NONLINEAR ADAPTIVE INTERNAL MODEL CONTROL USING NEURAL NETWORKS

A Thesis

by

AMIT KRUSHNAVADAN GANDHI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2001

Major Subject: Electrical Engineering

NONLINEAR ADAPTIVE INTERNAL MODEL CONTROL

USING NEURAL NETWORKS

A Thesis

by

AMIT KRUSHNAVADAN GANDHI

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Aniruddha Datta
(Chair of Committee)

Shankar Bhattacharyya
(Member)

Krishna Narayanan
(Member)

Donald Friesen
(Member)

Chanan Singh
(Head of Department)

August 2001

Major Subject: Electrical Engineering

ABSTRACT

Nonlinear Adaptive Internal Model Control Using Neural Networks.

(August 2001)

Amit Krushnavadan Gandhi, B.E., University of Bombay

Chair of Advisory Committee: Dr.Aniruddha Datta

The IMC structure, where the controller implementation includes an explicit model of the plant, has been shown to be very effective for the control of the stable plants typically encountered in process control. A nonlinear internal model control(NIMC) strategy based on neural network models is presented for SISO processes. The nonlinearities of the dynamical system are modelled by neural network architectures. Recurrent neural networks can be used for both the identification and control of nonlinear systems. Identification schemes based on neural network models are developed using two different techniques, namely, the Lyapunov synthesis approach and the gradient method. Both identification schemes are shown to guarantee stability, even in the presence of modelling errors.

The NIMC controller consists of a model inverse controller and a robust filter with single adjustable parameter. Using the theoretical results, we show how an inverse controller can be produced from a neural network model of the plant,without the need to train an additional network to perform the inverse control.

This NIMC approach is currently restricted to processes with stable inverses and with relative degree equal to one. Computer simulations demonstrate the proposed design procedure.

To my Parents, Neha, Arpan, Bhargavi and Avinash

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisor, Dr. Aniruddha Datta, for all the support and guidance he has given me. I would also like to thank him, as well as Dr. Shankar Bhattacharyya, for encouraging me to come to Texas A&M University, supporting me once I arrived and having unwavering faith that I would perform well all through my graduate program. I'm also indebted to other members of my committee, Dr. Krishna Narayanan and Dr. Donald Friesen, for their timely assistance as well as their enthusiastic support.

The road to my graduate degree has been long and winding and it would be truly amiss of me if I did not thank the people who have helped me in achieving my goal. First of all I would like to thank Stevan Dubljevic for having confidence in me and for providing some valuable insights into the problems I faced. Neeraj Upasani and Kamesh Subbarao have always been there with their support. I also sincerely appreciate the patience of Prithi Ramkrishnan, Shirisha Kondury, Karthik Sundaresan and Deepali Limaye for having to repeatedly listen to excruciatingly detailed progress reports on my thesis.

I would also like to thank those closest to me, whose presence helped make the completion of my graduate work possible. They are Bhargavi Sankaran and Avinash Bhangaonkar who kept listening to my babble and helped me to make sense of what I was saying, and kept me going. Most importantly, I would like to express my gratitude to my family for their love, support, encouragement and absolute confidence in me. Finally, in a way that only a mother can, my mother has been my greatest motivator.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The Internal Model Control (IMC) structure continues to be a very popular one in process control applications [1] [2]. This structure, in which the controller includes an explicit model of the plant, is particularly appropriate for the design and implementation of controllers for open-loop stable systems. The fact that many of the plants encountered in process control happen to be open-loop stable possibly accounts for the immense popularity of IMC among practicing engineers.

Although many processes exhibit significant nonlinear behavior, most controller design techniques are based on linear models. Controller design for nonlinear models is considerably more difficult than for linear models. Novel techniques in adaptive control of nonlinear systems were facilitated by advances in geometric nonlinear control theory and, in particular, feedback linearization methods [3]. During the last few years, adaptive nonlinear control has evolved as a vigorous control strategy leading to global stability results for a reasonably large class of nonlinear systems.

The emergence of the neural network paradigm as a powerful tool for learning complex mappings from a set of examples has generated a great deal of excitement in using neural network models for identification and control of dynamical systems with unknown nonlinearities. Due to their approximation capabilities as well as their inherent adaptivity features, artificial neural networks present a potentially appealing alternative to modeling of nonlinear systems. More precisely, it is shown that there exists a set of weights such that for a given input, the outputs of the real system and the proposed neural network model remain arbitrarily close over a finite interval of

---

The journal model is *IEEE Transactions on Automatic Control.*

time. During the past few years, numerous models for the practical identification of nonlinear dynamical systems were proposed and later used for the design of controllers [4]. Extensive simulation studies carried out have shown that these proposed models have been particularly effective for the identification and control of nonlinear systems.

We present a design procedure, based on stability theory, for modeling, identification and adaptive control of continuous-time nonlinear dynamical systems using neural network architectures. The techniques developed here share some fundamental features with the parametric methods of both adaptive nonlinear control as well as adaptive control theory.

The control strategy considered in this paper is that of Internal Model Control (IMC). The applicability of IMC to nonlinear systems control has been demonstrated by Economou *et al.* [5]. The motivating factor for using recurrent neural networks for IMC, is that if a network can model the input-output relationship of a system, it can also surely model the inverse of this relationship, and the production of an inverse model is of great importance when using IMC. By using the Hirschorn inversion theorem, the left inverse of a recurrent neural network is the same network, but with the requirement of a different input [6].

We demonstrate how recurrent neural networks and their inverses can be used in the IMC of a nonlinear system. The system under consideration is that of penicillin fermentation and the recurrent networks used are Radial Basis Function (RBF) networks. The relevant theories about recurrent neural networks and their inverses are reviewed and the stability of the overall IMC system, including the recurrent neural networks, is analyzed.

The Thesis is organized as follows. Chapter II describes the IMC structure for stable plants and mentions its advantages for nonlinear systems. In Chapter III, we talk about modeling of dynamical systems by neural networks and the conditions to be satisfied both by the system and the neural networks. In Chapter IV, we introduce the architecture of Radial Basis Function (RBF) Networks and show that the outputs of a real system and those of the neural network converge. In Chapter V, we deal with Identification of dynamical systems. In particular we discuss the RBF network model considered and the Lyapunov synthesis approach for identifying the states. Then we go on to identify the output of the system using the gradient method. Chapter VI deals with Plant Inversion. We first discuss a few key terms and then move on to talk about the Hirschorn Inverse theorem for constructing the left inverse of the system. Chapter VII supports our design with a simple simulation example. Chapter VIII concludes the thesis by summarizing the main results and outlining the directions for future research.

CHAPTER II

INTERNAL MODEL CONTROL

The IMC structure is well known and has been shown to underlie a number of control design techniques of apparently different origin. IMC has been shown to have a number of desirable properties; a detailed analysis has been given by Morari and Zafiriou [1]. The IMC structure provides a direct method for the design of nonlinear feedback controllers.

A. The IMC Structure

Consider the nonlinear IMC structure shown in Fig. 1. Here the nonlinear operators denoted by P, M and C represent the plant, plant model and the controller respectively. The operator F denotes a filter. The double lines used in the block diagram emphasize that the operators are nonlinear and that the usual block diagram manipulations do not hold.

B. Properties of the IMC Structure for Nonlinear Systems

The attractive characteristics of IMC are the consequences of three properties which we will now state [5].

1. **Property P1 (Stability).** Assume that C and P are input-output stable and that a perfect model of the plant is available, i.e., $M = P$. Then the closed-loop system is input-output stable.

2. **Property P2 (Perfect Control).** Assume that the inverse of the model operator M^I exists, that $C = M^I$, and that the closed-loop system is input-output stable with this controller. Then the control will be perfect, i.e., $y = r$.

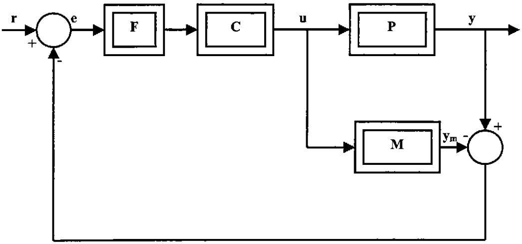

Fig. 1. Nonlinear IMC Structure

3. **Property P3 (Zero Offset).** Assume that the inverse of the steady-state model operator M_∞^I exists, that the controller satisfies $C_\infty = M_\infty^I$, and that the closed-loop system is input-output-stable with this controller. Then offset free control is attained for asymptotically constant inputs.

These properties provide the non-linear analogue of the desirable linear internal model control properties. The importance of nonlinear internal model control is that it provides a direct and transparent method for nonlinear control system design.

For nonlinear systems, general guidelines are not available on how to design a feedback controller for which the closed-loop system is stable, even less for which the overall control structure has some desired performance characteristics. The IMC formalism is aimed at alleviating this problem at least for systems which are input-output stable (or stabilizable by output feedback) and systems which do not exhibit multiple output steady states. Under these assumptions, if a good model of the plant is available, P2 prescribes exactly the structure and parameters of the controller which will result in "perfect control", i.e., exact set point following despite unmeasured disturbances. Moreover, P1 guarantees the stability of the closed-loop nonlinear control system. The underlying idea is that as far as the design is concerned, IMC transforms the problem into a feedforward control problem, which can be solved even for nonlinear systems. But on the other hand, IMC preserves all the important characteristics of feedback control, in particular the suppression of unmeasured plant disturbances as properties P2 and P3 show.

It is intuitively obvious that a feedback controller with infinite gain is necessary to achieve the performance stipulated in P2. Therefore it is evident from this interpretation that the control system suggested in P2 will suffer from stability/sensitivity problems. The closed-loop system can be unstable if the compensator is not correct (the model of the plant is not exact). At this point in the IMC design procedure, we can back off from the "perfect" controller in an orderly fashion and reduce the gain to improve the robustness characteristics. This is accomplished by employing the IMC filter F in series with the controller C.

The primary reason for including this filter is to introduce robustness in the IMC structure in the face of modeling errors, by appropriately reducing the loop gain. It also serves a number of other functions. The input space of the inverse operator M^I is defined to be the range of the operator M; consequently the operator M^I is not defined for inputs outside the range of M. However, there is no assertion that the error signal e will belong to this space. In this case, the filter is used to project e to the appropriate space. Finally, the filter smoothes out noisy and/or rapidly changing signals in order to reduce the transient response of the IMC controller (or, using linear systems terminology, it makes the controller "proper").

