u^{(m)} + d(\cdot) \tag{2.21}$$

where $d(\cdot)$ is a disturbance term of the form

$$d(t, x, z, v, \theta) = d_f(t, x, z, \theta) + d_g(x, z, \theta)v$$

The error equation (2.3) becomes

$$\begin{aligned} \dot{e} &= A_m e + b\{Ke + f_0(\cdot) + \theta^T f(\cdot) + (g_0(\cdot) + \theta^T g(\cdot))v \\ &\quad + d(t, e + \mathcal{Y}_r, z, v, \theta) - y_r^{(n)}\} \end{aligned}$$

Suppose the disturbance d satisfies $\|d(t, e+\mathcal{Y}_r, z, \psi^s(\cdot), \theta)\| \leq d_1 \; \forall \, t \geq 0$, $e \in E$, $\mathcal{Y}_r \in Y$, $z \in Z$, and $\xi \in R^n$. Suppose further that for sufficiently small d_1 Assumptions 1 and 2 hold uniformly in d and the set Z has the property (2.10) for all d. Using the same controller of the previous section, it can be shown that, for all $(e, \hat{\theta}, \tilde{\zeta}, \xi) \in R_s$, the derivative of V satisfies

$$\dot{V} \leq -e^T Q e + k_1 \epsilon + k_d d_1 \tag{2.22}$$

for some $k_d > 0$. For every $\epsilon < \epsilon^* = k(c_3 - c_2)/k_1$ and $d_1 < k_1(\epsilon^* - \epsilon)/k_d$, $\dot{V} < 0$ on $V = c_3$. Hence, as in the previous case, it can be shown that there is $d_1^* = d_1^*(\epsilon)$ such that for all $d_1 < d_1^*$, all variables are bounded and (2.22) is satisfied for all $t \geq T(\epsilon)$, where $T(\epsilon) \to 0$ as $\epsilon \to 0$. Therefore, the mean-square tracking error is of order $O(\epsilon + d_1)$.

If Assumption 3 is satisfied in the ideal case $d = 0$, then it can be shown that, in the presence of d, the derivative of W with respect to the closed-loop system satisfies

$$\dot{W} \leq -\frac{\delta_5}{2}\|e\|^2 - \frac{\delta_6}{2}\|\tilde{\theta}_1\|^2 + c_d d_1 \tag{2.23}$$

for some $c_d > 0$. Since all signals are bounded, the mean-square tracking error and the mean-square error of $\tilde{\theta}_1$ are of order $O(d_1)$. From [2], if w_r is persistently exciting, then $\dot{W}$ satisfies

$$\dot{W} \leq -k_w W + c_d d_1 \tag{2.24}$$

for some $k_w > 0$, which shows that all variables, including the parameter error $\tilde{\theta}$, converge to a ball centered at the origin, whose size is of the order of $O\left(\sqrt{d_1}\right)$.

2.5 Robust Output Tracking

We introduce 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. Once again, we consider the perturbed system (2.19) and assume that Assumptions 1 and 2 are satisfied uniformly in $d(\cdot)$. Moreover, we assume that the set Z has the property (2.10) in the presence of d. Let

$$v = \frac{-Ke + y_r^{(n)} - f_0(e + \mathcal{Y}_r, z) - \hat{\theta}^T f(e + \mathcal{Y}_r, z) + v_1}{g_0(e + \mathcal{Y}_r, z) + \hat{\theta}^T g(e + \mathcal{Y}_r, z)} \tag{2.25}$$

We will choose the robustifying component v_1 using the Lyapunov redesign technique, e.g., [25, Section 13.1]. Assume that

$$\|d(t, x, z, v, \theta)\| \leq \rho(e, z) + k_v \|v_1\|, \quad 0 \leq k_v < 1$$

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

$$\psi_r(e, z) = \begin{cases} -\frac{\eta(e,z)}{(1-k_v)} \cdot \frac{s}{|s|} & \text{for } \eta(e, z)|s| \geq \mu \\ \\ -\frac{\eta^2(e,z)}{(1-k_v)} \cdot \frac{s}{\mu} & \text{for } \eta(e, z)|s| < \mu \end{cases} \tag{2.26}$$

and [4]

$$\psi(e, z, \mathcal{Y}_R, \hat{\theta}) = \frac{-Ke + y_r^{(n)} - f_0(e + \mathcal{Y}_r, z) - \hat{\theta}^T f(e + \mathcal{Y}_r, z) + \psi_r(e, z)}{g_0(e + \mathcal{Y}_r, z) + \hat{\theta}^T g(e + \mathcal{Y}_r, z)} \tag{2.27}$$

The adaptive controller is taken as $v = \psi^s(\hat{e}, z, \mathcal{Y}_R, \hat{\theta})$. Using Lyapunov redesign ideas and the same adaptation law as before, it can be shown that

$$\dot{V} \leq -e^T Q e + k_c \epsilon + \frac{\mu}{4} \tag{2.28}$$

[4]This definition of ψ replaces (2.6) for the current case. Quantities defined in terms of ψ, like ϕ, S, $\hat{w}$, and w_r are now defined in terms of the new ψ. Notice, however, that w_r remains the same because ψ_r vanishes at $e = 0$.

for all $(e, \hat{\theta}, \tilde{\zeta}, \xi) \in R_s$, for some $k_c \geq 0$. Repeating the argument described in Section 2.3, it can be shown that there exist $\epsilon^* > 0$ and $\mu^* > 0$ such that for all $0 < \epsilon < \epsilon^*$ and $0 < \mu < \mu^*$, all state variables are bounded and there is time $T(\epsilon)$, with $T(\epsilon) \to 0$ *as* $\epsilon \to 0$, such that $(e, \hat{\theta}, \tilde{\zeta}, \xi) \in R_s$ for all $t \geq T(\epsilon)$. Consequently, (2.28) is satisfied for all $t \geq T(\epsilon)$. Therefore, the mean-square tracking error is of order $O(\mu + \epsilon)$ where the design parameters μ and ϵ can be made arbitrarily small.

If Assumption 3 is satisfied in the ideal case $d = 0$, inequalities similar to (2.23) and (2.24) can be shown in the current case. The right-hand side of such inequalities will have a term proportional to the disturbance upper bound despite the presence of the robustifying control component. Thus, such analysis does not reveal an advantage for the robustifying control. The only advantage we can show is the fact that the mean square tracking error can be made of the order $O(\mu + \epsilon)$.

Finally, in the ideal case $d = 0$, the controller with the robustifying component recovers the tracking-error convergence property of Section 2.3, provided Assumption 2.3 is satisfied. This can be seen by noting that in the ideal case $\dot{W}$ satisfies an inequality similar to (2.28) with M replaced by M_1, where M_1 equals M except for the constants γ_4, γ_6, δ_5, δ_6 and δ_7. The argument of Section 2.3 can be repeated to show that M_1 is positive definite.

The robustness results of this section and the preceding one have potential application to adaptive control of nonlinear systems using neural networks or other nonlinear function approximators. Consider a system whose input-output model is of the form $y^{(n)} = F(\cdot) + G(\cdot)u^{(m)}$. Using neural networks, the nonlinear functions $F(\cdot)$ and $G(\cdot)$ can be approximated to any desired tolerance. In the special case of linear-in-the-weights neural networks, as in radial basis function networks, the functions F and G can be represented by $F(\cdot) = \sum_{i=1}^{p_1} h_i(\cdot)V_i + \delta_1(\cdot)$, $G(\cdot) = \sum_{i=1}^{p_2} h_i(\cdot)W_i + \delta_2(\cdot)$ for some weights V_i and W_i. It follows that the system can be represented in the form (2.21) with $d = \delta_1 + \delta_2 u^{(m)}$. The discussion of adaptive control using RBF networks is taken up in Chapter 4.

Chapter 3

Longitudinal Control of a Platoon of Vehicles

3.1 Introduction

The subject of design and analysis of various longitudinal control laws for automated highway systems (AHS) has been studied extensively since the late 1960's. The goal is to significantly increase the traffic capacity of existing highways through vehicle and roadway automation. Furthermore, since many of today's automobile accidents are caused by human error, automating the driving process may actually increase highway safety. In such a system, vehicles will be driven automatically with onboard lateral and longitudinal controllers. The lateral controller will be used to steer the vehicle around corners, make lane changes, and perform additional steering tasks. The longitudinal controller will be used to maintain a steady velocity if the vehicle is traveling alone (conventional cruise control) or follow a lead vehicle at a safe distance (vehicle following). In this chapter, we discuss the application of the adaptive control technique of Chapter 2 to the vehicle following problem. A simplified nonlinear longitudinal powertrain model is used for designing the controller. The vehicle parameters are partially known or completely unknown and are adapted for. We assume that the following measurements are available to the vehicle's sensors (i) the relative distance [1] between the controlled car and the car in front of it and (ii) the forward velocity of the controlled car. The other quantities of interest, namely the relative velocity, relative acceleration and the acceleration/deceleration of the controlled car, are estimated from the measured quantities. The idea of replacing measured quantities by their estimates has also been used in earlier works. For example [37] mentions the possibility of "direct computation" of relative velocity and acceleration using the measured value for the relative distance. However, in [37] (i) the control objective is different from the one we consider here, (ii) the model used is a simplified one where all parameters are assumed exactly known and there are no disturbances and (iii) no analysis is presented for the case where estimates are used in feedback. Similarly, [1] uses an estimate of the leading vehicle's acceleration in the control. However, the measured quantities still include (in addition to the relative distance and the controlled vehicle's velocity) the relative velocity between the controlled and leading vehicles, and the acceleration/deceleration and propulsion force of the controlled vehicle.

[1] Referred to as the intervehicle spacing in the next section.

3.2 Longitudinal Vehicle Model

A widely proposed stategy for effectively increasing traffic throughput on existing highways through automation is to group the controlled vehicles into tightly spaced vehicle group formations called **platoons** [43]. A configuration of a platoon of N+1 vehicles is shown in Fig 3.1. The lead vehicle is numbered 0 and the ith follower (henceforth referred to as the ith vehicle) is numbered i. L_i denotes the length of the ith vehicle and x_i its position. Let $\delta_i = x_{i-1} - x_i - L_i$ for i = 1,2, ... , N. δ_i is the intervehicle spacing between the $(i-1)$th and ith vehicles. In developing a model

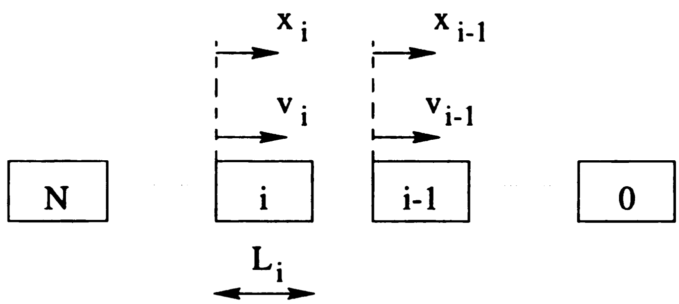

Figure 3.1: A platoon of N+1 vehicles

for the system, we assume that the road surface is horizontal and that all vehicles travel in the same direction at all times. From Newton's Second Law, the relationship between the acceleration of the ith vehicle, its propulsion force, and the drag forces acting on it can be derived as

$$m_i \ddot{x}_i = f_i - k_{d_i} \dot{x}_i^2 - d + d_{1_i}(t) \tag{3.1}$$

where m_i is the mass of the vehicle, $\ddot{x}_i$ its acceleration, f_i the propulsion force, $k_{d_i}\dot{x}_i{}^2$ the aerodynamic drag force, d a nominal constant mechanical drag and $d_{1_i}(t)$ the resultant of the external disturbances (such as wind gust,...etc.) The propulsion system which represents the engine dynamics of the vehicle can be modelled as a first order system [1]

$$\dot{f}_i = \frac{1}{\tau_i}(-f_i + u_i) + d_{2_i}(t) \tag{3.2}$$

where τ_i denotes the vehicle's engine time-constant, u_i is the throttle/brake input and $d_{2_i}(t)$ is a disturbance term (possibly due to engine transmission variations, ... etc.) This model differs from the one in [1] in that both the engine time-constant and the mechanical drag term are independent of the vehicle's velocity. However, we note that the effects of neglecting this dependence can be incorporated into the disturbance terms $d_{1_i}(t)$ and $d_{2_i}(t)$. The constants k_{d_i}, m_i and τ_i are unknown but belong to known compact subsets of R^1.

3.3 Control Objective and Design

The dynamics of the ith vehicle maybe described by the state vector $[\delta_i, v_i, f_i]^T$, where $v_i = \dot{x}_i$ is the ith vehicles's velocity. With this choice of state variables, (3.1) and (3.2) maybe rewritten as

$$\left.\begin{aligned} \dot{\delta}_i &= v_{i-1} - v_i, \\ \dot{v}_i &= (f_i - k_{d_i}v_i{}^2 - d + d_{1_i}(t))/m_i \\ \dot{f}_i &= (-f_i + u_i)/\tau_i + d_{2_i}(t) \end{aligned}\right\} \tag{3.3}$$

for $1 \leq i \leq N$. The control objective is to design u_i in such a way that the intervehicle spacing δ_i tracks a desired reference. It is well known (see for example [36, 38]) that for the case where the desired intervehicle spacing is constant, asymptotic platoon stability can be guaranteed only if the lead vehicle is transmiting its velocity and acceleration to all other vehicles in the platoon. This approach yields stable platoons with small intervehicle spacings at the cost of introducing and maintaining continuous intervehicle communication with high reliability and small delays. In [8], it is shown that platoon stability can be recovered in a non-cooperative or autonomous operation if a speed dependent spacing policy is adopted, which incorporates a constant time headway in addition to the constant distance. This takes the form $\delta_{d_i} = \lambda v_i + \lambda_0$, where δ_{d_i} is the desired intervehicle spacing and λ and λ_0 are suitably chosen positive constants. The introduce more spacing between the ith and $(i-1)$th vehicles as the velocity of the ith vehicle increases, which intuitively makes sense. Following [39], we set λ_0 to zero, which basically allows for the minimum desired distance between two adjacent vehicles to be zero provided the vehicle that is following has zero velocity. With this choice, we define the plant output as $y_{p_i} = \delta_i - \lambda v_i$. The control objective is thus to regulate y_{p_i} to zero. Differentiating the output twice and making use of (3.3), the following error equation is obtained

$$\ddot{y}_{p_i} = \ddot{\delta}_i + {\theta_i}^T[F_i(v_i, \dot{v}_i) + Gu_i] + D_i(t) \tag{3.4}$$

where $\theta_i = [k_{d_i}/m_i, 1/\tau_i, k_{d_i}/(m_i\tau_i), 1/(m_i\tau_i)]^T$, $F_i(\cdot) = [2\lambda v_i\dot{v}_i, \lambda\dot{v}_i, \lambda {v_i}^2, \lambda d]^T$, $G = [0, 0, 0, -\lambda]^T$ and $D_i(t) = -\lambda(d_{1_i}/\tau_i + \dot{d}_{1_i} + d_{2_i})/m_i$. From the knowledge of the intervals in which k_{d_i}, m_i and τ_i lie, it is possible to calculate the compact subset of R^4 to which θ_i belongs. In particular, suppose that $k_{d_i} \in [k_{d_i}^m, k_{d_i}^M]$, $m_i \in [m_i^m, m_i^M]$, and $\tau_i \in [\tau_i^m, \tau_i^M]$. Then

$$\theta \in \Omega \stackrel{\text{def}}{=} [\frac{k_{d_i}^m}{m_i^M}, \frac{k_{d_i}^M}{m_i^m}] \times [\frac{1}{\tau_i^M}, \frac{1}{\tau_i^m}] \times [\frac{k_{d_i}^m}{m_i^M \tau_i^M}, \frac{k_{d_i}^M}{m_i^m \tau_i^m}] \times [\frac{1}{\tau_i^M m_i^M}, \frac{1}{\tau_i^m m_i^m}] \subset R^4.$$

