Neural Networks for Control: Theory and Practice

KUMPATI S. NARENDRA, FELLOW, IEEE

Invited Paper

The past five years have witnessed a great deal of progress in both the theory and the practice of control using neural networks. After a long period of experimentation and research, neural network-based controllers are finally emerging in the marketplace, and the benefits of such controllers are now being realized in a wide variety of fields. The practical applications are also calling for a better understanding of the theoretical principles involved. In this paper we review the current status of control practice using neural networks and the theory related to it and attempt to assess the advantages of neurocontrol for technology.

I. Introduction

The field of control is inherently interdisciplinary in nature and extends from design, development, and production on the one hand to mathematics on the other. Since the very beginning over six decades ago, control research has been driven by the diverse and changing nature of technology. Today, control techniques have become pervasive in a wide spectrum of applications which are of major scientific, technological, and economic importance. Mathematical control theory has been described as an indispensable component of a partnership between mathematical theory, engineering practice, and hardware and software capabilities.

The objective of control is to influence the behavior of dynamical systems. The latter includes maintaining the outputs of systems at constant values (regulation) or forcing them to follow prescribed time functions (tracking). The control problem is to determine the control inputs to the system using all available data. Achieving fast and accurate control even while assuring stability and robustness is the aim of all control systems design.

The best developed part of control theory deals with linear systems and powerful methods for designing controllers for such systems are currently available. In fact, most of the controllers used in modern industry belong to this class. However, as applications become more complex, the processes to be controlled are increasingly characterized by poor models, distributed sensors and actuators, multiple subsystems, high noise levels and complex information patterns. The difficulties encountered in designing controls for such processes can be broadly classified under three headings: 1) complexity, 2) nonlinearity, and 3) uncertainty. The control problems addressed in this paper exhibit one or more of the above characteristics.

The term artificial neural networks (ANN's) has come to mean any architecture that has massively parallel interconnection of simple processors. From a systems theoretic point of view, a neural network can be considered as a conveniently parameterized class of nonlinear maps. During the 1980's and the early 1990's, conclusive proofs were given by numerous authors that multilayer feedforward networks (MNN's) are capable of approximating any continuous function on a compact set in a very precise and satisfactory sense. As a result, such networks found wide application in many fields, both for function approximation and pattern recognition. Since dynamics is an essential part of physical systems, it was proposed in [1] that neural networks should be used as components in dynamical systems and that a study of such networks should be undertaken within a unified framework of systems theory. A great deal of progress has been made since that time both in the theory and practice of control using neural networks, i.e., the field of neurocontrol. Numerous dynamical systems have been identified and controlled in computer simulations, and a few have been practically implemented. It has become increasingly evident (Sections II and VI) that ANN's are capable of coping with all three categories of difficulties which arise in complex control systems.

This paper is written with three objectives in mind. The first is to review the current status of control using neural networks. The theoretical background needed in this context is provided in Sections III–V. The second objective is to discuss practical considerations which call for control solutions based on neural networks. The final objective is

Manuscript received July 1, 1996; revised July 17, 1996. This work was supported by the National Science Foundation under Grant ECS-9521405 and the Office of Naval Research under Grant N00014-95-1-0844.

The author is with the Center for Systems Science, Yale University, New Haven, CT 06520-8267 USA (e-mail: narendra@koshy.eng.yale.edu).

Publisher Item Identifier S 0018-9219(96)07478-6.

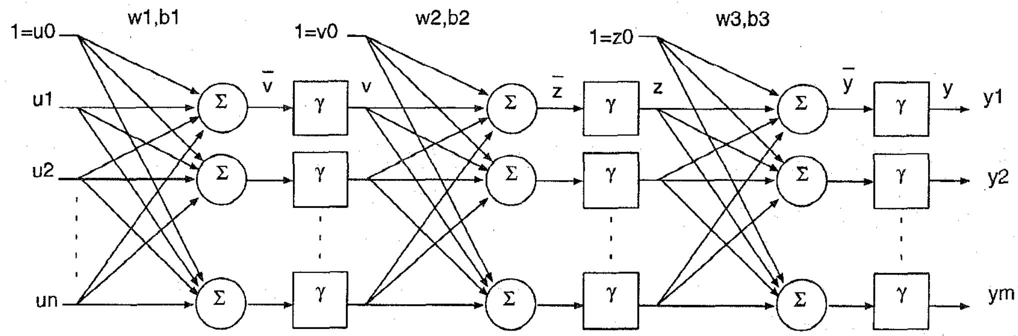


Fig. 1. A three-layer neural network.