We propose a two step procedure for using neural networks directly within the IMC structure. The first step involves training a network to represent the plant response. This network is then used as the plant model operator M in the control structure of Fig. 1.

Following standard IMC practice (guided by property P2 above) we select the controller as the plant inverse. The second step in the procedure is to train a second network to represent the inverse of the plant.

CHAPTER III

MODELING OF DYNAMICAL SYSTEMS BY NEURAL NETWORKS

Artificial Neural Networks are computational paradigms which implement simplified models of their biological counterparts, biological neural networks. Neural Networks are better interpreted as versatile mappings represented by the composition of many basic functions structured in a parallel fashion. They have the following characteristics [7]:

1. Can make use of readily available process data;
2. Are easy to apply; and
3. Can be used across a broad spectrum of process applications.

As a result Artificial Neural Networks can be used for modeling and control of dynamical systems.

A. Stone-Weierstrass Theorem

A generalization of Weierstrass's theorem due to Stone, called the Stone-Weierstrass theorem forms the starting point for all the approximation procedures for dynamical systems.

Theorem 1 ***Stone-Weierstrass*** *[8]: Let $\mathcal{U}$ be a compact metric space. If $\hat{\mathcal{P}}$ is a subalgebra of $C(\mathcal{U}, \mathbb{R})$ which contains the constant functions and separates points of $\mathcal{U}$ then $\hat{\mathcal{P}}$ is dense in $C(\mathcal{U}, \mathbb{R})$.*

In the problems of interest to us we shall assume that the plant P to be identified belongs to the space $\mathcal{P}$ of bounded, continuous, time-invariant and causal operators.

By the Stone-Weierstrass theorem, if $\hat{\mathcal{P}}$ satisfies the conditions of the theorem, a model belonging to $\hat{\mathcal{P}}$ can be chosen which approximates any specified operator $P \in \mathcal{P}$.

The recurrent neural network is an alternative to the feedforward neural network. It was introduced by Hopfield [9]. Recurrent neural networks are feedforward neural networks with feedback connections. The introduction of feedback enables the description of temporal behaviour and hence the capacity to directly account for the dynamics of nonlinear systems. Recurrent neural networks also have the ability to model nonlinear functions, an ability which makes them attractive for use in nonlinear control strategies.

B. Neural Network Requirements

One of the basic requirements in using neural network architectures to represent,identify and control nonlinear dynamical systems is the capability of these architectures to accurately model the behavior of a large class of dynamical systems that are encountered in science and engineering problems. The input-output response of neural networks,whether static or recurrent, is determined by the values of a set of parameters which are referred to as weights. Therefore the representation capabilities of a given network depend on whether there exists a set of weight values such that the neural network configuration approximates the behavior of a given dynamical system. The terms "weights" and "parameters" are used interchangeably.

We consider the problem of constructing a neural network architecture that is capable of approximating the behavior of continuous-time dynamical systems, whose input-state-output representation is described by

$$\begin{aligned} \dot{x} &= f(x,u) \qquad\qquad x(0) = x^0 \\ y &= h(x,u) \end{aligned} \tag{3.1}$$

where $u \in \mathcal{R}$ is the input, $x \in \mathcal{R}^n$ is the state, $y \in \mathcal{R}$ is the output and $t \in \mathcal{R}^+$ is the temporal variable.The input u belongs to a class $\mathcal{U}$ of (piecewise continuous) admissible inputs. By adding and subtracting Ax, where A is a Hurwitz or stability matrix (i.e., has all of its eigenvalues in the open left-half complex plane), (3.1) becomes

$$\dot{x} = Ax + g(x,u), \qquad y = h(x,u) \tag{3.2}$$

where $g(x,u) := f(x,u) - Ax$. Based on (3.2), we construct a recurrent network model by replacing the mappings g and h by feedforward (static) neural network architectures [10], denoted by N_1 and N_2 respectively. Therefore we consider the model

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + \hat{g}(\hat{x}, u, \theta_g) \qquad \hat{x}(0) = \hat{x}^0 \\ \hat{y} &= \hat{h}(\hat{x}, u, \theta_h) \end{aligned} \tag{3.3}$$

where $\hat{g}$ and $\hat{h}$ are the outputs of the static neural networks N_1 and N_2 respectively, while θ_g and θ_h denote the adjustable weights of these networks. $\hat{x}$ and $\hat{y}$ denote the state and output respectively of the recurrent network model.

Corresponding to the Hurwitz matrix A, we let $W(s) := (sI - A)^{-1}$ be an $n \times n$ matrix whose elements are stable transfer functions and s denotes the differential (Laplace) operator. Based on this definition of $W(s)$ as a stable filter, a block diagram representation of the recurrent network model described by (3.3) is depicted in Fig. 2. This interconnection of static neural nets and dynamic components is proposed for modeling the input-output response of the general dynamical system described by (3.1). If we suppose that the real system and the proposed model are initially at the same state (i.e., $\hat{x}^0 = x^0$), then we need to determine whether there exist weights θ_g^*,

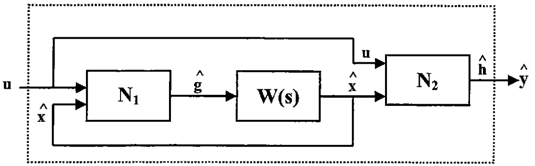

Fig. 2. A Recurrent Neural Network for Modeling a Dynamical System

θ_h^* such that the input-output behavior $(u \mapsto \hat{y})$ of the neural network model (3.3) approximates, in some sense, the input-output behavior $(u \mapsto y)$ of the real system (3.1). This leads to the validity of the proposed model.

C. Conditions on the System and the Neural Network Topologies

We impose the following mild assumptions on the system to be approximated [10]:

(S1) Given a class $\mathcal{U}$ of admissible inputs, then for any $u \in \mathcal{U}$ and any finite initial condition x^0 the state and output trajectories do not escape to infinity in finite time, i.e., for any finite $T > 0$ we have $|x(T)| + |y(T)| < \infty$.

(S2) The vector fields $f\colon \mathcal{R}^{n+m} \mapsto \mathcal{R}^n$ and $h\colon \mathcal{R}^{n+m} \mapsto \mathcal{R}^p$ are continuous with respect to their arguments. Furthermore, f satisfies a local Lipschitz condition so that the solution $x(t)$ to the differential equation (3.1) is unique for any finite initial condition x^0 and $u \in \mathcal{U}$.

The above assumptions are required in order to guarantee that the solution to the system described by (3.1) exists and is unique for any finite initial condition x^0 and any admissible input $u \in \mathcal{U}$.

We will also assume that the static neural network topologies $N_i, i = 1, 2$, that are used to represent the mappings g and h satisfy the following conditions:

(N1) Given a positive constant ϵ and a continuous function $f : \mathcal{C} \mapsto \mathcal{R}^p$, where $\mathcal{C} \subset \mathcal{R}^q$ is a compact set, there exists a weight vector $\theta = \theta^*$ such that the output $\hat{f}(X, \theta)$ of the neural network architecture N_i with n^* nodes (where n^* may depend on ϵ and f) satisfies

$$\max_{X \in \mathcal{C}} |\hat{f}(X, \theta^*) - f(X)| \leq \epsilon$$

(N2) The output $\hat{f}(X, \theta^*)$ of the neural network architecture N_i is continuous with respect to its arguments for all finite (X, θ).

We next describe Radial Basis Function (RBF) networks which satisfy conditions (N1), (N2).

CHAPTER IV

RADIAL BASIS FUNCTION NEURAL NETWORKS

Radial basis function networks have increasingly attracted interest for engineering applications due to their advantages over traditional multilayer perceptrons, namely faster convergence, smaller extrapolation errors, and higher reliability. Radial Basis Function networks were introduced to the neural network literature by Broomhead *et al.* [11] and have since gained significance in the field due to several application and theoretical results [12] [13]. Radial basis function (RBF) networks have a static Gaussian function as the nonlinearity for the hidden layer processing elements. The Gaussian function responds only to a small region of the input space where the Gaussian is centered. The key to a successful implementation of these networks is to find suitable centers for the Gaussian functions. This can be done with supervised learning, but an unsupervised approach usually produces better results. Recently,RBF networks have also been considered in adaptive control of nonlinear dynamical systems [14].

A. Architecture

The input-output response ($x \mapsto y$) of a RBF neural network (shown in Fig.3) with m inputs, n outputs and n^* hidden, or kernel units, is characterized by

$$\begin{aligned} \xi_i &= g(|x - c_i|/\sigma_i) \qquad\qquad i = 1, 2, \ldots n^* \\ y &= A\xi \end{aligned} \tag{4.1}$$

where $x \in \mathcal{R}^m$ is the input, $\xi \in \mathcal{R}^{n^*}$ is the output of the hidden layer, $y \in \mathcal{R}^n$ is the output of the network; $A \in \mathcal{R}^{n \times n^*}$ is the weight matrix, while $c_i \in \mathcal{R}^m$ and $\sigma_i > 0$

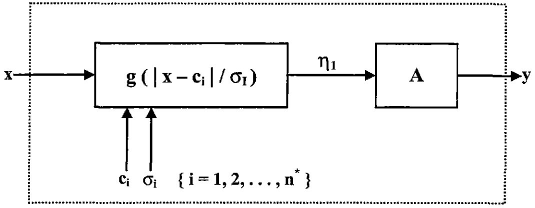

Fig. 3. Block Diagram Representation of a Radial Basis Function Neural Network

are the center and width (or smoothing factor) of the ith kernel unit respectively. The Euclidean or a weighted Euclidean norm $|.|$ is often used. The continuous function $g : [0, \infty) \mapsto \mathcal{R}$ is the activation function which is usually chosen to be the Gaussian function $g(\beta) : = e^{-\beta^2}$. Depending on the application, the centers c_i and/or widths σ_i of the network can either be adjustable (during learning) or they can be fixed. It has recently been shown that under mild assumptions RBF neural networks are capable of universal approximation, i.e. approximation of any continuous function over a compact set to any degree of accuracy [15]. Since Gaussian RBF networks satisfy (N1), (N2) they are good candidates for modeling nonlinearities of dynamical systems.