By defining $Y_{p_i} = [y_{p_i}, \dot{y}_{p_i}]^T$, it is possible to rewrite (3.4) as

$$\dot{Y}_{p_i} = A_m Y_{p_i} + b\{KY_{p_i} + \ddot{\delta}_i + \theta_i^T [F_i(v_i, \dot{v}_i) + Gu_i] + D_i(t)\},$$

where (A, b) is a controllable canonical pair that represents a chain of two integrators and K is chosen such that $A_m = A - bK$ is Hurwitz. To estimate $\dot{\delta}_i$, $\ddot{\delta}_i$ and $\dot{v}_i$, we use two high-gain observers, driven by δ_i and v_i respectively. Denote the estimates by $\hat{\dot{\delta}}_i$, $\hat{\ddot{\delta}}_i$ and $\hat{\dot{v}}_i$ respectively. The high-gain observers are described by the following equations

$$\left.\begin{aligned} \dot{w}_{1_i} &= w_{2_i} + \beta_1(\delta_i - w_{1_i})/\epsilon \\ \dot{w}_{2_i} &= w_{3_i} + \beta_2(\delta_i - w_{1_i})/\epsilon^2 \\ \dot{w}_{3_i} &= \beta_3(\delta_i - w_{1_i})/\epsilon^3 \\ \hat{\dot{\delta}}_i &= w_{2_i}, \ \hat{\ddot{\delta}}_i = w_{3_i} \end{aligned}\right\} \text{ and } \left.\begin{aligned} \dot{z}_{1_i} &= z_{2_i} + \alpha_1(v_i - z_{1_i})/\epsilon \\ \dot{z}_{2_i} &= \alpha_2(v_i - z_{1_i})/\epsilon^2 \\ \hat{\dot{v}}_i &= z_{2_i} \end{aligned}\right\}$$

where $\epsilon > 0$ and the α's and β's are chosen such that the roots of $s^2 + \alpha_1 s + \alpha_2 = 0$ and $s^3 + \beta_1 s^2 + \beta_2 s + \beta_3 = 0$ have negative real parts. Let $\hat{\dot{y}}_{p_i} = \hat{\dot{\delta}}_i - \lambda \hat{\dot{v}}_i$, $\hat{Y}_{p_i} = [y_{p_i}, \hat{\dot{y}}_{p_i}]^T$ and assume that an upper bound on the term $D_i(t)$ is known. Then the control u_i is designed as

$$u_i = \frac{-\hat{\ddot{\delta}}_i - \hat{\theta}_i^T F_i(v_i, \hat{\dot{v}}_i) - K\hat{Y}_{p_i} + v_{r_i}}{\hat{\theta}_i^T G} \tag{3.5}$$

where $\hat{\theta}_i$ is an estimate of θ_i and v_{r_i} is a robustifying component designed using the Lyapunov redesign technique,e.g., [25, Section 13.1]. The control u_i is saturated outside a compact set of interest to prevent the peaking induced by the high-gain observers [9]. [2] The parameter adaptation law is chosen as in Chapter 2.

Proceeding along the lines of Chapter 2, it is possible to show ultimate boundedness of the spacing deviation error. We use the Lyapunov function candidate

$$V_i = Y_{p_i}^T P Y_{p_i} + \tfrac{1}{2}\tilde{\theta}_i^T \Gamma^{-1} \tilde{\theta}_i,$$

where $P = P^T > 0$ is the solution of the Lyapunov equation $PA_m + A_m^T P = -I$, $\tilde{\theta}_i = \hat{\theta}_i - \theta_i$ the parameter estimation error and $\Gamma = \Gamma^T > 0$ the "adaptation gain". Though the proof in Chapter 2 was done for the single-output case, an extension to the multi-output case is not very difficult, and has been addressed, for example, in [28]. It is worth mentioning that the proof in Chapter 2 guarantees only the boundedness of y_{p_i} and $\dot{y}_{p_i}$. To argue boundedness of δ_i, $\dot{\delta}_i$, $\ddot{\delta}_i$, v_i and $\dot{v}_i$, we first assume that there exist achievable bounds on the leading vehicle's velocity v_0 [39] and acceleration $\dot{v}_0$ [8]. Noting that

$$\lambda \dot{v}_1 + v_1 = v_0 - \dot{y}_{p_1}$$

and that $\dot{y}_{p_1}$ is bounded and $\lambda > 0$, we see that v_1 is bounded. Extending this

[2]For the purpose of simulations, the control is saturated at a value slightly higher than the observed value under state feedback.

argument inductively shows that v_i is bounded for all i. Since $y_{p_i} = \delta_i - \lambda v_i$, boundedness of δ_i follows. Furthermore, since each v_i is bounded, so is $\dot{\delta}_i = v_{i-1} - v_i$. From $\dot{y}_{p_i} = \dot{\delta}_i - \lambda \dot{v}_i$, each $\dot{v}_i$ is bounded. And finally, since $\ddot{\delta}_i = \dot{v}_{i-1} - \dot{v}_i$, each $\ddot{\delta}_i$ is bounded.

3.4 Simulations

In this section, we present two sets of simulations for a platoon of five cars. In all simulations, we assume that all vehicles are initially travelling at a velocity of 15 m/s. The lead vehicle's velocity, acceleration and jerk profiles are shown in Fig 3.2. We assume that $m_i \in [1100, 1550]kg$, $\tau_i \in [0.15, 0.25]s$, $k_{d_i} \in [0.1, 0.5]Ns^2/m^2$ and $d = 100N$. These values arechosen to be the same as or close to the ones in [1, 39]. For the first set of simulations, we use a value of $\lambda = 0.9$ and for the second $\lambda = 0.2$. The particular values for λ are explained in some detail below.

3.4.1 Simulation 1

The value of $\lambda = 0.9$ is based on the *California rule of thumb*, [8, 39], which suggests an intervehicle spacing of one vehicle length for every 10 m.p.h. Assuming an average vehicle length of 4 m, this translates to a value of $\lambda = 0.9$ In all simulations, we assume the following values for the vehicle parameters, $m_1 = 1300$, $\tau_1 = 0.16$, $k_{d_1} = 0.3$, $m_2 = 1400$, $\tau_2 = 0.22$, $k_{d_2} = 0.35$, $m_3 = 1200$, $\tau_3 = 0.18$, $k_{d_3} = 0.2$, $m_4 = 1350$, $\tau_4 = 0.24$ and $k_{d_4} = 0.45$. For the first simulation, we assume perfect knowledge of the vehicle parameters and that $d_{1_i}(t)$ and $d_{2_i}(t)$ are identically zero. Fig 3.3 shows the velocity and acceleration profiles for the following vehicles, the spacing deviation errors and their relative positions relative to the leader . [3] Though not clear from the figure, the spacing deviation error does not exceed 1.5 cm in magnitude. [4] Fig 3.4 is for the case where the vehicle parameters are unknown and nominal values are used in the control. The disturbance terms are still identically zero. In Fig 3.5, the vehicle parameters are unknown but are adapted for. Compared to the previous case, the spacing deviation errors show a marked decrease. Fig 3.7 is for the case where d_{2_i} is still identically zero, but d_{1_i}/m_i is as shown in Fig 3.6. The disturbance profiles are similar to, though not identical to the ones in [1]. In particular, they are "smooth" functions of time. No robustifying control is used in this case. Fig 3.8 is for the case where a robustifying control is used. The spacing deviation error shows a marked decrease in this case. The spacing deviations do not exceed 1.6 cm in magnitude. The above results compare favourably with the results of [39, 1]. It is worth mentioning however, that the spacing policy in [1] is different from the one we adopt here. The spacing deviation errors reported above are also of the same order of magnitude as in [36, 20], where the spacing deviation errors are between 1 and 10 cm. However, as with [1], the results are not directly comparable owing to differences in the vehicle model and/or the spacing policy adopted.

3.4.2 Simulation 2

The California rule of thumb takes into account human reaction times and delays [8]. In automatic vehicle following, human delays are eliminated and we can afford to

[3] This is assumed to be the distance from the front of the car to the rear of the leader. We assume that $L_1 = 3.9, L_2 = 4$ and $L_3 = 3$, all values being in meters.

[4] We assumed a non-zero initial value for the estimation error $\delta_i - \hat{\delta}_i$.

have a smaller time headway without affecting safety. In [8], based on a worst case stopping scenario, where the lead vehicle is assumed to be at full deceleration and the following vehicle is at full acceleration at the instant the stop maneuver commences, a value of λ in the range of 0.1 to 0.2 is obtained. For this simulation, we assume $\lambda = 0.2$. For brevity, we present simulation results only for the case corresponding to Fig 3.8 of the previous set of simulations, i.e, $d_{1_i}(t)$ is not zero and a robustifying component is used. Fig 3.9 shows the results for this case, and the decrease in the relative distances from the leader as we move down the platoon is clear.

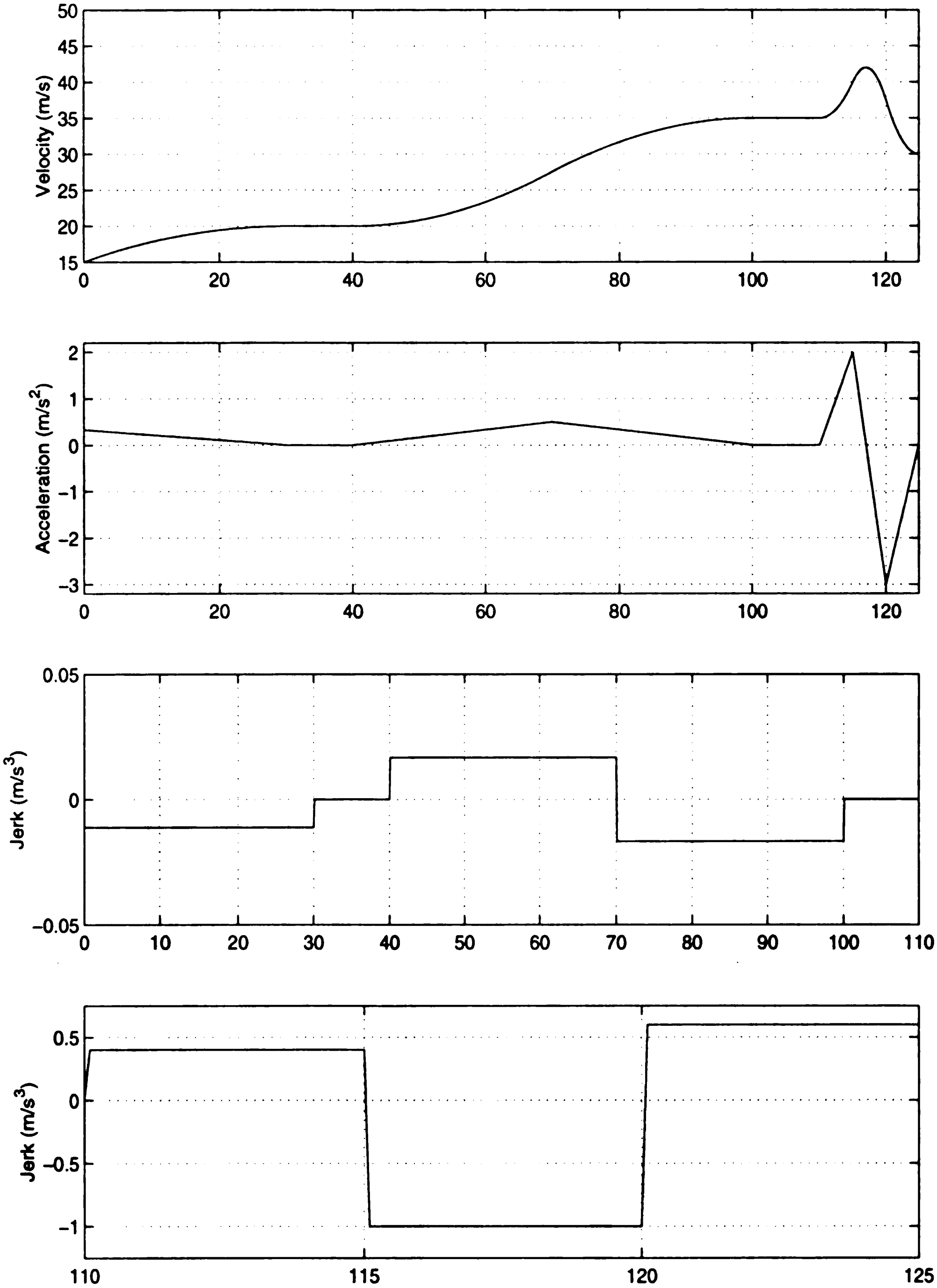

Figure 3.2: Velocity, acceleration and jerk profiles for the leader.

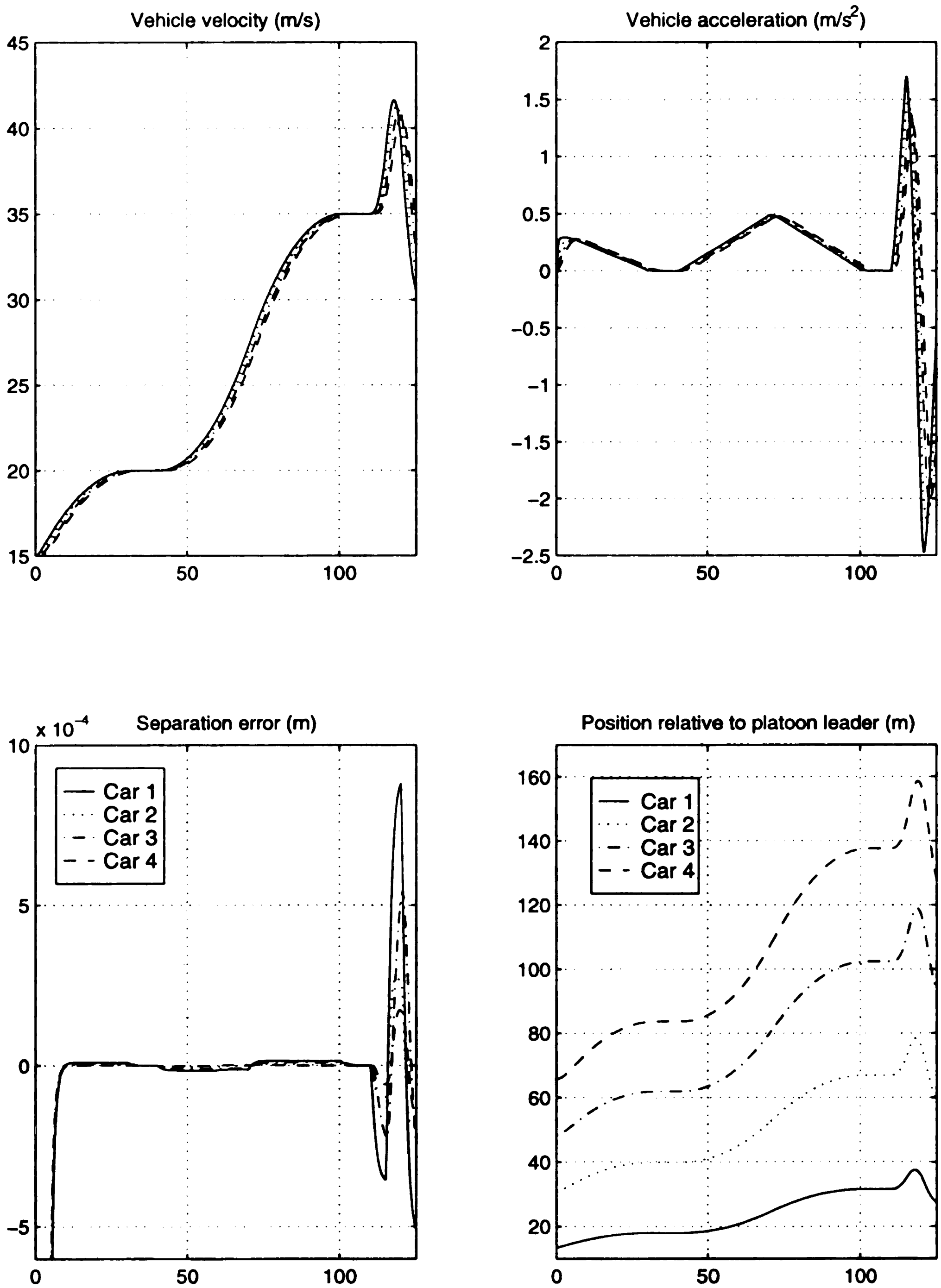

Figure 3.3: Perfect knowledge of vehicle parameters, $D_i(t) = 0$.