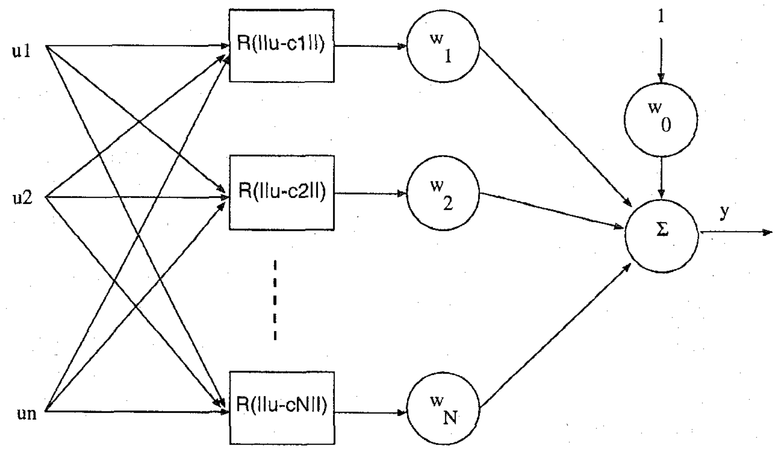


Fig. 2. The RBFN.

to describe a number of successful applications which have been achieved with relatively simple control concepts and to assess the need for further advances in theory to increase the potential of neurocontrol in future applications.

II. Neural Networks

In the 1980's the MNN shown in Fig. 1 was introduced for approximating continuous functions. Later, the radial basis function network (RBFN) shown in Fig. 2 was proposed as a viable alternative. The n layer MNN with input u and output y is described by the equation

$$\Gamma[W_n\Gamma[W_{n-1}\cdots\Gamma[W_1u+b_1]+\cdots+b_{n-1}]+b_n]=y$$

where W_i is the weight matrix in the ith layer, and the vectors b_i $(i=1,2,\cdots,n)$ represent the threshold values for each node in the ith layer. $\Gamma[\cdot]$ is a nonlinear operator with $\Gamma(x)=[\gamma_1(x),\gamma_2(x),\cdots,\gamma_n(x)]$ where $\gamma_i(\cdot)$ is a smooth activation function (generally a sigmoid).

In Fig. 2, the structure of the RBFN with an input $u\in\mathbb{R}^n$ and output $y\in\mathbb{R}$ is shown. The output is described by the equation

$$y=f(u)=\sum_{i=1}^{N}W_iR_i(u)+W_0$$

where W_i $(i=0,1,2,\cdots,N)$ are the weights, and the function $R_i\colon\mathbb{R}^n\to\mathbb{R}$ have generally the form

$$R_i(u)=\exp\left[-\sum_{j=1}^{n}\frac{(u_j-c_{ij})^2}{2\sigma_{ij}^2}\right]$$

where $c_i^T=[c_{i1},\cdots,c_{in}]$ is the center of the ith receptive field and σ_{ij} is referred to as its width. The matrices W_i and the vectors b_i denote the adjustable parameters of the MNN. In general, the parameters W_i, the centers c_i and the width σ_{ij} constitute the adjustable parameters of the RBFN. If, as in some cases c_i and σ_{ij} are chosen before the network is used in a specific application, the mapping represented by the RBFN becomes linear in the unknown parameters W_i. If a network (MNN or RBFN) is chosen to approximate a given mapping using input–output (*IO*) data, the parameter vector of that network is adjusted so that some specified norm $\|y-\hat{y}\|$ is minimized, where y is the output of the given map and $\hat{y}$ is the output of the neural network.

Both MNN and RBFN are capable, at least in theory, of coping with complexity, nonlinearity, and uncertainty encountered in complex systems. The massively parallel nature of the MNN permits computation to be performed at high rates. Since they can approximate nonlinear maps to any desired degree of accuracy, they can also be used to identify and control nonlinear dynamical systems. Finally, the fact that various algorithms are currently available for the adjustment of the parameters of the networks implies that they can deal with uncertainty, by realizing approximations of unknown static and dynamic mappings, from IO data.