B. Learning Paradigm

Using Assumptions (S1)-(S2), (N1)-(N2), the following theorem [10] establishes the capabilities of the proposed recurrent network architecture depicted in Fig.2 to approximate the behavior of the real system over a finite interval of time.

Theorem 2 *Suppose* $x(0) = \hat{x}(0) = x^0$ *and* $u \in \mathcal{U} \subset \mathcal{R}^m$ *where* $\mathcal{U}$ *is some compact set. Then given* $\epsilon > 0$ *and a finite* $T > 0$, *there exist weight values* θ_g^*, θ_h^* *such that for all* $u \in \mathcal{U}$ *the outputs of the real system and the recurrent neural network model satisfy*

$$\max_{t \in [0,T]} |y(t) - \hat{y}(t)| \leq \epsilon \tag{4.2}$$

Proof There exists a unique solution $x(t)$ to (3.1) such that $|x(t) - x^0| \leq k$ for all $t \in [0, T]$, where k is some positive constant. Let $\mathcal{K}$ be the compact set defined as

$$\mathcal{K} := \{(x, u) \in \mathcal{R}^{n+m} : |x - x^0| \leq k + \epsilon,\ u \in \mathcal{U}\}$$

By Assumption (N2), $\hat{g}$, $\hat{h}$ are continuous functions and therefore they satisfy a Lipschitz condition in the compact domain $\mathcal{K}$, i.e. there exists constants l_g, l_h such that for all (x_1, u), $(x_2, u) \in \mathcal{K}$

$$|\hat{g}(x_1, u, \theta_g) - \hat{g}(x_2, u, \theta_g)| \leq l_g |x_1 - x_2| \tag{4.3}$$

$$|\hat{h}(x_1, u, \theta_h) - \hat{h}(x_2, u, \theta_h)| \leq l_h |x_1 - x_2| \tag{4.4}$$

If we let $e_x := x - \hat{x}$ then from (3.1) and (3.3) we obtain

$$\dot{e}_x = A e_x + g(x, u) - \hat{g}(\hat{x}, u, \theta_g^*) \qquad e_x(0) = 0 \tag{4.5}$$

Based on Assumption (N1), the weight set θ_g^* in (4.5) is chosen such that

$$\max_{(x,u) \in \mathcal{K}} |g(x, u) - \hat{g}(x, u, \theta_g^*)| \leq \epsilon_g \tag{4.6}$$

where $\epsilon_g > 0$ is a constant to be chosen later.

$\therefore$ its solution,

$$\begin{aligned} e_x(t) &= \int_0^t e^{A(t-\tau)}[g(x(\tau),u(\tau)) - \hat{g}(\hat{x}(\tau),u(\tau),\theta_g^\star)]\,d\tau \\ &= \int_0^t e^{A(t-\tau)}[g(x(\tau),u(\tau)) - \hat{g}(x(\tau),u(\tau),\theta_g^\star)]\,d\tau + \\ &\quad \int_0^t e^{A(t-\tau)}[\hat{g}(x(\tau),u(\tau),\theta_g^\star) - \hat{g}(\hat{x}(\tau),u(\tau),\theta_g^\star)]\,d\tau \end{aligned} \tag{4.7}$$

Since A is a Hurwitz matrix there exist positive constants c, α (that depend on A) such that $\|e^{At}\| \leq ce^{-\alpha t}$ for all $t \geq 0$. Based on the constants $c, \alpha, l_g, l_h, \epsilon$ let ϵ_g in (4.6) be chosen as

$$\epsilon_g = \frac{\epsilon\alpha}{2cl_h}\, e^{-cl_g/\alpha} \; > \; 0 \tag{4.8}$$

First consider the case that $|\hat{x}(t) - x^0| < k + \epsilon$ for all $t \in [0,T]$, which implies that $(\hat{x}, u) \in \mathcal{K}$ in that interval. In this case, starting from (4.7), taking norms on both sides and using (4.3), (4.6), (4.8), the following inequalities hold for $t \in [0,T]$:

$$\begin{aligned} |e_x(t)| &\leq \int_0^t \|e^{A(t-\tau)}\|.\,|g(x(\tau),u(\tau)) - \hat{g}(x(\tau),u(\tau),\theta_g^\star)|\,d\tau + \\ &\quad \int_0^t \|e^{A(t-\tau)}\|.\,|\hat{g}(x(\tau),u(\tau),\theta_g^\star) - \hat{g}(\hat{x}(\tau),u(\tau),\theta_g^\star)|\,d\tau \\ &\leq \int_0^t ce^{-\alpha(t-\tau)}.\,\frac{\epsilon\alpha}{2cl_h}\,e^{\frac{-cl_g}{\alpha}}\,d\tau + \int_0^t ce^{-\alpha(t-\tau)}.\,l_g|e_x(\tau)|\,d\tau \\ &\leq \frac{\epsilon}{2l_h}\,e^{\frac{-cl_g}{\alpha}} + cl_g\int_0^t e^{-\alpha(t-\tau)}\,|e_x(\tau)|\,d\tau \end{aligned}$$

Therefore by using the Bellman-Gronwall Lemma [16] we obtain

$$\begin{aligned} |e_x(t)| &\leq \frac{\epsilon}{2l_h}\,e^{-cl_g/\alpha}.\,e^{cl_g\int_0^t e^{-\alpha(t-\tau)}\,d\tau} \\ |e_x(t)| &\leq \frac{\epsilon}{2l_h} \end{aligned} \tag{4.9}$$

Note that from (4.4) it can be assumed without loss of generality that $l_h \geq 1$ and therefore (4.9) implies that $|\hat{x}(t) - x^0| \leq k + \epsilon/2$. Now suppose (for the sake of contradiction) that $(\hat{x}, u)$ does not belong to $\mathcal{K}$ for all $t \in [0, T]$. Then, by the continuity of $\hat{x}(t)$, there exists a T^*, where $0 < T^* < T$, such that $|\hat{x}(T^*) - x^0| = k + \epsilon$. Therefore, if we carry out the same analysis for $t \in [0, T^*]$ we obtain that in this interval $|\hat{x}(t) - x^0| \leq k + \epsilon/2$, which is clearly a contradiction. Hence (4.9) holds for all $t \in [0, T]$.

Now consider the difference in outputs. By taking norms we obtain

$$\begin{aligned} |y(t) - \hat{y}(t)| &= |h(x, u) - \hat{h}(\hat{x}, u, \theta_h^*)| \\ &\leq |h(x, u) - \hat{h}(x, u, \theta_h^*)| + |\hat{h}(x, u, \theta_h^*) - \hat{h}(\hat{x}, u, \theta_h^*)| \end{aligned}$$

Again by Assumption (N1) we can choose θ_h^* such that

$$\max_{(x,u)\in K} \left| h(x, u) - \hat{h}(x, u, \theta_h^*) \right| \leq \frac{\epsilon}{2}$$

Therefore using (4.4) and (4.9) we obtain

$$|y(t) - \hat{y}(t)| \leq \frac{\epsilon}{2} + l_h \, |e_x(t)| \leq \epsilon \qquad \forall t \in [0, T]$$

which concludes the proof.

Based on the above result, we will assume in the subsequent sections that the nonlinear dynamical system to be identified and/or controlled is represented by a recurrent network configuration with static neural networks replacing the unknown nonlinearities. Thus, the real system is parameterized by neural network models with known underlying structure and unknown parameters or weights. In order to account for modeling inaccuracies arising, for example, from having insufficient number of adjustable weights, we will allow the presence of modeling errors, which appear as additive disturbances in the differential equation representing the system model. We develop parameter update laws for stable identification using various techniques derived from the Lyapunov synthesis approach.

CHAPTER V

IDENTIFICATION

A. State Estimation

In this section we consider the identification of nonlinear systems of the form

$$\begin{aligned} \dot{x} &= f(x) + g(x)\,u \\ y &= h(x) \end{aligned} \tag{5.1}$$

where $u \in \mathcal{R}$ is the input, $y \in \mathcal{R}$ is the output, $x \in \mathcal{R}^n$ is the state, which is assumed to be available for measurement, h is a scalar field and f, g are smooth vector fields defined on an open set of $\mathcal{R}^n$

The problem of identification consists of choosing an appropriate identification model and adjusting the parameters of the model according to some adaptive law such that the response $\hat{x}$ of the model to an input signal u (or a class of input signals) approximates the response x of the real system to the same input. Since a mathematical characterization of a system is often a prerequisite to analysis and controller design, system identification is important not only for understanding and predicting the behavior of the system, but also for obtaining an effective control law.

We will assume that the state $x(t)$ is bounded for all admissible bounded inputs $u(t)$. Note that even though the real system is bounded-input bounded-state (BIBS) stable, there is no apriori guarantee that the output $\hat{x}$ of the identification model or that the adjustable parameters in the model will remain bounded. Stability of the overall scheme depends on the particular identification model that is used as well on the parameter adjustment rules that are chosen. This section is concerned with the development of identification models, based on RBF neural networks, and

CHAPTER V

IDENTIFICATION

A. State Estimation

In this section we consider the identification of nonlinear systems of the form

$$\begin{aligned} \dot{x} &= f(x) + g(x)\,u \\ y &= h(x) \end{aligned} \tag{5.1}$$

where $u \in \mathcal{R}$ is the input, $y \in \mathcal{R}$ is the output, $x \in \mathcal{R}^n$ is the state, which is assumed to be available for measurement, h is a scalar field and f, g are smooth vector fields defined on an open set of $\mathcal{R}^n$

The problem of identification consists of choosing an appropriate identification model as in Fig. 4 and adjusting the parameters of the model according to some adaptive law such that the response $\hat{x}$ of the model to an input signal u (or a class of input signals) approximates the response x of the real system to the same input. Since a mathematical characterization of a system is often a prerequisite to analysis and controller design, system identification is important not only for understanding and predicting the behavior of the system, but also for obtaining an effective control law.