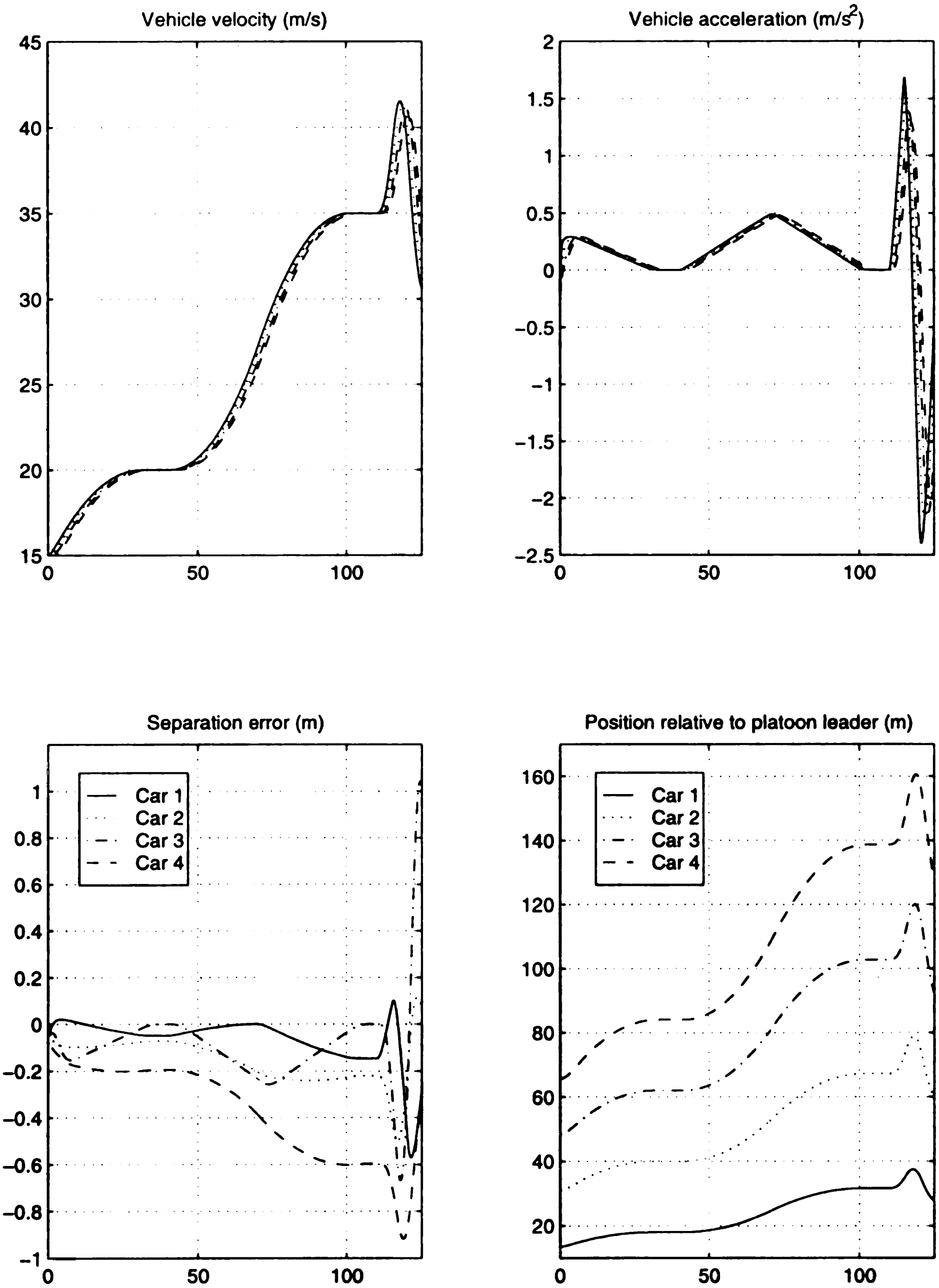

Figure 3.4: Vehicle parameters unknown, control based on nominal values, $D_i(t) = 0$.

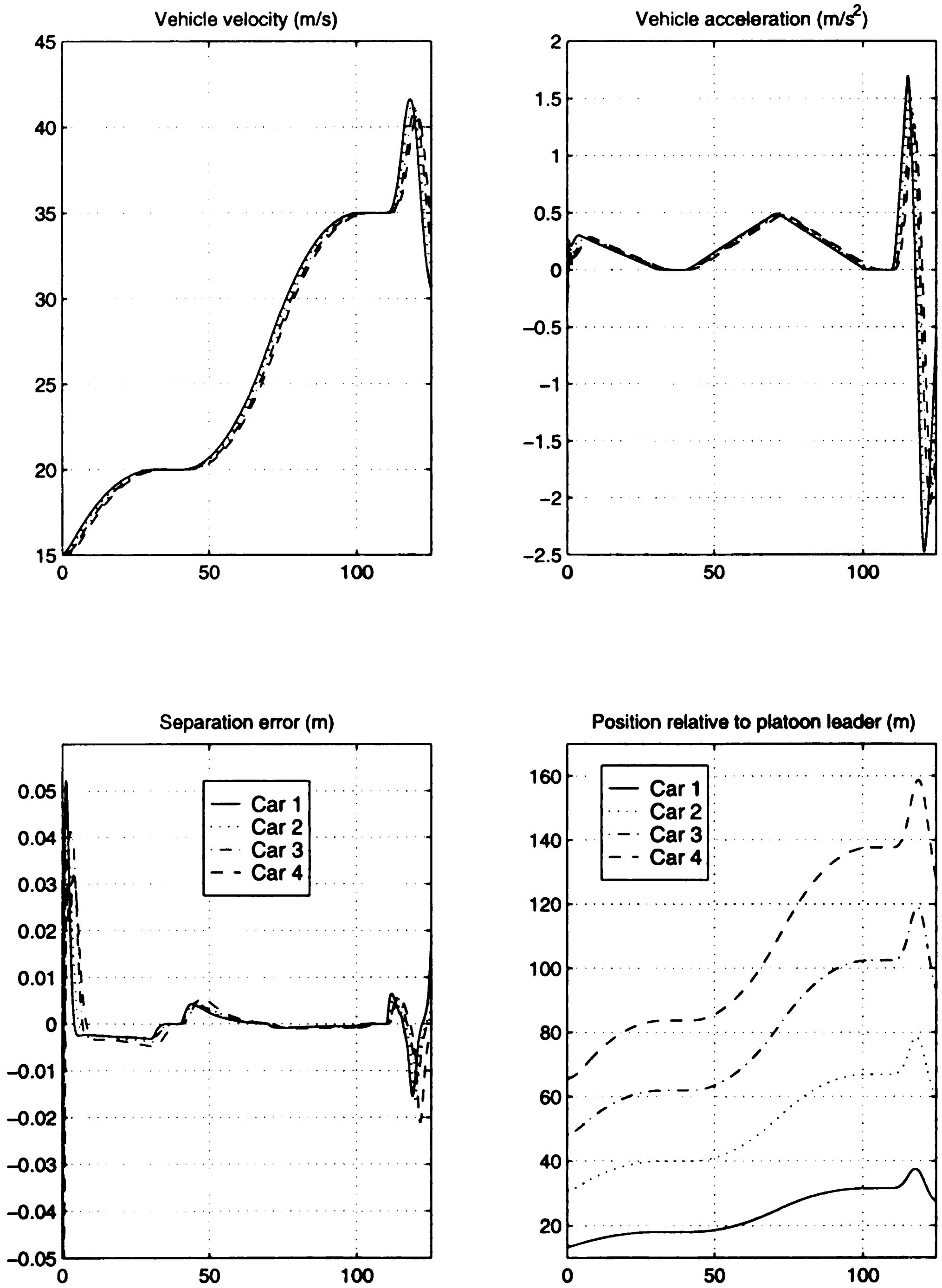

Figure 3.5: Vehicle parameters unknown but adapted for, $D_i(t) = 0$.

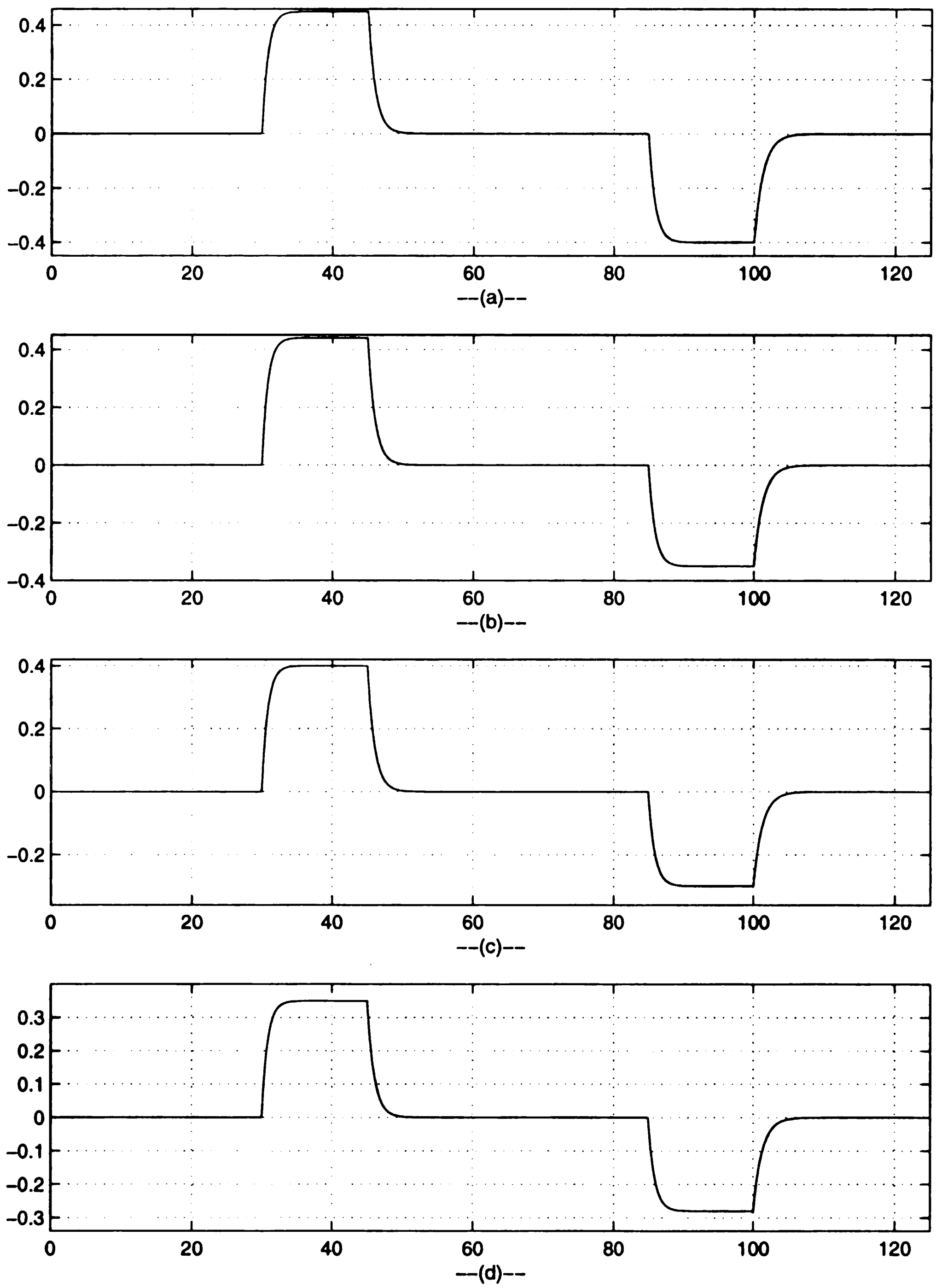

Figure 3.6: (a) $d_{1_1}(t)$, (b) $d_{1_2}(t)$, (c) $d_{1_3}(t)$, (d) $d_{1_4}(t)$; $d_{2_i}(t) = 0$.