While it has been shown that both MNN and RBFN can approximate arbitrary functions from one finite dimensional space to another to any desired degree of accuracy [2]–[5], it is also true that polynomials, trigonometric series, splines, and orthogonal functions share the same properties. Hence, questions naturally arise as to why ANN should be preferred over such methods. Extensive computer studies carried out during the past years have revealed that neural networks enjoy numerous practical advantages over conventional methods. In view of their architecture they are more fault tolerant and less sensitive to noise and they are more easily implementable in hardware because of the parametrization used. Barron's [6] work has also provided partial theoretical justification for using ANN's over its competitors. If the class of functions F is restricted to those whose Fourier transforms have a first moment, and $f: \mathbb{R}^d \rightarrow \mathbb{R}$ is a member of this class, the problem of approximating it with a network having one hidden layer and T nodes in that layer is considered [6]. If f_T is the map obtained by the neural network, it is shown that the L_2 norm of the approximation error $\|f - f_T\|_2$ is bounded by $O\,(C_f/T^{1/2})$ where C_f is the first moment of the Fourier transform. In contrast to this, no linear combination of T fixed basis functions can achieve an approximation error smaller than $O\,(C_f/T^{1/d})$. Hence, as the dimensionality of input space increases, it is clear that MNN are preferable to approximation schemes in which the adjustable parameters arise linearly.

The above theoretical result has great significance for the design of practical controllers for dynamical systems, since the dimensionality of the input space is invariably large in such cases. Both MNN, in which the parameters occur nonlinearly, and RBFN in which c_{ij} and σ_{ij} are adjusted, require substantially fewer parameters for a desired degree of accuracy.

III. Mathematical Preliminaries

The identification and control of nonlinear dynamical systems using neural networks was first proposed in 1990 [1]. It was suggested that procedures similar to those adopted successfully for the control of linear systems involving a combination of mathematics, modeling, computation, and experimentation should also be attempted in the nonlinear case. Although the mathematical properties of nonlinear control systems had been extensively studied by this time, no constructive procedures existed by which controllers could be synthesized for systems which were distinctly nonlinear. In the years following the publication of [1], extensive computer simulations were carried out, very much in the nature of experimental research in the pure sciences, to determine the efficacy of neural networks in control problems. The studies, apart from indicating the potential of neural networks as identifiers and controllers in relatively simple problems, also indicated the need for sound theoretical principles to guide the practical development. At the same time, attempts to apply the methods proposed to practical systems also raised new theoretical questions. Since the control of a nonlinear system is a difficult problem, fairly stringent assumptions were made in the early stages regarding the system to be controlled to assure the existence of solutions. Gradually these assumptions were relaxed and advances of an incremental nature were made. After six years of investigation, the number of theoretical results continues to grow and is providing better insight into the nature of the problems involved. The limitations of the methods currently available, and the areas where further progress is needed are also better understood.

Since an appreciation of the theoretical principles involved is essential for the efficient design of practical controllers, this section deals with some of the basic concepts and results that are currently available for the identification and control of dynamical systems.

A. Nonlinear Dynamical Systems

We shall assume in all the following sections that the nonlinear dynamical system Σ to be controlled can be described by the state equations

$$\begin{aligned} \Sigma: x(k+1) &= f[x(k), u(k)],\ f(0,0) = 0 \\ y(k) &= h[x(k)],\ h(0) = 0 \end{aligned} \tag{1}$$

where $u(k), y(k) \in \mathbb{R}^m$ and $x(k) \in \mathbb{R}^n$ and represent the input, output, and state vectors, respectively, at time k. The functions $f: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $h: \mathbb{R}^n \rightarrow \mathbb{R}^m$ are assumed to be smooth. Qualitatively, our objective is to determine conditions under which an input vector $u(k)$ can be found so that the outputs of the system behave in some desired fashion. If $m = 1$, the system has a single input and single output (SISO) system. If $m > 1$, the system has a multiple input and multiple output (MIMO) system.

The following two theorems are basic to the derivation of almost all the results discussed in this paper.

Theorem 1 (Inverse Function [7]): Let f be a C^1 mapping of an open set $E \in \mathbb{R}^n$ into $\mathbb{R}^n$. Let J be the Jacobian of f at any point x, so that $J(x) = f'(x)$. 1) If $J(a)$ is invertible at any point $a \in E$, and $b = f(a)$, then there exist open sets U and V in $\mathbb{R}^n$ such that $a \in U, b \in V$ and f is one-to-one on U and $f[U] = V$. 2) If g is the inverse of f [which exists by 1)] defined in V by $g[f(x)] = x,\ x \in U$, then $g \in C^1(V)$.

By the above theorem, the function f is well behaved in the neighborhood of a point a where the Jacobian is nonsingular and an inverse g exists in this neighborhood. The Implicit Function Theorem given below extends the results to situations where x and y are implicitly related.