We will assume that the state $x(t)$ is bounded for all admissible bounded inputs $u(t)$. Note that even though the real system is bounded-input bounded-state (BIBS) stable, there is no apriori guarantee that the output $\hat{x}$ of the identification model or that the adjustable parameters in the model will remain bounded. Stability of the overall scheme depends on the particular identification model that is used as well on the parameter adjustment rules that are chosen. This section is concerned

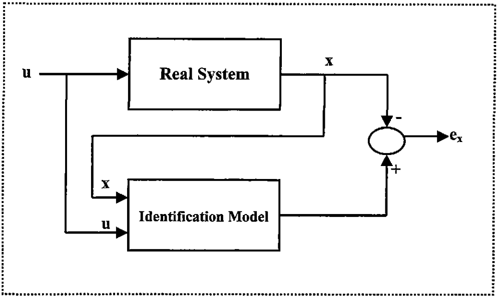

Fig. 4. A General Configuration for Identification of Nonlinear Dynamical System

with the development of identification models, based on RBF neural networks, and the derivation of adaptive laws that guarantee stability of the overall identification structure.

The unknown nonlinearities $f(x)$ and $g(x)$ are parameterized by static neural networks with outputs $\hat{f}(x, \theta_f)$ and $\hat{g}(x, \theta_g)$ respectively, where $\theta_f \in \mathcal{R}^{n_f}, \theta_g \in \mathcal{R}^{n_g}$ are the adjustable weights and n_f, n_g denote the number of weights in the respective neural network approximation of f and g.

$$\therefore \dot{x} = \hat{f}(x, \theta_f^*) + \hat{g}(x, \theta_g^*)\, u + [\, f(x) - \hat{f}(x, \theta_f^*)\,] + [\, g(x) - \hat{g}(x, \theta_g^*)\,]\, u \qquad (5.2)$$

where θ_f^*, θ_g^* denote the optimal weight values (in the L_∞-norm sense) in the approximation of $f(x)$ and $g(x)$ respectively, for x belonging to a compact set $\mathcal{X} \subset \mathcal{R}^n$.

We will consider "optimal" weights θ_f^*, θ_g^* that belong to the convex compact sets $\mathcal{B}(M_f)$, $\mathcal{B}(M_g)$ respectively, where M_f, M_g are design constants and $\mathcal{B}(M) := \{\theta \colon |\theta| \leq M\}$ denotes a ball of radius M. In the adaptive law, the estimates of θ_f^*, θ_g^*, which are the adjustable weights in the approximation networks, are also restricted to $\mathcal{B}(M_f)$, $\mathcal{B}(M_g)$ respectively, through the use of a projection algorithm. The optimal weight vector θ_f^* thus minimizes $|\, f(x) - \hat{f}(x, \theta_f)\,|$ for $x \in \mathcal{X} \subset \mathcal{R}^n$; i.e.,

$$\theta_f^* := arg \min_{\theta_f \in \mathcal{B}(M_f)} \left\{ \sup_{x \in \mathcal{X}} |\, f(x) - \hat{f}(x, \theta_f)\,| \right\} \tag{5.3}$$

Similarly, θ_g^* is defined as

$$\theta_g^* := arg \min_{\theta_g \in \mathcal{B}(M_g)} \left\{ \sup_{x \in \mathcal{X}} |\, g(x) - \hat{g}(x, \theta_g)\,| \right\} \tag{5.4}$$

(5.2) is now expressed as

$$\dot{x} = \hat{f}(x, \theta_f^*) + \hat{g}(x, \theta_g^*)\, u + \nu(t) \tag{5.5}$$

where $\nu(t)$ denotes the modeling error, defined as

$$\nu(t) := [\, f(x(t)) - \hat{f}(x(t), \theta_f^*)\,] + [\, g(x(t)) - \hat{g}(x(t), \theta_g^*)\,]\, u(t)$$

The modeling error $\nu(t)$ is bounded by a constant ν_0 where

$$\nu_0 := \sup_{t \geq 0} \left| \, (f(x(t)) - \hat{f}(x(t), \theta_f^*)) + (g(x(t)) - \hat{g}(x(t), \theta_g^*))\, u(t) \, \right|$$

Since by assumption, $u(t)$ and $x(t)$ are bounded,the constant ν_0 is finite. The value of ν_0 depends on many factors, such as the type of neural network that is used, the number of weights and layers, as well as the "size" of the compact sets $\mathcal{X}$, $\mathcal{B}(M_f)$, $\mathcal{B}(M_g)$. However, a very attractive feature of our synthesis and analysis procedure is that we do not need to know the value of ν_0.

By replacing the unknown nonlinearities with feedforward neural network models, we have essentially rewritten the system (5.1) in the form (5.5), where the parameters

θ_f^*, θ_g^* and the modeling error $\nu(t)$ are unknown, but the underlying structure of $\hat{f}$ and $\hat{g}$ is known. Based on (5.5), we now develop and analyze an identification scheme using Gaussian RBF networks.

1. RBF Network Models

The Network Architectures employed for modeling f and g are RBF networks. Therefore the functions $\hat{f}$ and $\hat{g}$ take the form

$$\hat{f} = W_1^* \eta_1(x), \qquad \hat{g} = W_2^* \eta_2(x) \tag{5.6}$$

where W_1^*, W_2^* are $n \times n_1$ and $n \times n_2$ matrices respectively, representing the optimal weight values, subject to the constraints $\|W_1^*\|_F \leq M_1$, $\|W_2^*\|_F \leq M_2$. The norm $\|.\|_F$ denotes the Frobenius matrix norm [17], defined as $\|A\|_F^2 := \sum_{ij} |a_{ij}|^2 = tr\left\{AA^T\right\}$, where $tr.$ denotes the trace of a matrix. The constants n_1, n_2 are the number of kernel units in each approximation and the vector fields $\eta_1(x) \in \mathcal{R}^{n_1}$, $\eta_2(x) \in \mathcal{R}^{n_2}$, which we refer to as *regressors*, are Gausssian type of functions, defined element-wise as

$$\begin{aligned} \eta_1(x) &= e^{-|x-c_{1i}|^2 / \sigma_{1i}^2} \qquad i = 1, 2, \cdots n_1 \\ \eta_2(x) &= e^{-|x-c_{2j}|^2 / \sigma_{2j}^2} \qquad j = 1, 2, \cdots n_2 \end{aligned}$$

We assume that that c_{1i}, c_{2j}, σ_{1i}, σ_{2j} are chosen apriori and kept fixed during adaptation of W_1, W_2. By substituting (5.6) in (5.5) we obtain

$$\dot{x} = W_1^* \eta_1(x) + W_2^* \eta_2(x)\, u + \nu \tag{5.7}$$

Based on the RBF network model described by (5.7), we next develop parameter update laws for stable identification using various techniques derived from the Lyapunov synthesis approach.

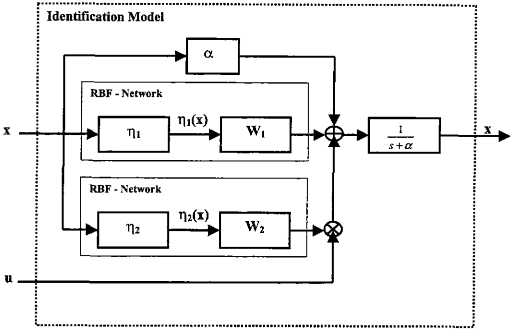

Fig. 5. A Block Diagram Representation of the Identification Model Developed Using RBF Networks

2. Lyapunov Synthesis Approach

The RBF network model (5.7) is rewritten in the form

$$\dot{x} = -\alpha x + \alpha x + W_1^* \eta_1(x) + W_2^* \eta_2(x)\, u + \nu \tag{5.8}$$

where $\alpha > 0$ is a scalar (design) constant. Based on (5.8) we consider the identification model

$$\dot{\hat{x}} = -\alpha \hat{x} + \alpha x + W_1 \eta_1(x) + W_2 \eta_2(x)\, u \tag{5.9}$$

where W_1, W_2 are the estimates of W_1^*, W_2^* respectively, while $\hat{x}$ is the output of the identification model. The RBF filtering model is depicted in Fig. 5. As can be seen from the figure, this identification model consists of two RBF network architectures in parallel and n first order stable filters $h(s) = 1/(s+\alpha)$.

If we define $e_x := \hat{x} - x$, the state error, and $\phi_1 := W_1 - W_1^*, \phi_2 := W_2 - W_2^*$, the weight estimation errors, then from (5.8) and (5.9) we obtain the error equation

$$\dot{e_x} = -\alpha e_x + \phi_1\, \eta_1(x) + \phi_2\, \eta_2(x)\, u - \nu \tag{5.10}$$

The Lyapunov synthesis method [18] consists of choosing an appropriate Lyapunov function candidate V and selecting weight adaptive laws so that the time derivative $\dot{V}$ satisfies $\dot{V} \le 0$. An adaptive law for generating the parameter estimates $W_1(t)$, $W_2(t)$ is developed by considering the Lyapunov function candidate

$$\begin{aligned} V(\,e_x, \phi_1, \phi_2) &= \frac{1}{2}\,|e_x|^2 + \frac{1}{2\gamma_1}\,\|\phi_1\|_F^2 + \frac{1}{2\gamma_2}\,\|\phi_2\|_F^2 \\ &= \frac{1}{2}\,e_x^T e_x + \frac{1}{2\gamma_1}\,tr\{\,\phi_1\phi_1^T\,\} + \frac{1}{2\gamma_2}\,tr\{\,\phi_2\phi_2^T\,\} \end{aligned} \tag{5.11}$$

where γ_1, γ_2 are positive constants referred to as *learning rates* or *adaptive gains*.