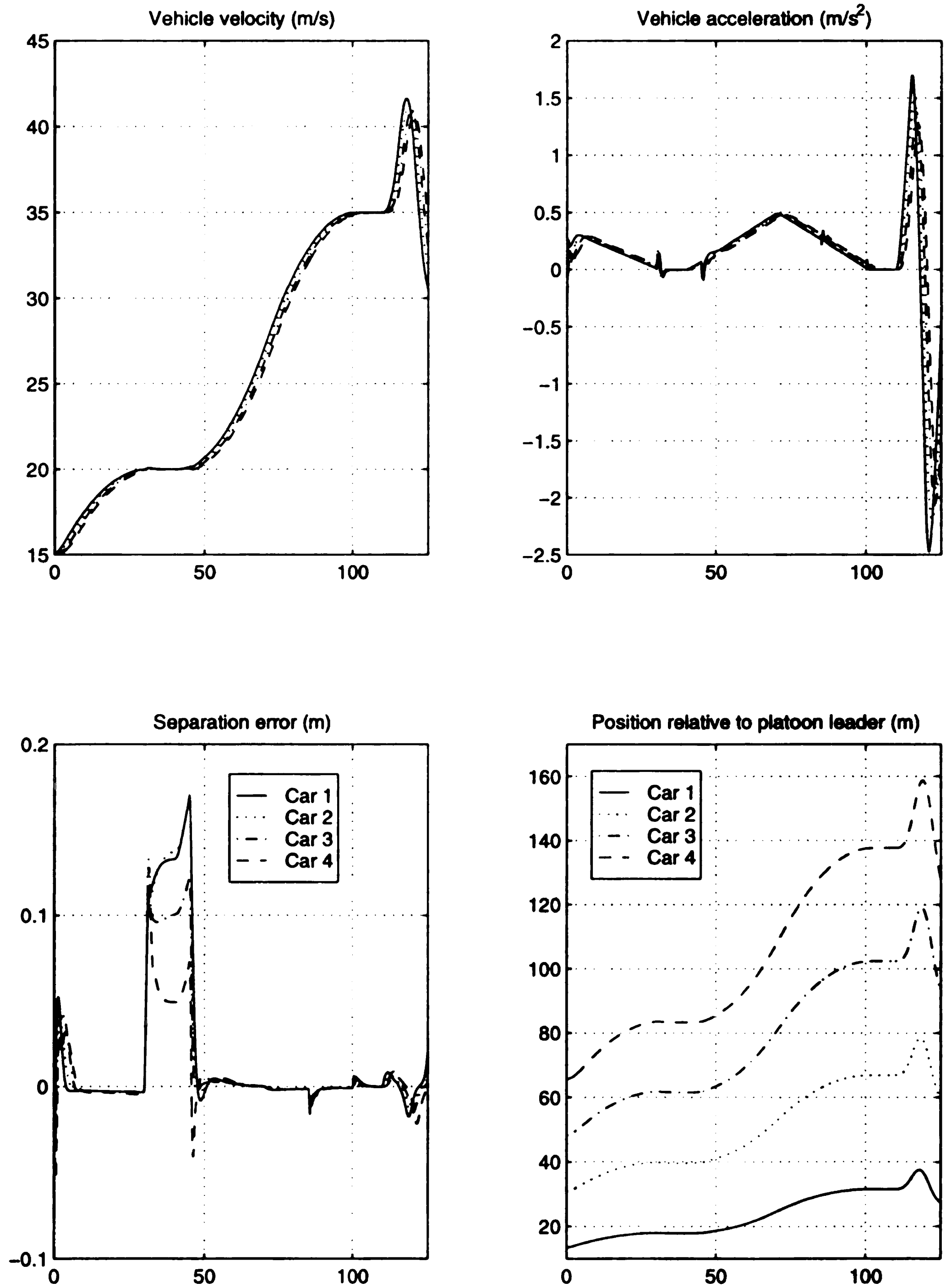

Figure 3.7: $D_i(t) \neq 0$, robustifying component not included.

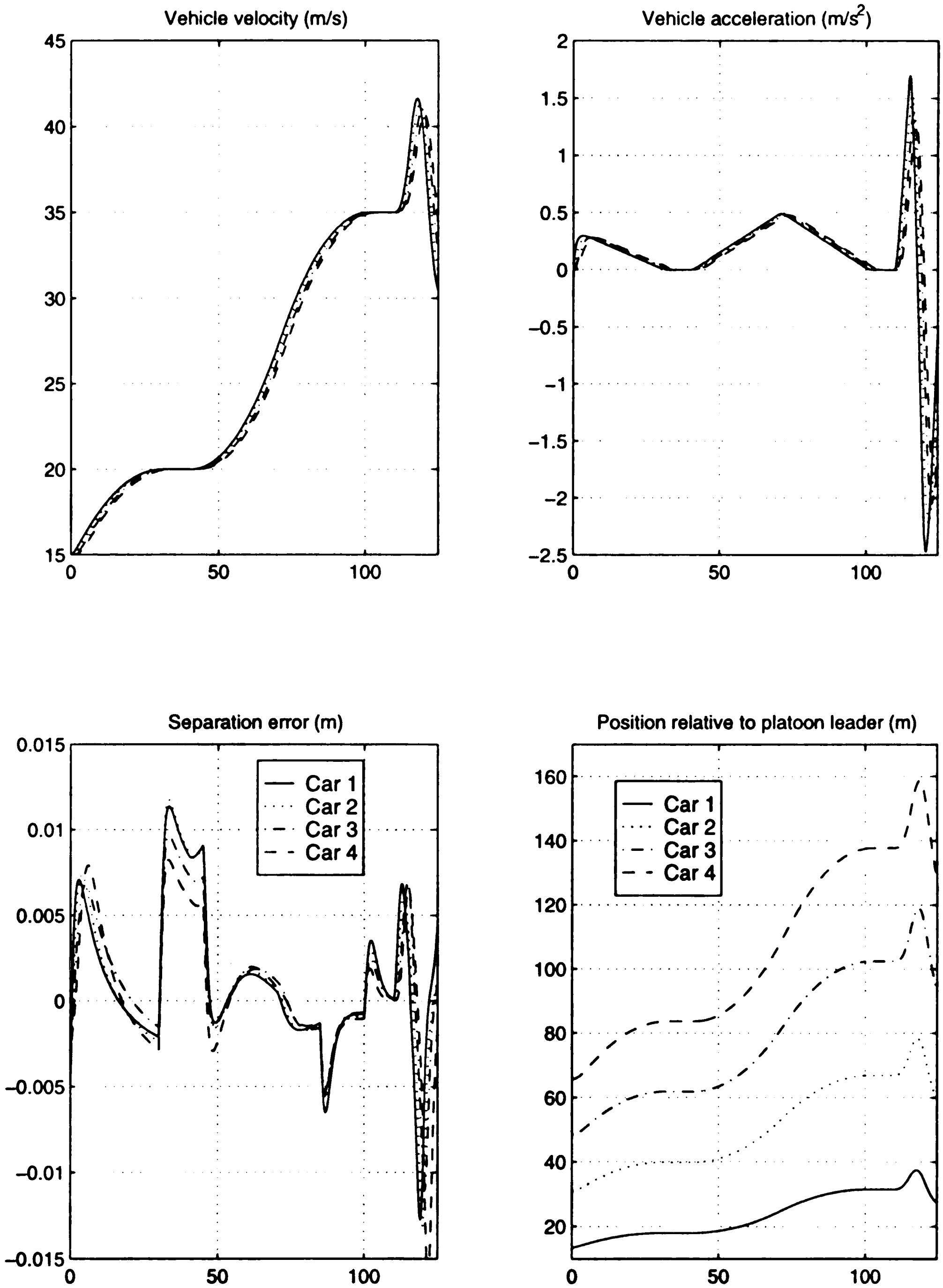

Figure 3.8: $D_i(t) \neq 0$, robustifying component included, $\lambda = 0.9$.

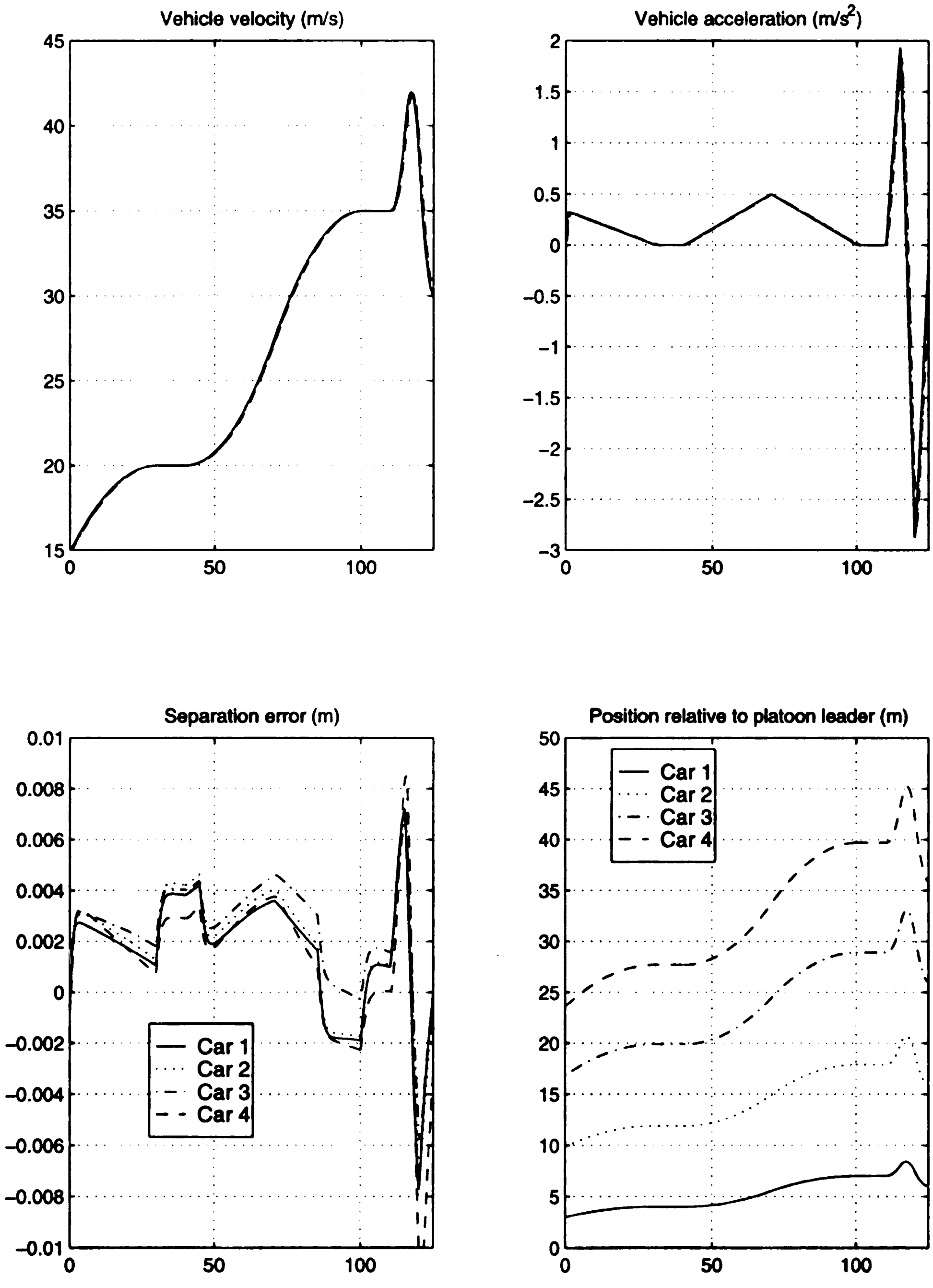

Figure 3.9: $D_i(t) \neq 0$, robustifying component included, $\lambda = 0.2$.

Chapter 4

Indirect Adaptive Contol Using RBF Neural Networks

4.1 Introduction

In recent years, the analytical study of adaptive nonlinear control systems using universal function approximators has received much attention (See [33] for references). Typically, these methods use neural networks as approximation models for the unknown system nonlinearities [3, 7, 10, 21, 22, 33, 34, 35, 42].

In section 5 of chapter 2, a mention was made of the potential application to adaptive control using neural networks. In this chapter, we investigate the use of a radial basis function (RBF) 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 CMAC (Cerebellar Model Articulation Controller) networks (See [10, 11] for references).

4.2 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, $y^{(1)}$, ..., $y^{(n-1)}$, u, $u^{(1)}$, ..., $u^{(m-1)}$. In particular,

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

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

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

$$\left.\begin{array}{rcl}
\dot{x}_i & = & x_{i+1}, \quad 1 \le i \le n-1 \\
\dot{x}_n & = & F(x,z) + G(x,z)v \\
\dot{z}_i & = & z_{i+1}, \quad 1 \le i \le m-1 \\
\dot{z}_m & = & v \\
y & = & x_1
\end{array}\right\} \tag{4.1}$$

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

Assumption 4 $|G(x,z)| \geq k_1 > 0 \ \forall \ x \in R^n$ and $z \in R^m$.

Assumption 4 ensures that (4.1) is input-output linearizable by full state feedback. As before, it can be shown that there exists a global diffeomorphism,

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

with $T_1(0,0) = 0$, which transforms the last m state equations of (4.1) into $\dot{\zeta} = H(\zeta, x)$. This, together with the first n state equations of (4.1), 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.

4.3 Function Approximation Using Gaussian Radial Basis Functions

The control design presented in this chapter 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 [35] for references). In particular, it has been shown that given a smooth function $F : \Omega \mapsto R$, where Ω is a compact subset of R^{m+n} and $\epsilon > 0$, there exist a Gaussian basis function vector $f : R^{m+n} \mapsto R^p$ and a weight vector $\theta^* \in R^p$ such that $|F(x) - \theta^{*T} f(x)| \leq \epsilon \ \forall \ x \in \Omega$. The quantity $F(x) - \theta^{*T} f(x) \stackrel{\text{def}}{=} d(x)$ 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 minimises $d(x)$ over Ω, that is,

$$\theta^* = arg \min_{\theta \in R^p} \{ \sup_{x \in \Omega} |F(x) - \theta^T f(x)| \} \tag{4.2}$$

The choice of the Gaussian network parameters used in our control design is motivated by the discussion in [35, Section III]. 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.

4.4 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 later in the chapter.

$$F(x,z) = \theta_f^{*T} f(x,z) + d_F(x,z), \quad G(x,z) = \theta_g^{*T} g(x,z) + d_G(x,z) \tag{4.3}$$

Assumption 5 *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 gradient descent technique [1] [13]. 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 (4.3) 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 6 $|\hat{G}(\cdot)| \geq k_2 > 0 \; \forall \; x \in Y, \; z \in Z$ and $\hat{\theta}_g \in \hat{\Omega}_g$, *where $\hat{\Omega}_g$ is a compact set that contains Ω_g in its interior.*

[1] For our purposes, a variant of Matlab's solverb function was used, see Appendix B

4.4.1 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. Let e, $\mathcal{Y}(t)$, $\mathcal{Y}_r(t)$ and $\mathcal{Y}_R(t)$ be defined as in Section 2.2.1 of Chapter 2 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 $\mathcal{Y}_R(t) \in Y_R \ \forall\ t \geq 0$. We rewrite (4.1) as

$$\left.\begin{aligned} \dot{e} &= A_m e + b\{Ke + \theta_f^{*T} f(e+\mathcal{Y}_r, z) + \theta_g^{*T} g(e+\mathcal{Y}_r, z)v \\ &\quad + d(e+\mathcal{Y}_r, z, \hat{\theta}_f, \hat{\theta}_g) - y_r^{(n)}\} \\ \dot{z} &= A_2 z + b_2 v \end{aligned}\right\} \tag{4.4}$$

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 7 *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*

$$\begin{aligned} \dot{\tilde{\zeta}} &= H(\bar{\zeta} + \tilde{\zeta}, e + \mathcal{Y}_r) - H(\bar{\zeta}, \mathcal{Y}_r) \\ &\stackrel{\text{def}}{=} \tilde{H}(\tilde{\zeta}, e, \mathcal{Y}_r, \bar{\zeta}) \end{aligned} \tag{4.5}$$

---

[2]The dependence on $\hat{\theta}_f$ and $\hat{\theta}_g$ comes through v. See (4.7).

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

$$\eta_1 ||\tilde{\zeta}||^2 \leq V_1(t, \tilde{\zeta}) \leq \eta_2 ||\tilde{\zeta}||^2$$

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

where $\eta_1, \eta_2, \eta_3 > 0$, *and* $\eta_4 \geq 0$ *are independent of* $\mathcal{Y}_r$.

Assumption 7 implies that (4.5), with e as input, is input-to-state stable. Consequently, the zero dynamics of (4.1) are exponentially stable and (4.1) is minimum phase.