Theorem 2 (Implicit Function [7]): Let f be a C^1 mapping of an open set $E \in \mathbb{R}^{n+m}$ into $\mathbb{R}^n$ such that $f(a,b) = 0$ for some point (a,b) in E. Let $f_x(x,y)$ be the Jacobian of f with respect to x at the point (x,y) in E and let $f_x(a,b)$ be invertible. Then there exists an open set $U \in \mathbb{R}^{n+m}$ and $W \subset \mathbb{R}^m$ with $(a,b) \in U$ and $b \in W$

such that to every $y \in W$ there corresponds a unique x satisfying $f(x,y) = 0$, $(x,y) \in U$. If this x is defined to be $g(y)$, then $g \in C^1$ and is a mapping of W into $\mathbb{R}^n$, $g(b) = a$, and $f(g(y),y) = 0$ for $y \in W$.

The importance of the above theorems for neurocontrol is that they can be used to prove the existence of exact local nonlinear models based on the properties of the linearized system described in Section III-B.

From a purely mathematical point of view, the precise control of a nonlinear dynamical system is a formidable one. It becomes substantially more difficult when uncertainty is also present in the system. An approach that has proved successful during the past years is that by restricting the class of nonlinear systems to those whose linearizations are well behaved at an equilibrium state, the implicit function theorem can be used to assure the existence of appropriate nonlinear maps in some open domain containing the equilibrium state. Neural networks can then be used to approximate these maps using the data available concerning the system. Computer simulations of a large number of nonlinear systems have experimentally justified this approach, and have resulted in nonlinear controllers which yield significantly better performance than linear controllers. In view of the important role played by the linearization of the nonlinear system in the above approach, we consider it in detail in the following section.

B. The Linearized System Σ_L

As mentioned in the introduction, the best developed part of control theory deals with linear time-invariant systems. A number of properties of such systems can be found in standard textbooks on control theory [8], [9]. From the approach proposed in the preceding section it is clear that the properties of the linearized system at an equilibrium state are closely related to the existence of solutions for the class of nonlinear control problems described in the paper. We consider briefly some of these properties in this section and relegate to following sections a more detailed discussion of their implications in nonlinear control.

The linearization of Σ around (0, 0) is denoted by Σ_L where

$$\begin{aligned}\Sigma_L\colon x(k+1) &= Ax(k) + Bu(k)\\ y(k) &= Cx(k)\end{aligned} \tag{2}$$

where the $(n \times n)$ matrix A and the $(n \times m)$ and $(m \times n)$ vectors b and c are defined by

$$\left.\frac{\partial f(x,u)}{\partial x}\right|_{(0,0)} = A; \quad \left.\frac{\partial f(x,u)}{\partial u}\right|_{(0,0)} = b; \quad \left.\frac{\partial h(x)}{\partial x}\right|_{0} = c.$$

The discrete-time multivariable system Σ_L with m-inputs and m-outputs described by (2) can be represented in several equivalent forms. If z denotes the shift operator (i.e., $zx(k) = x(k+1)$), Σ_L can be described by its transfer matrix $W(z)$ where

$$W(z) = C(zI - A)^{-1}B. \tag{3}$$

For SISO systems, c and b are vectors, $W(z)$ is a transfer function, and

$$W(z) = c(zI - A)^{-1}b.$$

By the successive application of (2) we have

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k)\\ x(k+2) &= A^2x(k) + ABu(k) + Bu(k+1)\\ \cdots &\qquad \cdots\\ x(k+n) &= A^n x(k) + \sum_{i=0}^{n-1} A^{n-1-i}Bu(k+i)\end{aligned} \tag{4}$$

and

$$y(k+i) = Cx(k+i) \quad i = 0,1,2,\cdots,n \tag{5}$$

so that

$$y(k+n) = CA^n x(k) + \sum_{i=0}^{n-1^2} CA^{n-1-i}Bu(k+i). \tag{6}$$

Equation (6) relates the state at any instant k and the inputs applied at instants $k, k+1, \cdots$, to the output at $k+n$. If the initial state $x(0) = 0$, it follows from (6) that

$$y(n) = \sum_{i=0}^{n-1} CA^{n-1-i}Bu(i) = \sum_{i=0}^{n-1} CA^iBu(n-1-i) \tag{7}$$

or the output at any instant can be expressed as a linear combination of the past values of the input. The coefficients of the right hand side of (7) (i.e., CA^iB) are called the Markovian parameters of the system.