Using (5.10), the time derivative of V in (5.11) is expressed as

$$\begin{aligned} \dot{V} &= \dot{e_x}^T e_x + \frac{1}{\gamma_1}\,tr\{\,\dot{\phi}_1\phi_1^T\,\} + \frac{1}{\gamma_2}\,tr\{\,\dot{\phi}_2\phi_2^T\,\} \\ &= -\alpha e_x^T e_x + \eta_1^T\phi_1^T e_x + \eta_2^T\phi_2^T e_x u - \nu^T e_x + \frac{1}{\gamma_1}\,tr\{\,\dot{\phi}_1\phi_1^T\,\} + \frac{1}{\gamma_2}\,tr\{\,\dot{\phi}_2\phi_2^T\,\} \end{aligned}$$

Using properties of the trace, such as

$$\eta_1^T\phi_1^T e_x = tr\{\,\eta_1^T\phi_1^T e_x\,\} = tr\{\,e_x\,\eta_1^T\phi_1^T\,\}$$

we obtain

$$\begin{aligned} \dot{V} &= -\alpha|e_x|^2 + tr\left\{e_x\eta_1^T\phi_1^T + \frac{1}{\gamma_1}\dot{\phi}_1\phi_1^T\right\} + tr\left\{e_x u\eta_2^T\phi_2^T + \frac{1}{\gamma_2}\dot{\phi}_2\phi_2^T\right\} - \nu^T e_x \\ &= -\alpha|e_x|^2 + \frac{1}{\gamma_1}tr\left\{(\gamma_1 e_x\eta_1^T + \dot{\phi}_1)\phi_1^T\right\} + \frac{1}{\gamma_2}tr\left\{(\gamma_2 e_x u\eta_2^T + \dot{\phi}_2)\phi_2^T\right\} - \nu^T e_x \end{aligned} \tag{5.12}$$

Since W_1^*, W_2^* are constant, we have that $\dot{W}_1 = \dot{\phi}_1$ and $\dot{W}_2 = \dot{\phi}_2$. Therefore it is clear from (5.12) that if the parameter estimates W_1, W_2 are generated according to the adaptive laws

$$\dot{W}_1 = -\gamma_1 e_x\eta_1^T, \qquad \dot{W}_2 = -\gamma_2 e_x u\eta_2^T \tag{5.13}$$

then (5.12) becomes

$$\dot{V} = -\alpha|e_x|^2 - \nu^T e_x \leq -\alpha|e_x|^2 + \nu_0|e_x| \tag{5.14}$$

If there is no modeling error (i.e.,ν_0=0), then from (5.15) we have that $\dot{V}$ is negative semidefinite; hence stability of the overall identification scheme is guaranteed. However in the presence of modeling error, if $|e_x| < \nu_0/\alpha$ then it is possible that $\dot{V} > 0$, which implies that the weights $W_1(t)$, $W_2(t)$ may drift to infinity with time. Therefore we confine W_1 and W_2 so that $\|W_1\|_F \leq M_1$ and $\|W_2\|_F \leq M_2$ through the use of a Projection Algorithm [19] [20] as:

$$\dot{W}_1 = \begin{cases} -\gamma_1 e_x\eta_1^T & if\ \|W_1\|_F < M_1 \text{ or } (\ \|W_1\|_F = M_1 \text{ and } e_x^T W_1\eta_1 \geq 0\ ) \\ \mathcal{P}\{-\gamma_1 e_x\eta_1^T\} & if\ \|W_1\|_F = M_1 \text{ and } e_x^T W_1\eta_1 < 0. \end{cases} \tag{5.15}$$

$$\dot{W}_2 = \begin{cases} -\gamma_2 e_x \eta_2^T & if\ \|W_2\|_F < M_2 \text{ or } (\ \|W_2\|_F = M_2 \text{ and } e_x^T W_2 \eta_2 u \geq 0) \\ \mathcal{P}\{-\gamma_2 e_x u \eta_2^T\} & if\ \|W_2\|_F = M_2 \text{ and } e_x^T W_2 \eta_2 u < 0. \end{cases} \tag{5.16}$$

where $\mathcal{P}\{\cdot\}$ denotes the projection onto the supporting hyperplane, defined as

$$\mathcal{P}\{-\gamma_1 e_x \eta_1^T\} := -\gamma_1 e_x \eta_1^T + \gamma_1 \frac{e_x^T W_1 \eta_1}{\|W_1\|_F^2} W_1 \tag{5.17}$$

$$\mathcal{P}\{-\gamma_2 e_x u \eta_2^T\} := -\gamma_2 e_x u \eta_2^T + \gamma_2 \frac{e_x^T W_2 \eta_2 u}{\|W_2\|_F^2} W_2 \tag{5.18}$$

Therefore, if the initial weights are chosen such that $\|W_1(0)\|_F \leq M_1$, $\|W_2(0)\|_F \leq M_2$ then we have $\|W_1(t)\|_F \leq M_1$, $\|W_2(t)\|_F \leq M_2$ for all $t \geq 0$.

With the adaptive laws (5.15), (5.16), (5.12) becomes:

$$\begin{aligned} \dot{V} &= -\alpha|e_x|^2 - \nu^T e_x + I_1^* \, tr\left\{\frac{e_x^T W_1 \eta_1}{\|W_1\|_F^2} W_1 \phi_1^T\right\} + I_2^* \, tr\left\{\frac{e_x^T W_2 \eta_2 u}{\|W_2\|_F^2} W_2 \phi_2^T\right\} \\ &\leq -\alpha|e_x|^2 - \nu^T e_x + I_1^* \frac{e_x^T W_1 \eta_1}{\|W_1\|_F^2} tr\{W_1 \phi_1^T\} + I_2^* \frac{e_x^T W_2 \eta_2 u}{\|W_2\|_F^2} tr\{W_2 \phi_2^T\} \end{aligned} \tag{5.19}$$

where I_1^*, I_2^* are indicator functions defined as $I_1^* = 1$ if $\|W_1\|_F = M_1$ and $e_x^T W_1 \eta_1 < 0$ are satisfied and $I_1^* = 0$ otherwise (and correspondingly for I_2^*). As the following Lemma shows, additional terms introduced by the projection can only make $\dot{V}$ more negative, which implies that the projection modification guarantees boundedness of the weights without affecting the rest of the stability properties established in the absence of projection.

Lemma 1 *Based on the adaptive laws (5.15), (5.16) the following inequalities hold:*

(i) $I_1^* \, (e_x^T W_1 \eta_1 / \|W_1\|_F^2) \; tr\{W_1 \phi_1^T\} \leq 0$.

(ii) $I_2^* \, (e_x^T W_2 \eta_2 u / \|W_2\|_F^2) \; tr\{W_2 \phi_2^T\} \leq 0$

Proof We prove part (i), since the proof of part (ii) follows from the same reasoning. Suppose $\|W_1\|_F = M_1$ and $e_x^T W_1 \eta_1 < 0$. If that is not the case then $I_1^* = 0$ and the inequality holds trivially. The term $tr\left\{ W_1 \phi_1^T \right\}$ can be expressed as

$$\begin{aligned} tr\left\{ W_1 \phi_1^T \right\} &= tr\left\{ (\phi_1 + W_1^*)\phi_1^T \right\} \\ &= tr\left\{ \frac{1}{2}\phi_1\phi_1^T + \left(\frac{1}{2}\phi_1\phi_1^T + W_1^*\phi_1^T \right) \right\} \\ &= \frac{1}{2}\|\phi_1\|_F^2 + \frac{1}{2}\|W_1\|_F^2 + \frac{1}{2}\|W_1^*\|_F^2 \end{aligned}$$

Therefore

$$I_1^* \frac{e_x^T W_1 \eta_1}{\|W_1\|_F^2} \, tr\{W_1 \phi_1^T\} = \frac{1}{2M_1^2} \left(e_x^T W_1 \eta_1\right) \left(\|\phi_1\|_F^2 + M_1^2 - \|W_1^*\|_F^2 \right)$$

Since the optimal weights W_1^* satisfy $\|W_1^*\|_F \leq M_1$, $(\|\phi_1\|_F^2 + M_1^2 - \|W_1^*\|_F^2)$ is positive and therefore

$$I_1^* \frac{e_x^T W_1 \eta_1}{\|W_1\|_F^2} \, tr\{W_1 \phi_1^T\} \leq 0$$

which concludes the proof.

Now, using Lemma 1, (5.19) becomes

$$\dot{V} \leq -\alpha |e_x|^2 - e_x^T \nu \leq -\alpha |e_x|^2 + \nu_0 |e_x| \tag{5.20}$$

Based on (5.20), we next summarize the properties of the weight adaptive laws (5.15), (5.16).

Theorem 3 *The weight adaptive laws given by (5.15), (5.16) guarantee the following properties:*

(a) For $\nu_0 = 0$ (no modeling error), we have

- $e_x, \hat{x}, \phi_1, \phi_2 \in L_\infty$, $\quad e_x \in L_2$.

- $\lim_{t\to\infty} e_x(t) = 0, \;\; \lim_{t\to\infty} \dot{\phi_1}(t) = 0 \;\; \lim_{t\to\infty} \dot{\phi_2}(t) = 0.$

(b) For $\sup_{t\geq 0} |v(t)| \leq v_0$ *we have*

- $e_x, \hat{x}, \phi_1, \phi_2 \in L_\infty$.
- *there exist constants* k_1*,* k_2 *such that*

$$\int_0^t |e_x(\tau)|^2 \, d\tau \leq k_1 + k_2 \int_0^t |\nu(\tau)|^2 \, d\tau.$$

Proof:

(a) With $\nu_0 = 0$, (5.20) becomes

$$\dot{V} \leq -\alpha |e_x|^2 \leq 0 \tag{5.21}$$

Hence $V \in L_\infty$, which from (5.11) implies e_x, ϕ_1, $\phi_2 \in L_\infty$. Furthermore, $\hat{x} = e_x + x$ is also bounded. Since V is a non-increasing function of time and bounded from below the $\lim_{t\to\infty} V(t) = V_\infty$ exists. Therefore by integrating (5.20) from 0 to ∞ we have

$$\int_0^\infty |e_x(\tau)|^2 \, d\tau \leq \frac{1}{\alpha} \left[V(0) - V_\infty \right] < \infty$$

which implies that $e_x \in L_2$. By the definition of the Gaussian radial basis function, the regressor vectors $\eta_1(x)$ and $\eta_2(x)$ are bounded for all x and by assumption u is also bounded. Hence from (5.10) we have that $\dot{e_x} \in L_\infty$. Since $e_x \in L_2 \cap L_\infty$ and $\dot{e_x} \in L_\infty$, using Barbalat's Lemma [21] we conclude that $\lim_{t\to\infty} e_x(t) = 0$. Now, using the boundedness of $\eta_1(t)$ and the convergence of $e_x(t)$ to zero, we have that $\dot{\phi}_1 = \dot{W}_1$ also converges to zero. Similarly, $\dot{\phi}_2 \to 0$ as $t \to \infty$.