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

$$V = e^T P e + \tfrac{1}{2} \tilde{\theta}_f^T \Gamma_f^{-1} \tilde{\theta}_f + \tfrac{1}{2} \tilde{\theta}_g^T \Gamma_g^{-1} \tilde{\theta}_g \tag{4.6}$$

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 (4.4) the derivative of V along the trajectories of the system is given by

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

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

Taking

$$v = \frac{-Ke + y_r^{(n)} - \hat{F}(e + \mathcal{Y}_r, z, \hat{\theta}_f)}{\hat{G}(e + \mathcal{Y}_r, z, \hat{\theta}_g)} \stackrel{\text{def}}{=} \psi(e, z, \mathcal{Y}_R, \hat{\theta}_f, \hat{\theta}_g) \tag{4.7}$$

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

$$\dot{V} = -e^T Q e + \tilde{\theta}_f^T \Gamma_f^{-1}[\dot{\hat{\theta}}_f - \Gamma_f \phi_f] + \tilde{\theta}_g^T \Gamma_g^{-1}[\dot{\hat{\theta}}_g - \Gamma_g \phi_g] + 2e^T Pb\, d(\cdot) \tag{4.8}$$

where

$$\phi_f = 2e^T Pbf(e + \mathcal{Y}_r, z), \;\; \phi_g = 2e^T Pbg(e + \mathcal{Y}_r, z)\psi(\cdot)$$

Let $\hat{\Omega}_f$ be a compact subset of R^{p_1} that contains Ω_f in its interior. Define

$$\theta = \begin{bmatrix} \theta_f \\ \theta_g \end{bmatrix}, \hat{\theta} = \begin{bmatrix} \hat{\theta}_f \\ \hat{\theta}_g \end{bmatrix}, \tilde{\theta} = \begin{bmatrix} \tilde{\theta}_f \\ \tilde{\theta}_g \end{bmatrix}, \phi = \begin{bmatrix} \phi_f \\ \phi_g \end{bmatrix},$$

$$\Gamma = \text{diag}[\Gamma_f, \Gamma_g], \;\; \Omega = \Omega_f \times \Omega_g \text{ and } \hat{\Omega} = \hat{\Omega}_f \times \hat{\Omega}_g$$

The parameter adaptation law is chosen as in Chapter 2. 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 $c_1 = \max_{e \in E_0} e^T P e$. Choose $c_4 > c_1$ and define $E \stackrel{\text{def}}{=} \{e^T P e \leq c_4\}$ and $Y \stackrel{\text{def}}{=} \{e + y_r | e \in E, y_r \in Y_R\}$. Let Z be a compact subset of

R^m such that Z_0 is in the interior of Z and

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

The RBF networks are used to approximate $F(\cdot)$ and $G(\cdot)$ over the compact set $Y \times Z$.

Let Ω_δ be as in Chapter 2. 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 Chapter 2 where the adaptation gain is not required to be large. This is because, in Chapter 2, the parameter vector θ has some physical meaning and the compact set Ω to which it belongs is known apriori. 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 $E \times Z$. Hence the set E 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 $d_1 = max\|d(e + \mathcal{Y}_r, z, \hat{\theta}_f, \hat{\theta}_g)\|$, where the maximization is done over all $e \in E, \mathcal{Y}_r \in Y, z \in Z, \hat{\theta}_f \in \Omega_{\delta_f}$ and $\hat{\theta}_g \in \Omega_{\delta_g}$. From (4.8) and (2.7), we have

$$\dot{V} \leq -e^T Q e + k_d d_1 \ \forall e \in E, \text{where } k_d = \max_{e \in E} 2\|e\| \ \|P\ b\| \tag{4.9}$$

If $d_1 < d^* = k(c_4 - c_3 - c_2)/k_d$, where $k = \frac{\lambda_{min}(Q)}{\lambda_{max}(P)}$, 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 $d_1 < d^*$. 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 (4.9), the mean-square tracking error is of the order $O(d_1)$.

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. 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 $e \in E_1 \stackrel{\text{def}}{=} \{e^T P e \le c_5\}$, $z \in Z$, $\mathcal{Y}_R \in Y_R$, $\hat{\theta}_f \in \Omega_{\delta_f}$, $\hat{\theta}_g \in \Omega_{\delta_g}$ 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 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 Chapter 2, i.e,

$$\left.\begin{aligned} \dot{\hat{e}}_i &= \hat{e}_{i+1} + \tfrac{\alpha_i}{\epsilon^i}(e_1 - \hat{e}_1), \quad 1 \le i \le n-1 \\ \dot{\hat{e}}_n &= \tfrac{\alpha_n}{\epsilon^n}(e_1 - \hat{e}_1) \end{aligned}\right\} \qquad (4.10)$$

where $\epsilon > 0$ is a design parameter that will be specified shortly. The positive constants α_i are chosen such that the roots of $s^n + \alpha_1 s^{n-1} + \cdots + \alpha_{n-1}s + \alpha_n = 0$ have negative real parts. Let $\xi_i = (e_i - \hat{e}_i)/\epsilon^{n-i}$, $1 \leq i \leq n$, $\xi = [\xi_1, \ldots, \xi_n]^T$ and $V_\xi = \xi^T \bar{P}\xi$, where $\bar{P} = \bar{P}^T > 0$ is the solution of the Lyapunov equation $\bar{P}(A-HC)+(A-HC)^T\bar{P} = -I$. Boundedness of all signals of the closed-loop system can be proved by an argument similar to the one in the state feedback case. First, it is not difficult to show that for all $(e, \hat{\theta}) \in \{V \leq c_4\} \cap \Omega_\delta$, there exist constants $c_6,\ c_7 > 0$ such that the sets $\{V_1 \leq c_6\}$ and $\{V_\xi \leq c_7\epsilon^2\}$ are positively invariant. Next, using the results of Section 2.4, for all $(e, \hat{\theta}, \tilde{\zeta}, \xi)$ belonging to the set $R = \{\{V \leq c_4\} \cap \Omega_\delta\} \times \{V_1 \leq c_6\} \times \{V_\xi \leq c_7\epsilon^2\}$, the derivative of V satisfies

$$\dot{V} \leq -e^T Q e + k_\epsilon \epsilon + k_d d_1, \text{ where } k_\epsilon > 0. \tag{4.11}$$

Hence for all

$$d_1 < d^* = \frac{k(c_4 - c_3 - c_2)}{2k_d} \text{ and } \epsilon < \epsilon^* = \frac{k(c_4 - c_3 - c_2)}{2k_\epsilon} \tag{4.12}$$

$\dot{V} < 0$ 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 $\dot{V}_\xi \leq -(1/2\epsilon)||\xi||^2$ outside $\{V_\xi \leq c_7\epsilon^2\}$ 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 (4.11) is satisfied for all $t \geq T(\epsilon)$. Hence the mean-square tracking error is of the order $O(\epsilon + d_1)$.

4.4.2 Reconstruction Error With a Known Bound

As in Section 2.5, we design an additional robustifying control component to make the mean-square tracking error arbitrarily small, irrespective of the bound on the disturbance $d(\cdot)$, provided this bound is known. Let

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

Assume that

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

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

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

and

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

The adaptive controller is taken as $v = \psi^s(\hat{e}, z, \mathcal{Y}_R, \hat{\theta}_f, \hat{\theta}_g)$. As in Chapter 2, it can be shown that $\exists\ \epsilon^* > 0$ and $\mu^* > 0$ such that $\forall\ 0 < \epsilon < \epsilon^*$ and $0 < \mu < \mu^*$, all signals are bounded and the mean-square tracking error is of the order $O(\epsilon + \mu)$, where the design parameters ϵ and μ can be made arbitrarily small.

4.5 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. Lastly, 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 [35, 42], namely

$$\begin{aligned} \ddot{y} &= F(y,\dot{y}) + G(y)u, \text{ where} \\ F(y,\dot{y}) &= 16\frac{sin(4\pi y)}{4\pi y}\left(\frac{sin(\pi\dot{y})}{\pi\dot{y}}\right)^2 \text{ and} \\ G(y) &= 2 + sin(3\pi(y-0.5)) \end{aligned}$$

4.5.1 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 $(1+s/10)^{-3}$, 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 4.1. As can be seen, the set $\mathcal{Y}_R$ can be taken as $[0,1]\times[-3,3]\times[-25,25]$. Since $m = 0$, there is no need to augment integrators at the system's input. Let $\tilde{Y} \stackrel{\text{def}}{=} [-1,1]\times[-3,3]$. We use 2 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. Based on the values of θ_{f_0} and θ_{g_0}, the sets Ω_f and Ω_g in Assumption 4 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 values of the various design parameters are $\delta = 0.001$, $\eta = 40$ and $k_v = 0.7$. The initial condition $x(0)$ is taken as (-0.5,2.0). Fig 4.2(a) shows the tracking error for the state feedback case with $\mu = 0.5$, Fig 4.2(b) is the output feedback case with $\epsilon = 10^{-3}$, Fig 4.2(c) with ϵ reduced to 10^{-4}, and Fig 4.2(d) with μ reduced to 0.1. Fig 4.3 shows the corresponding control inputs. The simulation illustrates several points : (a) by using a robustifying component, it is possible to obtain reasonable performance even with networks that give large reconstruction errors; (b) as ϵ is decreased, we recover the performance obtained under state feedback; and (c) an n-fold decrease in μ results in approximately an n-fold decrease in the tracking error. Thus, by decreasing μ, we can meet more stringent requirements on the tracking error.

4.5.2 Simulation 2

The initial weights obtained by off-line training may not be close to their optimal values. This might, for example, be the case when the off-line training is done (based)

[4] See [35] for a definition of this term in relation to RBF networks

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 4.4 shows the function $F(y,\dot{y})$ and the error $F(\cdot) - {\theta_{f_0}}^T f(\cdot)$ that results from using the nominal

[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.

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 4.5(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 4.5(b) for the case when the weights are adapted but there is no robust component, Fig 4.5(c) for the case when the weights are not adapted but there is a robust component, and Fig 4.5(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.

4.5.3 Simulation 3

In Section 4.4.2 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 unmodelled dynamics. Decreasing μ implies a "high-gain like" feedback inside the layer $|s| < \frac{\mu}{\eta}$ which might result in the excitation of the unmodelled 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 $\sqrt{\frac{\sum e_i{}^2}{n}}$ 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 $0.4+0.1sin(t)$. 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 4.6 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

[6] As observed by simulation

capabilities. Since the error is of the order $O(\epsilon + d)$, 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. Lastly, the performance in the last case is the best of the three cases.

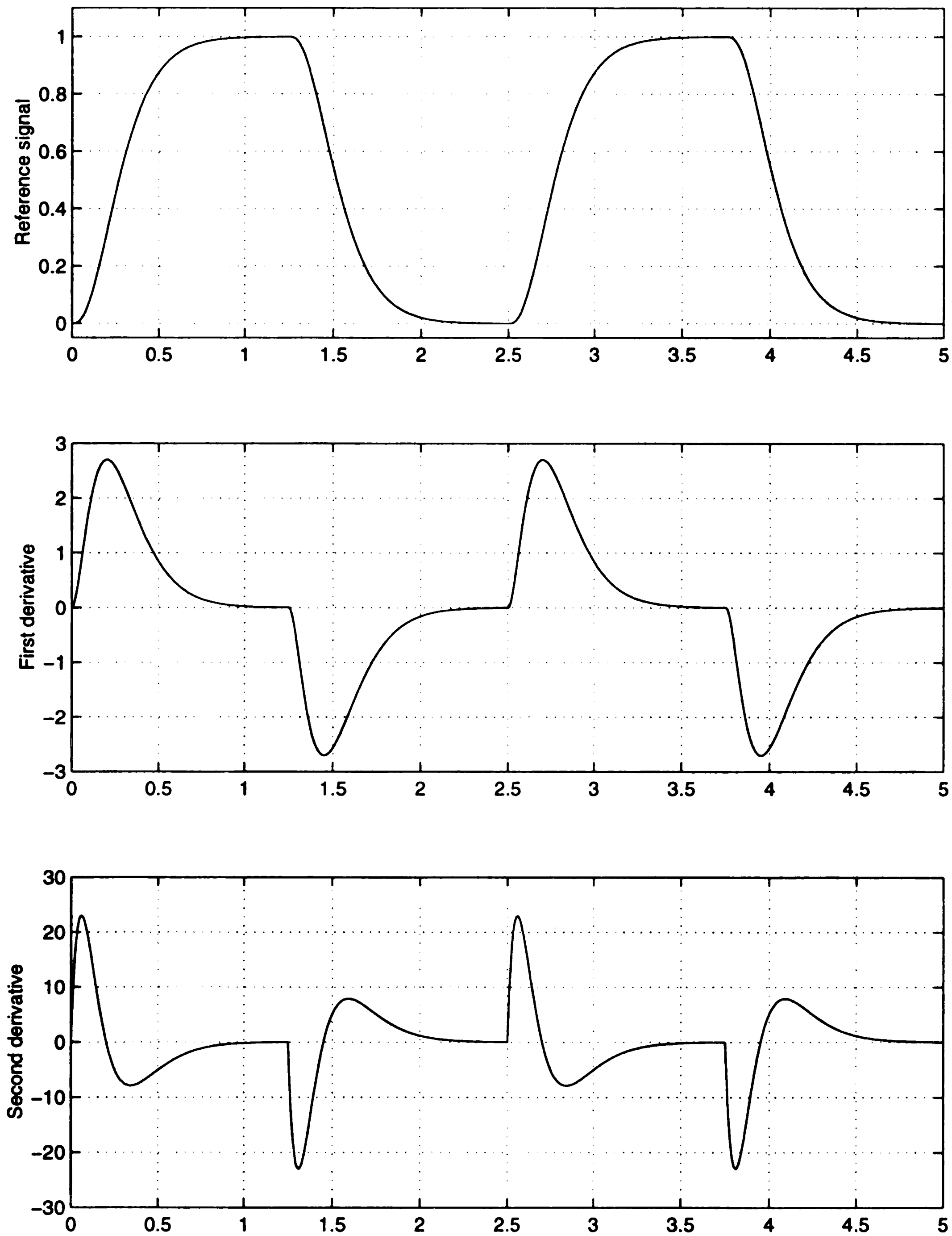

Figure 4.1: The reference signal and its derivatives.