(b) With the projection algorithm, it is guaranteed that $\|W_1\|_F \leq M_1$, $\|W_2\|_F \leq M_2$. Therefore the weight estimation errors are also bounded, i.e. $\phi_1, \phi_2 \in L_\infty$. From (5.20) it is clear that if $|e_x| > \nu_0/\alpha$ then $\dot{V} < 0$ which implies that $e_x \in L_\infty$ and consequently $\hat{x} \in L_\infty$. In order to prove the second part, we proceed to complete the square in (5.20):

$$\begin{aligned} \dot{V} &\leq -\frac{\alpha}{2}|e_x|^2 - \frac{\alpha}{2}\left[|e_x|^2 + \frac{2}{\alpha}e_x^T\nu\right] \\ &\leq -\frac{\alpha}{2}|e_x|^2 + \frac{1}{2\alpha}|\nu|^2 \end{aligned}$$

Therefore, by integrating both sides and using the fact that $V \in L_\infty$ we obtain

$$\begin{aligned} \int_0^t |e_x(\tau)|^2 d\tau &\leq \frac{2}{\alpha}[V(0) - V(t)] + \frac{1}{\alpha^2}\int_0^t |\nu(\tau)|^2 d\tau \\ &\leq k_1 + k_2 \int_0^t |\nu(t)|^2 d\tau \end{aligned}$$

where $k_1 := 2/\alpha \left(V(0) - \sup_{t\geq 0} V(t)\right)$ and $k_2 := 1/\alpha^2$.

B. Output Identification

The output of our system is

$$y = h(x) \tag{5.22}$$

The unknown nonlinearity $h(x)$ is parameterized by a static neural network with output $\hat{h}(x, \theta_h)$ where $\theta_h \in \mathcal{R}_h^n$ are the adjustable weights and n_h denote the number of weights in the approximation of h. By adding and subtracting the term $\hat{h}$ (5.22) can be rewritten as

$$y = \hat{h}(x, \theta_h^*) + [\,h(x) - \hat{h}(x, \theta_h^*)\,] \tag{5.23}$$

where θ_h^* denotes the optimal weight values (in the L_∞-norm sense) in the approximation of $h(x)$, for x belonging to a compact set $\mathcal{X} \subset \mathcal{R}^n$.

We will consider "optimal" weights θ_h^* that belong to the convex compact set $\mathcal{B}(M_h)$, where M_h is a design constant and $\mathcal{B}(M) := \{\theta\colon |\theta| \leq M\}$ denotes a ball of radius M. In the adaptive law, the estimates of θ_h^*, which are the adjustable weights in the approximation networks, are also restricted to $\mathcal{B}(M_h)$, through the use of a projection algorithm. The optimal weight vector θ_h^* thus minimizes $|\, h(x) - \hat{h}(x, \theta_h) \,|$ for $x \in \mathcal{X} \subset \mathcal{R}^n$; i.e.,

$$\theta_h^* := arg \min_{\theta_h \in \mathcal{B}(M_h)} \left\{ \sup_{x \in \mathcal{X}} |\, h(x) - \hat{h}(x, \theta_h) \,| \right\} \tag{5.24}$$

(5.23) is now written in compact form as

$$y = \hat{h}(x, \theta_h^*) + d(t) \tag{5.25}$$

where $d(t)$ denotes the modeling error, defined as

$$d(t) := \left[h(x(t)) - \hat{h}(x(t), \theta_h^*) \right]$$

The modeling error $d(t)$ is bounded by a constant d_0 where

$$d_0 := \sup_{t \geq 0} \left| \left(h(x(t)) - \hat{h}(x(t), \theta_h^*) \right) \right|$$

Since by assumption $x(t)$, $y(t)$ is bounded, the constant d_0 is finite. Based on (5.25), we again develop and analyze an identification scheme using Gaussian RBF networks.

1. RBF Network Models

The Network Architecture employed for modeling h is a RBF network. Therefore the function $\hat{h}$ takes the form

$$\hat{h} = W_3^{*T} \eta_3(x) \tag{5.26}$$

where W_3^* is a $1 \times n$ matrix representing the optimal weight values, subject to the constraint $\|W_3^*\|_F \leq M_3$. By substituting (5.26) in (5.25) we get

$$y = W_3^{*T} \eta_3(x) + d \tag{5.27}$$

Based on the RBF network model described by (5.27), we next formulate parameter update laws for stable identification using the Gradient Method. Based on (5.27) we consider the identification model

$$\hat{y} = W_3^T \eta_3(\hat{x}) \tag{5.28}$$

where W_3 is the estimate of W_3^*, while $\hat{y}$ is the output of the identification model. We now define $e_y := y - \hat{y}$, the output error, and $\phi_3 := W_3 - W_3^*$, the weight estimation error. Then from (5.27) and (5.28) we obtain the error equation

$$\begin{aligned} e_y &= y - \hat{y} \\ &= W_3^{*T}\eta_3(x) + d - W_3^T\eta_3(\hat{x}) \\ &= W_3^{*T}\eta_3(\hat{x}) - W_3^T\eta_3(\hat{x}) - W_3^{*T}\eta_3(\hat{x}) + W_3^{*T}\eta_3(x) + d \\ &= -\phi_3^T\eta_3(\hat{x}) + W_3^{*T}[\,\eta_3(x) - \eta_3(\hat{x})\,] + d \\ &= -\phi_3^T\eta_3(\hat{x}) + \bar{d} \end{aligned} \tag{5.29}$$

where $\bar{d} = W_3^{*T}[\,\eta_3(x) - \eta_3(\hat{x})\,] + d$

Approximating $[\,\eta_3(x) - \eta_3(\hat{x})\,]$ to the first order:

$$[\,\eta_3(x) - \eta_3(\hat{x})\,] = \frac{\partial \eta_3}{\partial x}(x - \hat{x})$$

$$\therefore |\eta_3(x) - \eta_3(\hat{x})| = |\frac{\partial \eta_3}{\partial x}|\,|e_x|$$

Let $\bar{d}_0$ be an upper bound on $\bar{d}$,

$$\|\bar{d}_0\| = \|W_3^*\|\,\|\frac{\partial \eta_3}{\partial x}\|\,\|e_x\| + \|d\|$$

Since $\|W_3^*\|_F \leq M_3$, $e_x \in L_\infty$, $\|\frac{\partial \eta_3}{\partial x}\|$ is finite (η is a Gaussian Function) and d is bounded by d_0, therefore $\bar{d}_0$ is also finite.

2. Parameter Projection

An effective method for eliminating parameter drift and keeping the parameter estimates within some apriori bounds is to use the gradient projection method to constrain the parameter estimates to lie inside a *bounded* convex set in the parameter space [22]. Consider the adaptive law:

$$\dot{W}_3 = \gamma_3 e_y \eta_3(\hat{x}) \tag{5.30}$$

Since $\phi_3 := W_3 - W_3^*$,

$\therefore \dot{\phi}_3 = \dot{W}_3$

i.e. $\dot{\phi}_3 = \gamma_3 e_y \eta_3(\hat{x})$

Applying the gradient projection method, we obtain

$$\dot{\phi}_3 = \dot{W}_3 = \begin{cases} \gamma_3 e_y \eta_3(\hat{x}) & if\{\|W_3\|_F < M_3\} \text{ or } \{\|W_3\|_F = M_3 \text{ and } e_y W_3 \eta_3 \leq 0\} \\ 0 & if\{\|W_3\|_F = M_3 \text{ and } e_y W_3 \eta_3 > 0\} \end{cases} \tag{5.31}$$

We choose our Lyapunov function candidate as

$$V = \frac{\phi_3^T \phi_3}{2\gamma_3}$$

whose time derivative $\dot{V}$ along (5.31) is given by

$$\dot{V} = \begin{cases} -e_y^2 + e_y \bar{d} & if\{\|W_3\|_F < M_3\} \text{ or } \{\|W_3\|_F = M_3 \text{ and } e_y W_3 \eta_3 \leq 0\} \\ 0 & if\{\|W_3\|_F = M_3 \text{ and } e_y W_3 \eta_3 > 0\} \end{cases} \tag{5.32}$$

$$\therefore \dot{V} = -e_y^2 + e_y \bar{d} \leq -e_y^2 + |e_y| \bar{d}_0$$

Using completion of squares,

$$\begin{aligned} -e_y^2 + |e_y|\bar{d}_0 &\leq -\frac{e_y^2}{2} - \frac{1}{2}(e_y - \bar{d}_0)(e_y - \bar{d}_0)^T + \frac{\bar{d}_0^{\,2}}{2} \\ &\leq -\frac{e_y^2}{2} + \frac{\bar{d}_0^{\,2}}{2} \end{aligned}$$

$$\therefore \dot{V} \leq -\frac{e_y^2}{2} + \frac{\bar{d}_0^{\,2}}{2}, \quad \forall t \geq 0 \tag{5.33}$$

Thus we have $\phi_3, W_3 \in L_\infty$ and $e_y, \dot{W}_3 \in L_\infty$.

A bound for e_y in m.s.s. may be obtained by integrating both sides of (5.33) to get

$$\int_t^{t+T} e_y^2 d\tau \leq \bar{d}_0^{\,2} T + 2\{V(t) - V(t+T)\}$$

$\forall t \geq 0$ and any $T \geq 0$.

Because $V \in L_\infty$, it follows that $e_y \in \mathcal{S}(\bar{d}_0^{\,2})$.

We have thus identified the output y of the plant.

CHAPTER VI

PLANT INVERSION

We now perform a thorough study of both the theoretical and the practical implementation aspects of operator inversion. A Control System is invertible when the corresponding input-output map is injective. Thus, given an output function one can, in theory, recover the control which was applied [6].

A. Nonlinear Invertibility

We now list the conditions for the invertibility of nonlinear systems followed by a brief discussion of some of the terms, before we proceed to derive the nonlinear inverse operator.