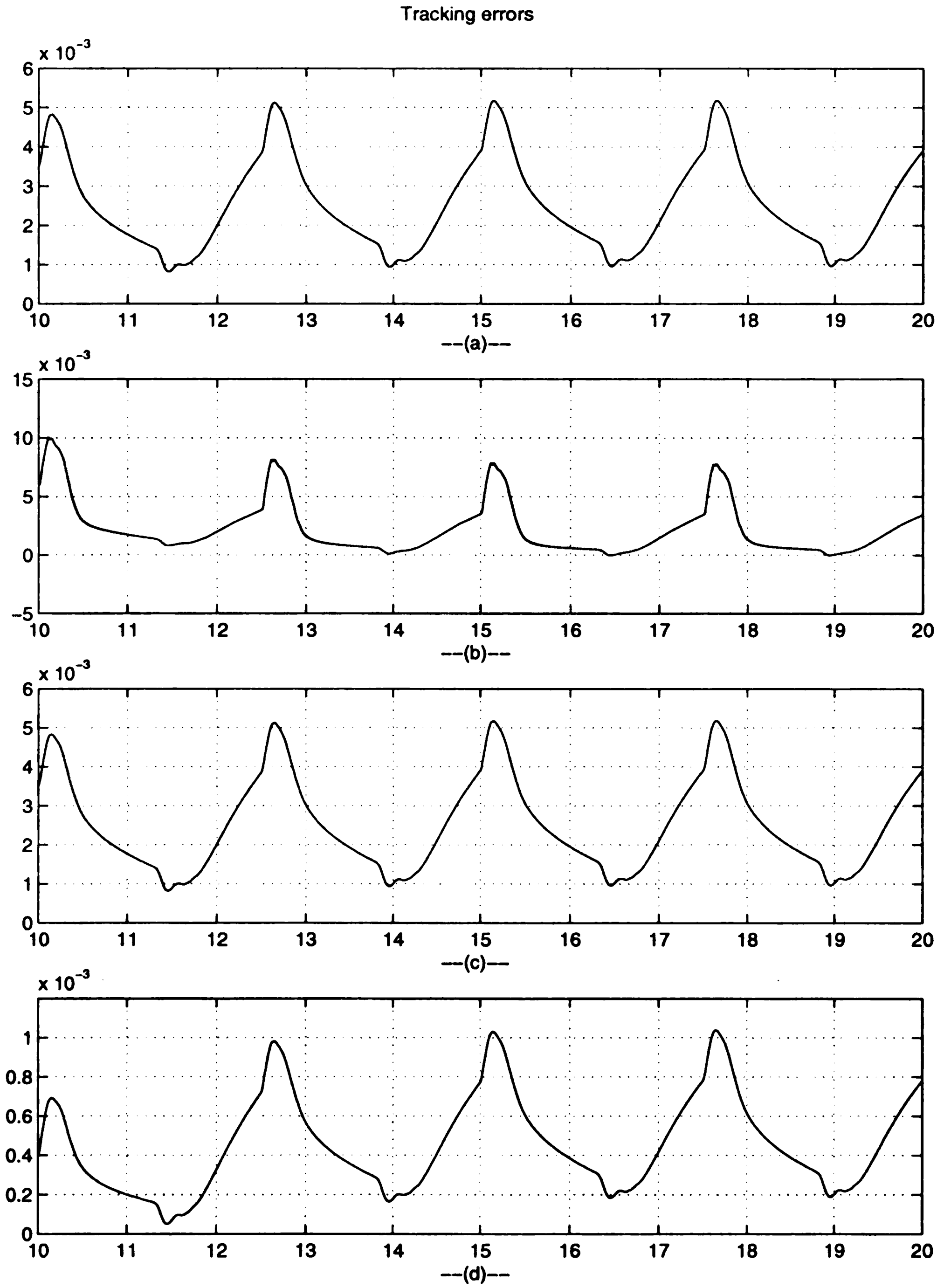

Figure 4.2: (a)State feedback $\mu = 0.5$ (b) Output feedback, $\epsilon = 10^{-3}$, $\mu = 0.5$ (c) Output feedback $\epsilon = 10^{-4}$, $\mu = 0.5$ (d) Output feedback $\epsilon = 10^{-4}$, $\mu = 0.1$

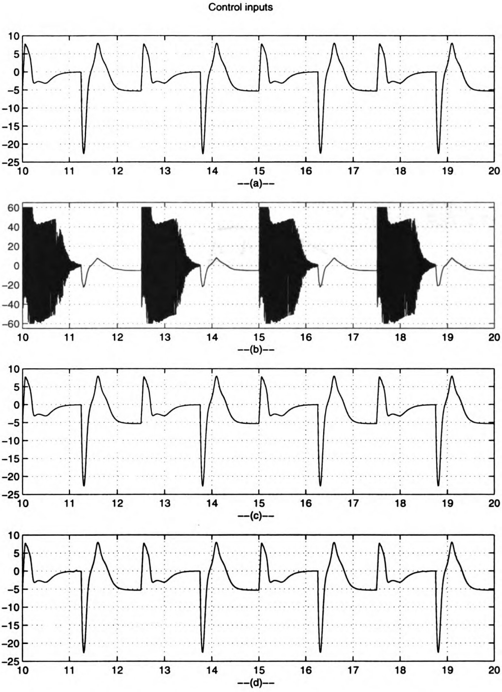

Figure 4.3: (a)State feedback $\mu = 0.5$ (b) Output feedback, $\epsilon = 10^{-3}$, $\mu = 0.5$ (c) Output feedback $\epsilon = 10^{-4}$, $\mu = 0.5$ (d) Output feedback $\epsilon = 10^{-4}$, $\mu = 0.1$

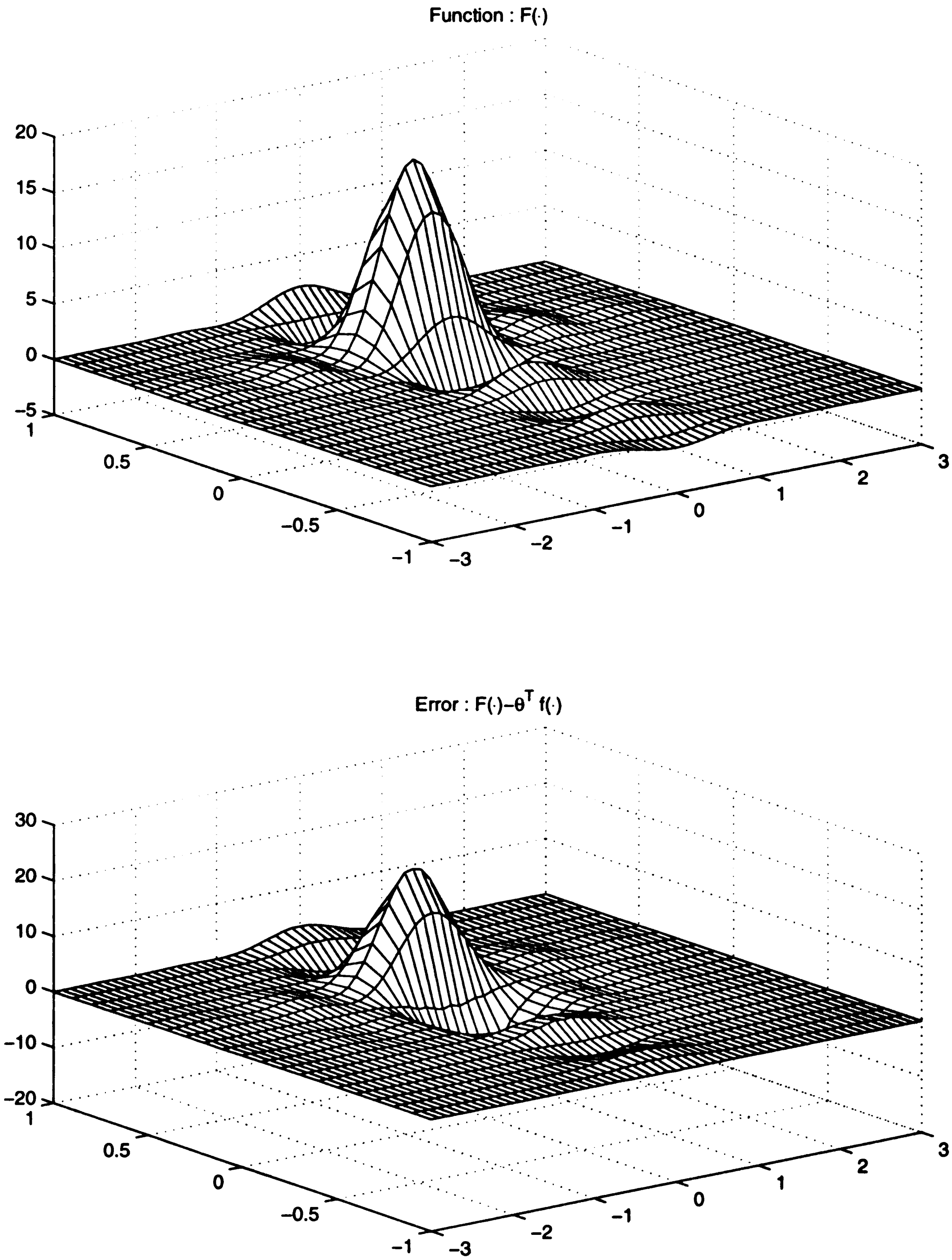

Figure 4.4: (a) $F(y, \dot{y})$ (b) $F(y, \dot{y}) - {\theta_f}^T f(y, \dot{y})$.

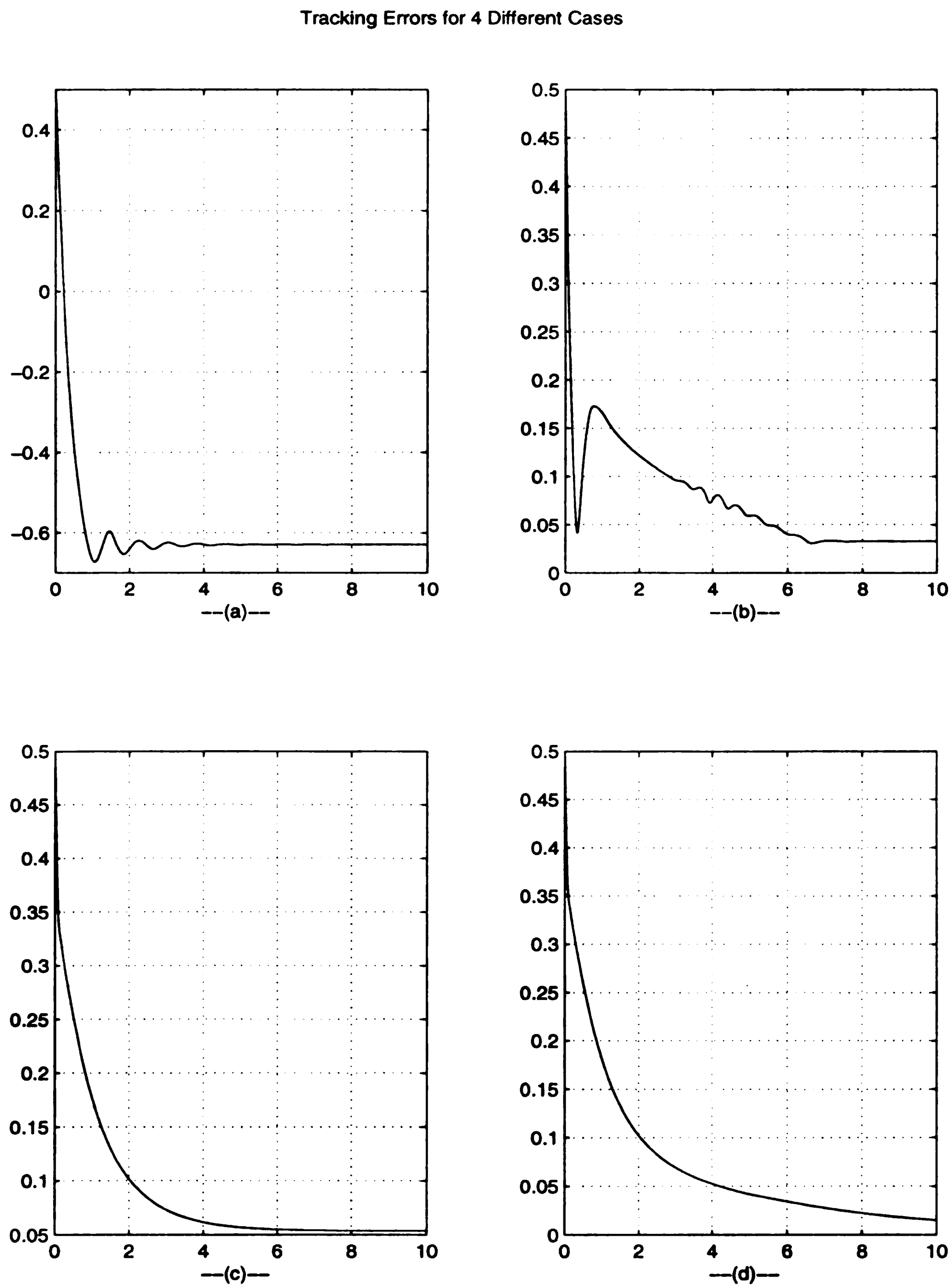

Figure 4.5: (a) No adaptation for weights, no robust control (b) Only adaptation for weights (c) Only robust control (d) Adaptation for weights and robust control.

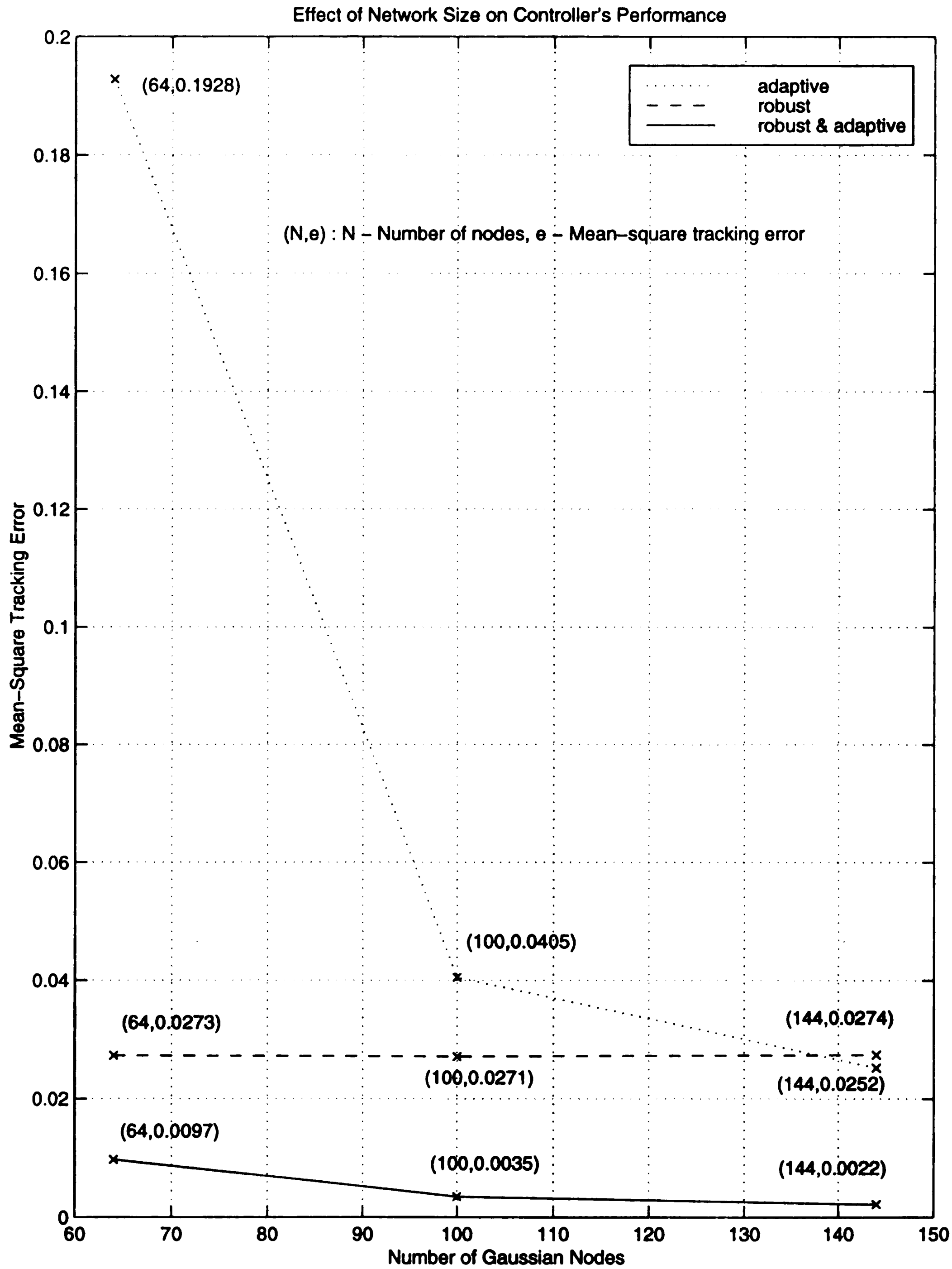

Figure 4.6: Effect of network size on the controller's performance.