1. Invertibility Conditions

The nonlinear system (5.1) is said to be invertible at $x_0 \in M$ if whenever $u1$ and $u2$ are distinct admissible controls, $y(\cdot\,, u1, x_0) \neq y(\cdot\,, u2, x_0)$ [23]. The condition for the invertibility of a nonlinear system, is that the relative order of the system should be finite, i.e. $\alpha < \infty$. Clearly, invertibility at x_0 is equivalent to the input-output map described by (5.1) being injective. Thus, given the output $y(\cdot)$ for a system which is invertible at x_0, one can, in theory, determine the control which was applied. If a system is invertible at x_0 it is natural to look for a second system which acts as a left-inverse for the original system. The left-inverse system is a nonlinear system which, when driven by appropriate derivatives of $y(\cdot\,, u, x_0)$, produces $u(\cdot)$ as its output. The important consequence of this definition is that, given a nonlinear map from u to y, the inverse map can be obtained by differentiating this mapping until the αth derivative of y is linear with respect to u. Thus the left-inverse system provides a practical method for determining $u(\cdot)$, and has many applications.

2. Zero Dynamics

The Dynamics describing the "internal" behavior of the system when input and initial conditions have been chosen in such a way as to constrain the output to remain identically zero are called the zero dynamics of the system.

The zero dynamics of a nonlinear system are the dynamics of a minimal-order realization of its inverse. There is no direct method for quantifying the behavior of the inverse dynamics. One approach to determine the stability of the inverse system is to analyze the unforced inverse dynamics, called the *zero dynamics* [7]. For example, in the case where the system output is constrained to a constant value (setpoint) - which can be assumed to be zero without loss of generality - the stability of the closed-loop system in which the inverse is employed as the controller is completely determined by the stability of these internal dynamics.

The concept of zero dynamics allows a classification of nonlinear systems into minimum phase and nonminimum phase [24]. A nonlinear system is called minimum phase if its zero dynamics is asymptotically stable around the origin. Otherwise, it is called nonminimum-phase.

3. Relative Order

The relative order of the nonlinear system (5.1) is the smallest integer α for which $L_g L_f^{\alpha-1} h(x) \neq 0$ and $L_g L_f^{\alpha-2} h(x) = 0 \forall x$ in some neighborhood of the defined operating point x_0.

A useful interpretation of the relative order for nonlinear systems can be obtained by calculating derivatives of the output:

$$
\begin{aligned}
\dot{y} &= \frac{dh}{dt} = \frac{\partial h}{\partial x}\frac{dx}{dt} = L_f h(x) \\
\ddot{y} &= L_f^2 h(x) \\
&\vdots \\
\frac{d^{\alpha-1}y}{dt^{\alpha-1}} &= L_f^{\alpha-1} h(x) \\
\frac{d^{\alpha}y}{dt^{\alpha}} &= L_f^{\alpha} h(x) + L_g L_f^{\alpha-1} h(x)\, u
\end{aligned}
$$

Thus for finite relative order α, the first $\alpha - 1$ derivatives of y, with respect to time, are just functions of the state vector x, while the αth derivative of y, also with respect to time, is a function of the input u, i.e. the relative order α is exactly equal to the number of times one has to differentiate the output $y(t)$ at time $t = t_0$ in order to have the value $u(t_0)$ of the input explicitly appearing.

From the above definitions, it is clear that if no integer α exists, then the system is of infinite relative order. In general however, the (finite) relative order of a system is bounded by the dimension of the state space of the system [3]. This concept of relative order allows one to determine whether the mapping between the input and output is one to one.

For our system with Relative Order 1,

$$L_g h(x) \neq 0$$

Now,

$$\begin{aligned} L_f h(x) &= \frac{\partial h}{\partial x} f(x) \\ &= W_3^T \frac{\partial \eta_3}{\partial x} W_1 \eta_1 \\ L_g h(x) &= \frac{\partial h}{\partial x} g(x) \\ &= W_3^T \frac{\partial \eta_3}{\partial x} W_2 \eta_2 \end{aligned}$$

$$\begin{aligned} \frac{dy}{dt} &= L_f^1 h(x) + L_g L_f^0 h(x) u \\ &= W_3^T \frac{\partial \eta_3}{\partial x} W_1 \eta_1 + W_3^T \frac{\partial \eta_3}{\partial x} W_2 \eta_2 u \end{aligned}$$

B. Inverting Recurrent Neural Networks

We consider the left inverse of the nonlinear system (5.1) that reconstructs the input from the plant output, its derivatives, and the state variables of the inverse (Fig. 6).

Using the concept of relative order, Hirschorn [6] was able to construct a state-space realization of the process inverse. His major result is summarized in the following theorem:

Theorem 4 *Hirschorn Inversion Theorem: The inverse of a dynamical nonlinear system given by (5.1) is as follows:*

$$\dot{Z} = f(Z) + g(Z)\, u' \tag{6.1}$$

where

$$u' = \frac{\frac{d^\alpha y}{dt^\alpha} - L_f^\alpha\, h(Z)}{L_g\, L_f^{\alpha-1}\, h(Z)} \tag{6.2}$$

where u' is the input to the left inverse system, and Z is the state of the inverse. $f(.), g(.)$ and $h(.)$ are the same as in the original system and $L_f^\alpha h(Z)$ and $L_g L_f^{\alpha-1} h(Z)$

are Lie derivatives. α is the relative order of the system.

Thus the inverse of the recurrent neural network is the network itself.

For $\alpha = 1$,

$$u' = \frac{\frac{dy}{dt} - L_f h(Z)}{L_g h(Z)}$$

$$L_g h(Z) \neq 0$$

$$\Rightarrow W_3^T \frac{\partial \eta_3}{\partial x} W_2 \eta_2 \neq 0$$

$$\therefore u' = \frac{\left[W_3^T \frac{\partial \eta_3}{\partial x} W_1 \eta_1 + W_3^T \frac{\partial \eta_3}{\partial x} W_2 \eta_2 u\right] - W_3^T \frac{\partial \eta_3}{\partial Z} W_1 \eta_1}{W_3^T \frac{\partial \eta_3}{\partial Z} W_2 \eta_2}$$

CHAPTER VII

SIMULATIONS

An important bioengineering process is the fermentation of penicillin. The dynamics of a continuous glucose-fed system can be modeled as [25]:

$$\begin{aligned} \dot{X}_1 &= \mu(X_2, X_1)\, X_1 - X_1\, D \\ \dot{X}_2 &= -\sigma(X_2, X_1)\, X_1 + (S_F - S)\, D \\ \dot{P} &= \pi(X_2)\, X_1 - KP - DP \end{aligned} \tag{7.1}$$

where X_1 is the cell mass concentration, P is the penicillin concentration, S_F is the glucose feed concentration, and D is the dilution rate of glucose added to the fermenter. The parameters $\mu(X_2, X_1)$, $\pi(X_2)$ and $\sigma(X_2, X_1)$ are the specific growth rate, specific product formation rate, and specific substrate utilization rates, respectively, and are given by empirical expressions:

$$\begin{aligned} \mu(X_2, X_1) &= \frac{\mu_x X_2}{K_x X_1 + X_2} \\ \pi(X_2) &= \frac{\mu_p X_2}{K_p + X_2(1 + X_2/K_1)} \\ \sigma(X_2, X_1) &= \frac{1}{Y_{ss}} \mu(X_2, X_1) + \frac{1}{Y_{ps}} \pi(X_2) + m \end{aligned}$$

The parameter K is the degradation constant of penicillin.

The dilution rate D is the most convenient manipulated input for the system. The cell mass concentration X_1 can be very accurately estimated on-line through CO_2 and O_2 measurements and atom balances. So, X_1 is a natural output for the system.

With $u = D$ as input and $y = X_1$ as output, the input-output relationship in a continuous penicillin fermenter is described by:

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} &= \begin{bmatrix} \mu(X_2, X_1)X_1 \\ -\sigma(X_2, X_1)X_1 \end{bmatrix} + \begin{bmatrix} -X_1 \\ S_F - S \end{bmatrix} u \\ y &= X_1 \end{aligned} \tag{7.2}$$

We now compute the Hirschorn inverse of (7.1). Since $L_g h(X_1, X_2) = -X_1 \neq 0$, it follows that the relative order of (7.1) is 1. Thus, the Hirschorn inverse can be calculated via:

$$\begin{aligned}\dot{Z} &= f(Z) + g(Z)\frac{dy/dt - L_f h(Z)}{L_g h(Z)} \\ u &= \frac{dy/dt - L_f h(Z)}{L_g h(Z)}\end{aligned}$$

Substituting the particulars f, g, and h, we obtain the following inverse of (7.1) :

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} &= \begin{bmatrix} \mu(Z_2, Z_1)Z_1 \\ -\sigma(Z_2, Z_1)Z_1 \end{bmatrix} + \begin{bmatrix} -Z_1 \\ S_F - S \end{bmatrix} \frac{dy/dt - \mu(Z_2, Z_1)Z_1}{-Z_1} \\ u &= \frac{dy/dt - \mu(Z_2, Z_1)Z_1}{-Z_1}\end{aligned} \tag{7.3}$$

We first simulate the Dynamical System using the parameters in Table I and get the States as in Fig. 6. We then estimate the states using Neural Networks whose weights are updated by Adaptive Laws as in Fig. 7. We have plotted the estimated States in Fig. 8, and the error between the actual and estimated states in Fig. 9. We then simulate the inverse of the system and fitting everything in the IMC framework and with a setpoint of 1.6, we plot the diagram in Fig. 10.