Chapter 5

Conclusions

We have studied the application of an adaptive control technique to two different problems, (a) the longitudinal control of a platoon of vehicles and (b) output feedback control of nonlinear systems using RBF neural networks.

For the platoon problem, good performance has been achieved in the presence of parameter uncertainities and unknown time-varying disturbances. The main contribution of the method is the use of high-gain observers to reduce the number of sensor measurements. In particular, we do not require direct measurement of the relative velocity or acceleration between the controlled and leading vehicles or the controlled vehicle's acceleration. A drawback in the present work is the use of the same model for the controller design and in the simulation, i.e, we do not consider imperfections in the plant such as time delays involved in actuators and sensors etc. This drawback, while present in the work of [8],[39], has been addressed in [20]. Future work on this problem must therefore include these imperfections. Another possible area of work is the study of robustness of the proposed method with respect to a large class of commanded maneuvers (such as when a vehicle exits the lane or when two platoons merge to form a single platoon) in the multi-platoon scenario.

The second problem we have studied is adaptive output feedback control 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. RBF networks are used 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. Simulations are done which illustrate the effect of changing various design parameters and of the network size on the controller's performance. Possible future work involves investigating the effect of unmodelled dynamics on the controler's performance and the use of nonlinear-in-the-weights networks. The latter especially holds promise because it is known that nonlinear networks have inherently "better" approximating capabilities in comparison to linear-in-the-weight networks [4]. In particular, for sigmoidal networks, it is shown that with the basis functions "tunable", the integrated square error can be made of the order $O(1/n)$, n being the number of basis functions (or hidden layer weights), whereas, for the case where the basis functions are fixed, the approximation error is typically of the order $O(1/n^{1/d})$, d being the dimension of the input space. Thus nonlinear networks can achieve more compact representations of nonlinear functions compared to linear networks, especially when the dimension of the network input is high.

Appendices

Appendix A

Matlab programs for simulations in Chapter 3

A.1 run.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is the main program, it uses the model simrun.mdl %
% which in turn calls the M-file runn.m                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; tf=125; [t,x]=sim('simrun',tf);
subplot(2,2,1)
plot(t,out(:,1),'-',t,out(:,4),':',t,out(:,7),'-.',t,out(:,10),'--');
grid; axis([0 125 15 45]); title('Vehicle velocity (m/s)')
subplot(2,2,2)
plot(t,out(:,2),'-',t,out(:,5),':',t,out(:,8),'-.',t,out(:,11),'--');
```

```
grid; axis([0 125 -3 2]); title('Vehicle acceleration (m/s^2)')

subplot(2,2,3)

plot(t,out(:,3),'-',t,out(:,6),':',t,out(:,9),'-.',t,out(:,12),'--');

grid; axis([0 125 -0.001 0.002]); title('Separation error (m)')

subplot(2,2,4)

L1=-out(:,3)+0.2*out(:,1); L2=L1+3.9-out(:,6)+0.2*out(:,4);

L3=L2+4.0-out(:,9)+0.2*out(:,7); L4=L3+3.8-out(:,12)+0.2*out(:,10);

plot(t,L1,'-',t,L2,':',t,L3,'-.',t,L4,'--');

grid; axis([0 125 0 50]); title('Position relative to platoon leader (m)')

legend('Car 1', 'Car 2','Car 3', 'Car 4')
```

A.2 runn.m

```
function [sys,xo,str,ts]=runn(t,x,u,flag)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This file is used in calculating by the simrun.mdl file and specifies  %

% all the relevant equations.The assumed definition of delta differs (in %

% sign) from that in Chapter 3. As a result, some signs differ from  the %

% ones mentioned in Ch 3,for eg, y=delta+lambda*v and not delta-lambda*v %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if flag==0,

% x=[delta v f delta^ deltadot^ deltaddot^ v^ vdot^ thetas] %

xo1=[-3 15 167.50 -2 0 0 15 0 -0.0001 -4.5 -0.0005 0.0030] ;

xo2=[-3 15 178.75 -2 0 0 15 0 -0.0003 -5.0 -0.0008 0.0040] ;
```

```
xo3=[-3 15 145.00 -2 0 0 15 0 -0.0002 -4.2 -0.0010 0.0050] ;

xo4=[-3 15 201.25 -2 0 0 15 0 -0.0004 -5.5 -0.0004 0.0045] ;

xo=[xo1 xo2 xo3 xo4]; sys=[12*4,0,3*4,0,0,0,0];

else

% ----- define constants and 'reference' ----- %

lambda=0.2; epsilon=1e-3; Gamma=1e-4; del=1e-3;

eta=4.5; kv=0; mu=0.1; d=100;

K=[3 4]; P=[1.1167 0.1667;0.1667 0.1167]; bb=[0 1]';

a=[-0.00046;-20/3;-0.003;0.0026]; b=[-0.000064;-4;-0.00025;0.0061];

if t<=30,

 wddot=-1/90; wdot=(30-t)/90; w=15+t/3-t^2/180;

elseif t<=40,

 wddot=0; wdot=0; w=20;

elseif t<=70,

 wddot=1/60; wdot=(t-40)/60; w=20+(t-40)^2/120;

elseif t<=100,

 wddot=-1/60; wdot=0.5+(70-t)/60; w=15+5*t/3-(t^2/120)-(190/3);

elseif t<=110,

 wddot=0; wdot=0; w=35;

elseif t<=115,

 wddot=0.4; wdot=0.4*(t-110); w=35+(t-110)^2/5;

elseif t<=120,

 wddot=-1; wdot=117-t; w=-6802.5+117*t-t^2/2;

else
```

```
 wddot=0.6; wdot=-3+0.6*(t-120); w=397.5-3*t+0.3*(t-120)^2;

end

% ---- define states, exogenous inputs and derivatives ---- %

m1=1300; A_pho1=0.30; tau1=0.16; m2=1400; A_pho2=0.35; tau2=0.22;

m3=1200; A_pho3=0.20; tau3=0.18; m4=1350; A_pho4=0.45; tau4=0.24;

th1nom1=-A_pho1/m1; th1nom2=-A_pho2/m2;

th1nom3=-A_pho3/m3; th1nom4=-A_pho4/m4;

th2nom1=-1/tau1; th2nom2=-1/tau2; th2nom3=-1/tau3; th2nom4=-1/tau4;

th3nom1=-A_pho1/(m1*tau1); th3nom2=-A_pho2/(m2*tau2);

th3nom3=-A_pho3/(m3*tau3); th3nom4=-A_pho4/(m4*tau4);

th4nom1=1/(m1*tau1); th4nom2=1/(m2*tau2);

th4nom3=1/(m3*tau3); th4nom4=1/(m4*tau4);

thnom1=[th1nom1;th2nom1;th3nom1;th4nom1];

thnom2=[th1nom2;th2nom2;th3nom2;th4nom2];

thnom3=[th1nom3;th2nom3;th3nom3;th4nom3];

thnom4=[th1nom4;th2nom4;th3nom4;th4nom4];

delta1=x(1); v1=x(2); f1=x(3); delta2=x(13); v2=x(14); f2=x(15);

delta3=x(25); v3=x(26); f3=x(27); delta4=x(37); v4=x(38); f4=x(39);

y1=delta1+lambda*v1; y2=delta2+lambda*v2; %

y3=delta3+lambda*v3; y4=delta4+lambda*v4;

delta_hat1=x(4); v_hat1=x(7); delta_hat2=x(16); v_hat2=x(19);

delta_hat3=x(28); v_hat1=x(31); delta_hat4=x(40); v_hat2=x(43);

th1=[x(9);x(10);x(11);x(12)]; th2=[x(21);x(22);x(23);x(24)];

th3=[x(33);x(34);x(35);x(36)]; th4=[x(45);x(46);x(47);x(48)];
```

```
x4dot1=x(5)+6*(delta1-x(4))/epsilon;

x4dot2=x(17)+6*(delta2-x(16))/epsilon;

x4dot3=x(29)+6*(delta3-x(28))/epsilon;

x4dot4=x(41)+6*(delta4-x(40))/epsilon;

x5dot1=x(6)+11*(delta1-x(4))/(epsilon^2);

x5dot2=x(18)+11*(delta2-x(16))/(epsilon^2);

x5dot3=x(30)+11*(delta3-x(28))/(epsilon^2);

x5dot4=x(42)+11*(delta4-x(40))/(epsilon^2);

x6dot1=6*(delta1-x(4))/(epsilon^3);

x6dot2=6*(delta2-x(16))/(epsilon^3);

x6dot3=6*(delta3-x(28))/(epsilon^3);

x6dot4=6*(delta4-x(40))/(epsilon^3);

x7dot1=x(8)+4*(v1-x(7))/epsilon;

x7dot2=x(20)+4*(v2-x(19))/epsilon;

x7dot3=x(32)+4*(v3-x(31))/epsilon;

x7dot4=x(44)+4*(v4-x(43))/epsilon;

x8dot1=3*(v1-x(7))/(epsilon^2);

x8dot2=3*(v2-x(19))/(epsilon^2);

x8dot3=3*(v3-x(31))/(epsilon^2);

x8dot4=3*(v4-x(43))/(epsilon^2);

if t<=30,

 d11=0; d12=0; d13=0; d14=0;

elseif t<=45

 d11=0.45*(1-exp(-1.2*(t-30))); d12=0.44*(1-exp(-1.2*(t-30)));
```

```
 d13=0.40*(1-exp(-1.2*(t-30))); d14=0.35*(1-exp(-1.2*(t-30)));
elseif t<=55
 d11=0.45*(1-exp(-1.2*(t-30)))-0.45*(1-exp(45-t));
 d12=0.44*(1-exp(-1.2*(t-30)))-0.44*(1-exp(45-t));
 d13=0.40*(1-exp(-1.2*(t-30)))-0.40*(1-exp(45-t));
 d14=0.35*(1-exp(-1.2*(t-30)))-0.35*(1-exp(45-t));
elseif t<=85
 d11=0.45*(1-exp(-30))-0.45*(1-exp(-10));
 d12=0.44*(1-exp(-30))-0.44*(1-exp(-10));
 d13=0.40*(1-exp(-30))-0.40*(1-exp(-10));
 d14=0.35*(1-exp(-30))-0.35*(1-exp(-10));
elseif t<=100
 d11=0.45*(1-exp(-30))-0.45*(1-exp(-10))-0.40*(1-exp(-1.1*(t-85)));
 d12=0.44*(1-exp(-30))-0.44*(1-exp(-10))-0.35*(1-exp(-1.1*(t-85)));
 d13=0.40*(1-exp(-30))-0.40*(1-exp(-10))-0.30*(1-exp(-1.1*(t-85)));
 d14=0.35*(1-exp(-30))-0.35*(1-exp(-10))-0.28*(1-exp(-1.1*(t-85)));
else
 d11=0.45*(1-exp(-30))-0.45*(1-exp(-10))+0.40*(1-exp(-0.8*(t-100)));
 d11=d11-0.40*(1-exp(-1.1*(t-85)));
 d12=0.44*(1-exp(-30))-0.44*(1-exp(-10))+0.35*(1-exp(-0.8*(t-100)));
 d12=d12-0.35*(1-exp(-1.1*(t-85)));
 d13=0.40*(1-exp(-30))-0.40*(1-exp(-10))+0.30*(1-exp(-0.8*(t-100)));
 d13=d13-0.30*(1-exp(-1.1*(t-85)));
 d14=0.35*(1-exp(-30))-0.35*(1-exp(-10))+0.28*(1-exp(-0.8*(t-100)));
```

```
 d14=d14-0.28*(1-exp(-1.1*(t-85)));
end
vdot1=(-A_pho1*v1^2-d+f1)/m1+d11; vdot2=(-A_pho2*v2^2-d+f2)/m2+d12;
vdot3=(-A_pho3*v3^2-d+f3)/m3+d13; vdot4=(-A_pho4*v4^2-d+f4)/m4+d14;
deltadot1=v1-w; deltadot2=v2-v1; deltadot3=v3-v2; deltadot4=v4-v3;
deltadhat1=x(5);  deltadhat2=x(17); deltadhat3=x(29);  deltadhat4=x(41);
deltaddhat1=x(6); deltaddhat2=x(18); deltaddhat3=x(30); deltaddhat4=x(42);
vdothat1=x(8); vdothat2=x(20); vdothat3=x(32); vdothat4=x(44);
ydothat1=deltadhat1+lambda*vdothat1; ydothat2=deltadhat2+lambda*vdothat2;
ydothat3=deltadhat3+lambda*vdothat3; ydothat4=deltadhat4+lambda*vdothat4;
Yhat1=[y1;ydothat1]; Yhat2=[y2;ydothat2];
Yhat3=[y3;ydothat3]; Yhat4=[y4;ydothat4];
s1=2*Yhat1'*P*bb; s2=2*Yhat2'*P*bb; s3=2*Yhat3'*P*bb; s4=2*Yhat4'*P*bb;
f1hat1=2*lambda*v1*vdothat1; f1hat2=2*lambda*v2*vdothat2;
f1hat3=2*lambda*v3*vdothat3; f1hat4=2*lambda*v4*vdothat4;
f2hat1=lambda*vdothat1; f2hat2=lambda*vdothat2;
f2hat3=lambda*vdothat3; f2hat4=lambda*vdothat4;
f31=lambda*v1^2; f32=lambda*v2^2; f33=lambda*v3^2; f34=lambda*v4^2;
f4common=-lambda*d; G=[0;0;0;lambda];
Fhat1=[f1hat1;f2hat1;f31;f4common]; Fhat2=[f1hat2;f2hat2;f32;f4common];
Fhat3=[f1hat3;f2hat3;f33;f4common]; Fhat4=[f1hat4;f2hat4;f34;f4common];
if eta*abs(s1)>=mu, vr1=-(eta*s1)/((1-kv)*abs(s1));
else vr1=-((eta^2)*s1)/((1-kv)*mu); end
if eta*abs(s2)>=mu, vr2=-(eta*s2)/((1-kv)*abs(s2));
```