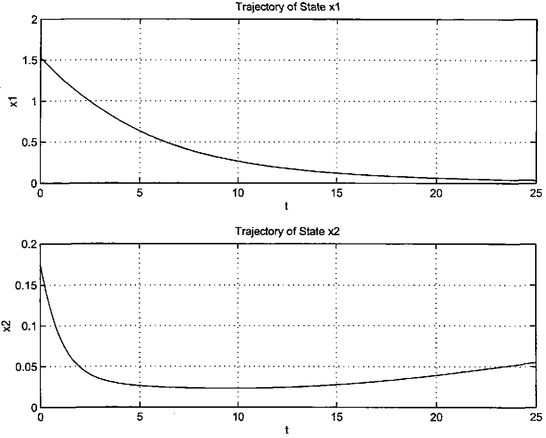

Fig. 6. Actual States

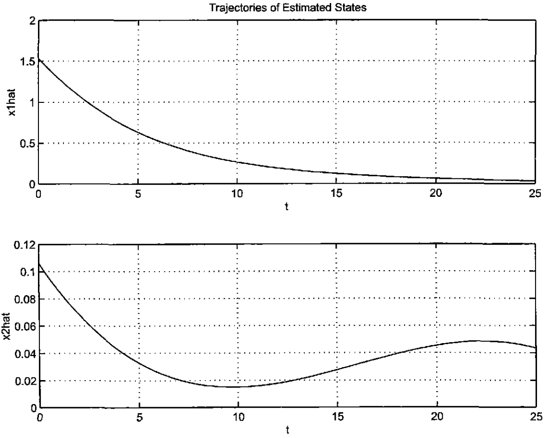

Fig. 7. Estimated States

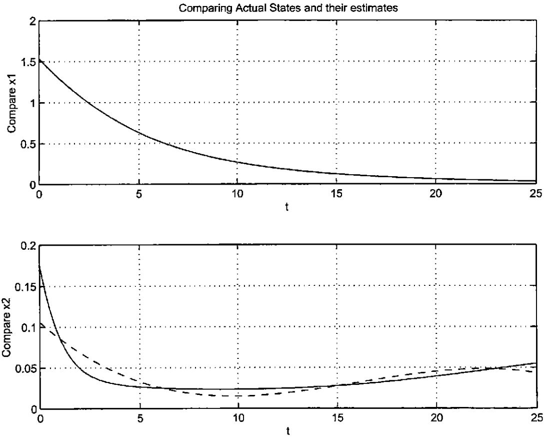

Fig. 8. Comparing Actual and Estimated States

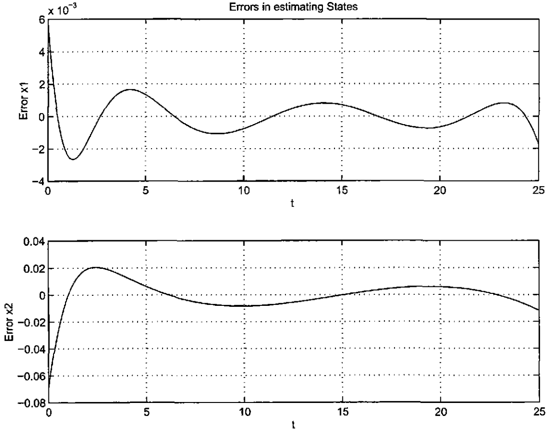

Fig. 9. Error in Actual and Estimated States

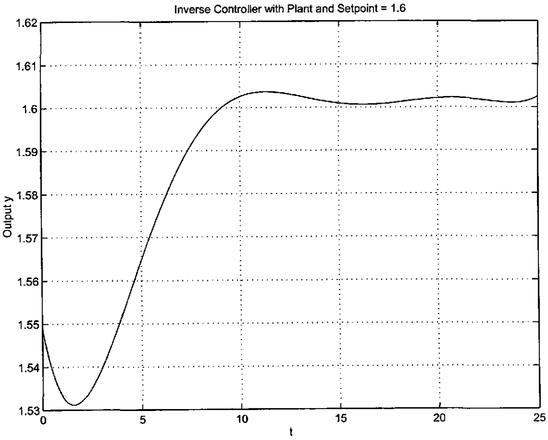

Fig. 10. Output of the Resulting IMC System

The following set of parameters were used for the simulations:

$$\mu_x = 0.092$$

$$K_x = 0.15$$

$$\mu_p = 0.005$$

$$K_p = 0.0002$$

$$K_1 = 0.1$$

$$\mathrm{K} = 0.04$$

$$\mathrm{m} = 0.014$$

$$S_F = 0.1$$

$$Y_{ss} = 0.45$$

$$Y_{sp} = 0.9$$

$$P_0 = 0.33$$

$$X_1^0 = 1.5302 \text{ (Initial Condition)}$$

$$X_2^0 = 0.17459 \text{ (Initial Condition)}$$

Table I. Parameter Values for Simulations

CHAPTER VIII

CONCLUDING REMARKS

In this thesis, an IMC based adaptive control scheme was designed and presented for single-input single output, first order minimum phase nonlinear systems by using Neural Networks. We used Radial Basis function Neural Network Architectures for identifying and controlling dynamical systems with unknown nonlinearities. The Nonlinear IMC controller consists of a model inverse controller and a robustness filter with a single tuning parameter. The inverse controller of the IMC strategy was produced using the Hirschorn inversion theory [6][24]. This is in marked contrast to other procedures for establishing the inverse of a neural network model [26] [27], since usually a separate training scheme is used.

The principal motivation for undertaking such a study was the immense popularity of IMC in industrial applications, on one hand, coupled with the capability of Neural Networks to uniformly approximate continuous nonlinear functions. Although we have considered RBF networks, other neural network architectures like multilayer networks with sigmoidal type of activation functions can also be used. In view of the flexibility that neural networks provide in modeling poorly understood processes, the proposed NIMC strategy is potentially applicable to a wide class of process control problems.

In addition to RBF networks, there are, of course, other representations that can be used for approximating static maps. Considerably more studies, both of theoretic and of practical nature, need to be performed before it is clear which architecture is best for approximating different classes of functions. The factors that influence how well a network is constructed, such as the number of weights and number of layers, are chosen for the most part by trial and error or other ad-hoc techniques. Also, the position of the centers and widths of the RBF networks were assumed. Further research is required for effectively choosing these quantities and to provide stability guarantees of new recurrent neural network architectures, if any.

REFERENCES

[1] M.Morari and E.Zafiriou, *Robust Process Control,* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[2] C.E.Garcia,D.M.Prett, and M.Morari, "Model predictive control: Theory and practice - a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[3] A.Isidori, *Nonlinear Control Systems,* Springer-Verlag, New York, 1989.

[4] K.S.Narendra and K.Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks,* vol. 1, pp. 4–27, Mar. 1990.

[5] Economou,C.E.Garcia,M.Morari and B.O.Palsson, "Internal model control.5.extension to nonlinear systems," *Ind. Eng. Chem. Process Des. Dev.*, vol. 25, pp. 403–411, 1986.

[6] R.M.Hirschorn, "Invertibility of multivariable nonlinear control systems," *IEEE Trans. Autom. Control,* vol. 24, pp. 855–865, 1979.

[7] M.A.Henson and D.E.Seborg, Ed., *Nonlinear Process Control,* Prentice Hall, Upper Saddle River, NJ, 1997.

[8] N.B.Haaser and J.A.Sullivan, *Real Analysis,* Van Nostrand Reinhold, New York, 1971.

[9] J.J.Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proc, Nat. Acad. Sci.*, 1982, vol. 79, pp. 2554–2558.

[10] M.M.Polycarpou and P.A.Ioannou, "Identification and control of nonlinear systems using neural network models: Design and stability analysis," Tech. Rep., University of Southern California, Los Angeles, CA, September 1991.

[11] D.S.Broomhead and D.Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[12] J.Moody and C.J.Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281–294, 1989.

[13] S.Renals and R.Rohwer, "Phoneme classification experiments using radial basis functions," in *Proc. Intern. Join Conf. on Neural Networks*, 1989, vol. 1, pp. 461–467.

[14] R.M.Sanner and J.J.E.Slotine, "Gaussian networks for direct adaptive control," in *Proc. Autom. Control. Conf.*, 1991, pp. 2153–2159.

[15] J.Park and I.W.Sandberg, "Universal approximation using radial-basis function networks," *Neural Computation*, vol. 3, pp. 246–257, 1990.

[16] J.K.Hale, *Ordinary Differential Equations*, Wiley-InterScience, New York, 1969.

[17] G.H.Golub and C.F.Van Loan, *Matrix Computations*, The John Hopkins Univ. Press, Baltimore, 2nd edition, 1989.

[18] P.C.Parks, "Lyapunov redesign of model reference adaptive control systems," *IEEE Trans. Aut. Control*, vol. AC-11, pp. 362–367, 1966.

[19] G.C.Goodwin and D.Q.Mayne, "A parameter estimation perspective of continuous time model reference adaptive control," *Automatica*, vol. 23, pp. 57–70, Jan. 1987.

[20] P.A.Ioannou and A.Datta, *Foundations of Adaptive Control*, pp. 71–152, Springer-Verlag, Berlin, 1991.

[21] S.Sastry and M.Bodson, *Adaptive Control: Stability, Convergence and Robustness*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[22] P.A.Ioannou and J.Sun, *Robust Adaptive Control*, Prentice-Hall, Upper Saddle River, NJ, 1996.

[23] R.M.Hirschorn, "Invertibility of multivariable nonlinear control systems," *IEEE Trans. on Aut. Control*, vol. AC-24, no. 6, pp. 855–865, 1979.

[24] C.Kravaris and J.C.Kantor, "Geometric methods for nonlinear process control. 1. background," *Ind. Eng. Chem. Res.*, vol. 29, pp. 2295–2310, 1990.

[25] R.K.Bajpai and M.A.Reuβ, "Mechanistic model for penicilin production," *J. Chem. Technol. Biotechnol.*, vol. 30, pp. 332–344, 1980.

[26] K.Hunt and D.Sbarbaro, "Neural networks for nonlinear internal model control," *IEE Proceedings-D*, vol. 138, no. 5, pp. 431–438, 1991.

[27] G.Lightbody and G.W.Irwin, "Nonlinear control structures based on embedded neural system models," *IEEE Trans. Neural Networks*, vol. 8, no. 3, pp. 553–567, 1997.

VITA

- Amit Krushnavadan Gandhi; Mulund, Bombay 400080, India.
- EDUCATION: M.S. in Electrical Engineering earned in August 2001 from Texas A&M University in College Station, Texas. **B.S.** in Electrical Engineering earned in June 1999 from University of Bombay in Bombay, India.
- RESEARCH EXPERIENCE: Research Assistant in the Department of Electrical Engineering at Texas A&M University from January 2000 through August 2000.
- TEACHING EXPERIENCE: Teaching Assistant in the Department of Electrical Engineering at Texas A&M University from September 1999 through December 1999 and from September 2000 through December 2000.
- PROJECTS:
  - Designing a simple UDP based protocol/application for retrieving files from a server that addresses duplication and uses stop and wait transmission of data.
  - LCAM(Life Cycle Artifact Manager):Design and development of a Lifecycle Artifact Manager, a software tool to aid the software development process. Involves usage of C++ and Rum Baugh's modeling techniques.
  - Theory and control of Chaos: Control and stabilization of the logistic map (chaotic!) using the OGY method.
  - Implementation of Voice Over IP using JDK1.2 and JMF 2.0.