```
else vr2=-((eta^2)*s2)/((1-kv)*mu); end
if eta*abs(s3)>=mu, vr3=-(eta*s3)/((1-kv)*abs(s3));
else vr3=-((eta^2)*s3)/((1-kv)*mu); end
if eta*abs(s4)>=mu, vr4=-(eta*s4)/((1-kv)*abs(s4));
else vr4=-((eta^2)*s4)/((1-kv)*mu); end
u1=(-deltaddhat1-th1'*Fhat1-K*Yhat1)/(th1'*G);
u2=(-deltaddhat2-th2'*Fhat2-K*Yhat2)/(th2'*G);
u3=(-deltaddhat3-th3'*Fhat3-K*Yhat3)/(th3'*G);
u4=(-deltaddhat4-th4'*Fhat4-K*Yhat4)/(th4'*G);
if abs(u1)>5000, u1=sign(u1)*5000; end
if abs(u2)>5000, u2=sign(u2)*5000; end
if abs(u3)>5000, u3=sign(u3)*5000; end
if abs(u4)>5000, u4=sign(u4)*5000; end
fdot1=(-f1+u1)/tau1; fdot2=(-f2+u2)/tau2;
fdot3=(-f3+u3)/tau3; fdot4=(-f4+u4)/tau4;
Psi1=s1*(Fhat1+G*u1); Psi2=s2*(Fhat2+G*u2);
Psi3=s3*(Fhat3+G*u3); Psi4=s4*(Fhat4+G*u4);
if abs(flag)==1,
 sys11=[deltadot1 vdot1 fdot1]; sys12=[deltadot2 vdot2 fdot2];
 sys13=[deltadot3 vdot3 fdot3]; sys14=[deltadot4 vdot4 fdot4];
 sys21=[x4dot1 x5dot1 x6dot1 x7dot1 x8dot1];
 sys22=[x4dot2 x5dot2 x6dot2 x7dot2 x8dot2];
 sys23=[x4dot3 x5dot3 x6dot3 x7dot3 x8dot3];
 sys24=[x4dot4 x5dot4 x6dot4 x7dot4 x8dot4];
```

```
  for i=1:4,
   sys31(i)=proj(th1(i),Psi1(i),a(i),b(i),Gamma,del);
   sys32(i)=proj(th2(i),Psi2(i),a(i),b(i),Gamma,del);
   sys33(i)=proj(th3(i),Psi3(i),a(i),b(i),Gamma,del);
   sys34(i)=proj(th4(i),Psi4(i),a(i),b(i),Gamma,del);
  end
 sys1=[sys11 sys21 sys31]; sys2=[sys12 sys22 sys32];
 sys3=[sys13 sys23 sys33]; sys4=[sys14 sys24 sys34];
 sys=[sys1 sys2 sys3 sys4];
elseif abs(flag) ==3,
 sys=[v1;vdot1;y1;v2;vdot2;y2;v3;vdot3;y3;v4;vdot4;y4];
end
end
```

A.3 proj.m

```
function Sdot=proj(Theta,Phi,a,b,Gamma,delta)
l1=Theta>=a; l2=Theta<=b; l3=(l1)&(l2);
l4=Theta>b; l5=Phi<=0; l6=(l4)&(l5);
l7=Theta<a; l8=Phi>=0; l9=(l7)&(l8);
if (l3)|(l6)|(l9) Sdot=Gamma*Phi;
elseif (~l2)&(~lc5) Sdot=Gamma*(1+(b-Theta)/delta)*Phi;
elseif (~l1)&(~l8) Sdot=Gamma*(1+(Theta-a)/delta)*Phi;
end
```

Appendix B

Matlab programs for simulations in Chapter 4

B.1 trainf.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program is to obtain initial estimates for the weights of %
% a network with 48 Gaussian nodes.The program for G is similar. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:48, a(i)=-(3/4)+0.5*mod(i-1,4); end
for i=1:48, b(i)=-(11/4)+0.5*floor((i-1)/4); end
W1f=[a;b]; B1f=2*pi.*ones(1,length(W1f));
Po=[]; y=[]; yd=[]; To=[]; k1=16; k2=0; k3=0;
for i=1:48,
 y(i)=-(3/4)+0.5*mod(i-1,4); yd(i)=-(11/4)+0.5*floor((i-1)/4);
```

```
 if y(i)==k2, f1=1;
 else f1=[sin(4*pi*(y(i)-k2))]/[4*pi*(y(i)-k2)]; end
 if yd(i)==k3, f2=1;
 else f2=[sin(pi*(yd(i)-k3))]/[pi*(yd(i)-k3)]; end
 To=[To,k1*f1*(f2^2)];
end
Tp=[100 0.0048]; Po=[y;yd]; [W2f,B2f] = mysolverb(Po,To,W1f',2*pi,Tp);
% mysolverb.m is similar to matlab's solverb.m with minor modifications.
% While in solverb.m the RBF's 'Gaussian wts' are chosen by the program
% depending on the training samples, in mysolverb.m, we use apriori
% fixed basis wts and determine only the linear-weights (ie, mainly
% use only the solvelin part of solverb.)
save weightsfs.mat W1f B1f W2f B2f;
```

B.2 nno.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is the main program,it uses the model simsloto.mdl, %
% which in turn calls the matlab M-file slotsimo.m         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear; clc; load weightsfs; load weightsg;
af=W2f-0.1; bf=W2f+0.1; ag=W2g-0.1; bg=W2g+0.1;
xo=[0 0 0 -0.5 2  0 0 W2f W2g]'; tf=[0 20];
delta=1e-3; Gamma=1e3;
```

```
[t,x]=sim('simsloto',tf);

e=simout(:,1); v=simout(:,2);

subplot(2,1,1); plot(t,simout(:,1)); grid;

xlabel(' Time in seconds') title('Tracking error e')

subplot(2,1,2); hold on; plot(t,simout(:,2)); grid;

xlabel(' Time in seconds'); title('Control input v')
```

B.3 slotsimo.m

```
% This program contains all relevant equations %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [sys,xi,str,ts]=slotsimo(t,x,u,flag,xo,W1f,W2f,B1f,

B2f,af,bf,W1g,W2g,B1g,B2g,ag,bg,Gamma,delta)

if flag==0,

 xi=xo; sys=[7+length(W2f)+length(W2g),0,2,0,0,0,0];

else

 P=[1.1167 0.0833;0.0833 0.1167]; bb=[0 1]'; K=[3 2];

 eta=40; kv=0.7; mu=0.5; epsilon=0.001; vmax=60;

 yr=x(1); yrd=x(2); yrdd=x(3);

 y=x(4); yd=x(5); W2f=x(8:(7+length(W2f)));

 W2g=x(8+length(W2f):7+length(W2f)+length(W2g));

 e1=y-yr; e2=yd-yrd; e1hat=x(6); e2hat=x(7)/epsilon;

 e=[e1 e2]; s=2*e*P*bb; U=[y e2+yrd]';

 [Phif,Fhat]=simurb(U,W1f',B1f',W2f',B2f);
```

```
 [Phig,Ghat]=simurb(U,W1g',B1g',W2g',B2g);

 if eta*abs(s)>=mu, vr=-(eta*s)/((1-kv)*abs(s));

  else vr=-((eta^2)*s)/((1-kv)*mu); end

 if y==0, f1=1; else f1=[sin(4*pi*y)]/[4*pi*y]; end

 if yd==0, f2=1; else f2=[sin(pi*yd)]/[pi*yd]; end

 F=16*f1*(f2^2); G=2+sin((3*pi*(y-0.5)));

 v=-K*e'-Fhat+yrdd+vr/Ghat;

 if abs(v)>vmax v=sign(v)*vmax; end

 ydd=F+v*G; Psif=s*Phif; Psig=s*Phig*v;

 x6dot=(x(7)+1*(e1-e1hat))/epsilon; x7dot=6*(e1-e1hat)/epsilon;

 if abs(flag)==1,

  for i=1:length(W2f),

   W2fdot(i)=proj(W2f(i),Psif(i),af(i),bf(i),Gamma,delta);

  end

  for i=1:length(W2g),

   W2gdot(i)=proj(W2g(i),Psig(i),ag(i),bg(i),Gamma,delta);

  end

  u=0.5*sign(sin(0.8*pi*t))+0.5;

  sys=[x(2);x(3);-1000*x(1)-300*x(2)-30*x(3)+1000*u;yd;ydd;

  x6dot;x7dot;W2fdot';W2gdot'];

 elseif abs(flag) ==3, sys=[e1,v];

 end

end
```

Bibliography

[1] C. C. Chien A. Stotsky and P. Ioannou. Robust platoon-stable controller design for autonomous intelligent vehicles. In *Proc. 1994 Conf. on Decision and Contr.*, pages 2431–2436, December 1994.

[2] B. Aloliwi and H. K. Khalil. Robust adaptive output feedback control of nonlinear systems without persistence of excitation. *Automatica*, 33(11):2025–2032, November 1997.

[3] George A.Rovithakis and Manolis A. Christodoulou. A robust neural adaptive control scheme. In *Proc. 1995 Conf. on Decision and Contr.*, pages 1831–1832, December 1995.

[4] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inform. Theory*, 39(3):930–945, May 1993.

[5] C. I. Byrnes and A. Isidori. Asmptotic stabilization of minimum phase nonlinear systems. *IEEE Trans. Automat. Contr.*, 36(10):1122–1137, October 1991.

[6] R. J. Caudill and W. L. Garrard. Vehicle-follower longitudinal control for automated transit vehicles. *ASME J. Dyn. Sys., Meas and Contr.*, pages 241–248, 1977.

[7] F. C. Chen and H. K. Khalil. Adaptive control of a class of nonlinear discrete-time systems using neural networks. *IEEE Trans. Automat. Contr.*, 40(5):791–801, May 1995.

[8] C. C. Chien and P. Ioannou. Automatic vehicle following. In *Proc. 1992 Amer. Contr. Conf.*, June 1992.

[9] F. Esfandiari and H. K. Khalil. Output feedback stabilization of fully linearizable systems. *Int. J. Contr.*, 56:1007–1037, 1992.

[10] S. Fabri and V. Kadirkamanathan. Dynamic structure neural networks for stable adaptive control of nonlinear systems. *IEEE Trans. Neural Networks*, 7(5):1151–1166, September 1996.

[11] J. A. Farrell. Persistence of excitation conditions in passive learning control. *Automatica*, 33(4):699–703, 1997.

[12] M. Fliess. Generalized controller canonical forms for linear and nonlinear dynamics. *IEEE Trans. Automat. Contr.*, 35(9):994–1001, 1990.

[13] Simon S. Haykin. *Neural Networks : A Comprehensive Foundation.* Macmillan, New York, 1994.

[14] P. V. Kokotovic I. Kanellakapoulos and R. H. Middleton. Indirect adaptive output feedback control of a class of nonlinear systems. In *Proc. 1990 Conf. on Decision and Contr.*, pages 2714–2719, 1990.

[15] P. V. Kokotovic I. Kanellakapoulos and R. H. Middleton. Observer based adaptive control of nonlinear systems under matching condition. In *Proc. 1990 Amer. Contr. Conf.*, pages 549–555, 1990.

[16] P. V. Kokotovic I. Kanellakapoulos and A. S. Morse. Systematic design of adaptive controllers for feedback linearizable systems. *IEEE Trans. Automat. Contr.*, 36:1241–1253, 1991.

[17] P. V. Kokotovic I. Kanellakapoulos and A. S. Morse. Adaptive output feedback control of systems with output nonlnearities. *IEEE Trans. Automat. Contr.*, 37:1266–1282, 1992.

[18] P. A. Ioannou and J. Sun. *Robust Adaptive Control.* Prentice Hall, Upper Saddle River, New Jersey, 1995.

[19] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, New York, 1995.

[20] V. K. Narendran J. K. Hedrick, D. H. McMahon and D. Swaroop. Longitudinal vehicle controller design for ivhs systems. In *Proc. 1991 Amer. Contr. Conf.*, pages 3107–3112, 1991.

[21] 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 Congress of IFAC*, pages 343–348, July 1996.

[22] S. Jagannathan and F. L. Lewis. Discrete-time neural net controller with guaranteed performance. *IEEE Trans. Automat. Contr.*, 41(11):1693–1699, November 1996.

[23] M. Jankovic. Adaptive output feedback control of nonlinear feedback linearizable systems. *Int. J. Adaptive Contr. Signal Proc.*, 10:1–18, 1996.

[24] H. K. Khalil. Adaptive output feedback control of nonlinear systems represented by input-output models. *IEEE Trans. Automat. Contr.*, 41(2):177–188, February 1996.

[25] H. K. Khalil. *Nonlinear Systems.* Prentice Hall, Upper Saddle River, New Jersey, 1996.

[26] K. Khorasani. Indirect adaptive control design for a class of nonlinear systems. In *Proc. 1993 Conf. on Decision and Contr.*, pages 2629–2634, 1993.

[27] M. Krstic and P. V. Kokotovic. Adaptive nonlinear output feedback schemes with marino-tomei controller. *IEEE Trans. Automat. Contr.*, 41:274–280, 1996.

[28] K. W. Lee and H. K. Khalil. Adaptive output feedback control of robot manipulators using high-gain observers. *Int. J. Contr.*, 67:869–886, 1997.

[29] R. Marino and P. Tomei. Dynamic output feedback linearization and global stabilization. *Syst. Contr. Lett.*, 37(3), 1991.

[30] R. Marino and P. Tomei. Global adaptive observers for nonlinear systems via filtered transformations. *IEEE Trans. Automat. Contr.*, 37:1239–1245, 1992.

[31] R. Marino and P. Tomei. Global adaptive output feedback control of nonlinear systems, part i: Linear parametrization. *IEEE Trans. Automat. Contr.*, 38:17–32, 1993.

[32] K. S. Narendra and A. M. Annaswamy. *Stable Adaptive Systems.* Prentice-Hall, Eaglewood Cliffs, New Jersey, 1989.

[33] M. Polycarpou. Stable adaptive neural control scheme for nonlinear systems. *IEEE Trans. Automat. Contr.*, 41(3):447–451, March 1996.

[34] M. Polycarpou and P. Ioannou. Modelling, identification and stable adaptive control of continuous-time nonlinear dynamical systems using neural networks. In *Proc. 1992 Amer. Contr. Conf.*, pages 36–40, June 1992.

[35] R. M. Sanner and J.-J. E. Slotine. Gaussian networks for direct adaptive control. *IEEE Trans. Neural Networks*, 3(6):837–863, November 1992.

[36] S. Sheikholeslam and C. A. Desoer. Longitudinal control of a platoon of vehicles. In *Proc. 1990 Amer. Contr. Conf.*, pages 291–297, 1990.

[37] S. Sheikholeslam and C. A. Desoer. A system level study of the longitudinal control of a platoon of vehicles. *ASME J. Dyn. Sys., Meas and Contr.*, 1991.

[38] S. E. Shladover. Operation of automated guideway transit vehicles in dynamically reconfigured trains and platoons. *UMTA-MA-06-0085-79*, 1979.

[39] J. T. Spooner and K. M. Passino. Stable adaptive control scheme fuzzy systems and neural network. *IEEE Trans. Fuzzy Systems*, 4(3):339–359, August 1996.

[40] A. Teel and L. Praly. Global stabilizability and observability imply semi-global stabilizability by output feedback. *Syst. Contr. Lett.*, 22:313–325, 1994.

[41] A. Tornambe. Output feedback stabilization of a class of non-minimum phase nonlinear systems. *Syst. Contr. Lett.*, 19:193–204, 1992.

[42] E. Tzirkel-Hancock and F. Fallside. Stable control of nonlinear systems using neural networks. *Int. J. Robust Nonlinear Contr.*, 2(1):63–86, April 1992.

[43] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Trans. Automat. Contr.*, 38:195–207, 1993.