1) Stability, Controllability, and Observability: Many of the fundamental properties of Σ_L can be stated in terms of the matrices A, B, and C. These include the system theoretic properties of stability, controllability and observability. Qualitatively, the system is said to be stable if small changes in initial conditions result in small changes in the state trajectories and asymptotically stable if the latter tends to zero asymptotically with time. The stability of Σ_L is determined entirely by the matrix A and it is known that Σ_L is asymptotically stable if the eigenvalues of A lie in the interior of the unit circle in the complex plane.

The ability to control the system depends on matrices A and B. The system Σ_L is said to be controllable if and only if every initial state x_I can be transformed to any final state x_F in a finite number of steps. From (4), it is clear that if the rank of the $(n \times nm)$ matrix

$$W_c = [B, AB, \cdots, A^{n-1}B]$$

is n, the system is controllable. For SISO systems in which $m = 1$ and $B = b$ (a vector), W_c is an $(n \times n)$ matrix and the nonsingularity of W_c is necessary and sufficient for controllability.

Observability of Σ_L implies that the initial state $x(0)$ (and hence the states $x(i)$, $i > 0$) can be determined by measuring the output at a finite number of time instants. Observability of Σ_L is determined by the matrices C and A and from (4) and (5). It follows that Σ_L is observable if the

$(n \times mn)$ matrix $W_o^T = [C^T, A^TC^T, \cdots, (A^T)^{n-1}C^T]$ is of rank n. For SISO systems W_o is an $(n \times n)$ matrix and the nonsingularity of W_o implies observability.

If Σ_L is controllable it is known that it is also stabilizable using state feedback, i.e., a matrix K exists such that $u = Kx$ makes the overall system asymptotically stable. Further, K can be chosen so that the overall system (i.e., $A + BK$) has desired eigenvalues. If, in addition, Σ_L is observable, the state of the system $x(i)$ at time i can be estimated in a stable fashion using IO data. Denoting the estimate by $\hat{x}(i)$, it can also be shown that the use of a control input $u(i) = K\hat{x}(i)$ will result in an asymptotically stable system.

The brief discussion above indicates the importance of controllability and observability in linear time-invariant systems. In the following sections, while considering the control of the nonlinear dynamical system Σ described by (1), we shall assume that the linearization Σ_L of Σ is both controllable and observable.

Comment: From a theoretical point of view, the assumption that Σ_L is both controllable and observable is a fairly stringent one. However, our main interest is in the practical control of nonlinear systems. Many of these systems which currently exist are controlled by linear controllers and perform satisfactorily while operating in a small neighborhood of the equilibrium state. This indicates that their linearizations do satisfy the above conditions. It is due to the demands of a growing technology that such systems are required to operate in larger regions in the state space, making them distinctly nonlinear.

2) IO Representation: From (4) and (5) it was seen that if Σ_L is observable, the state $x(k)$ at time k can be determined from a knowledge of $u(k)$, $u(k+1), \cdots, u(k+n-2)$ and $y(k)$, $y(k+1), \cdots, y(k+n-1)$. Using the notation $V_n(k)$ to denote any sequence $v(k)$, $v(k+1), \cdots, v(k+n-1)$ of length n, starting at instant k, it follows that

$$x(k) = \gamma^T \begin{bmatrix} Y_n(k) \\ U_{n-1}(k) \end{bmatrix}$$

where γ is a constant vector with $\gamma \in \mathbb{R}^{2n-1}$. Using (6) and appropriately shifting the time axis, we have the important result for SISO systems that

$$y(k+1) = \sum_{i=0}^{n-1} \alpha_i y(k-i) + \sum_{j=0}^{n-1} \beta_j u(k-j) \tag{8}$$

or equivalently that the system has an IO representation which is completely determined by $2n$ parameters $(\alpha_i, \beta_j, i, j = 1, 2, \cdots, n)$. The same also applies to MIMO systems where

$$y(k+1) = \sum_{i=0}^{n-1} A_i y(k-i) + \sum_{j=0}^{n-1} B_j u(k-j) \tag{9}$$

where A_i and B_j are constant $(m \times m)$ matrices. The representations (8) and (9) are also referred to as the SISO and MIMO autoregressive moving average (ARMA) models.

3) Relative Degree: Equation (6) relates the output at instant $(k+n)$ to the state at instant k and the inputs at the following n instants $(k, k+1, \cdots, k+n-1)$. If $cb \neq 0$, the input at time $(k+n-1)$ affects the output at time $k+n$. If, however, $cb, cAb, \cdots, cA^{d-2}b$ are all zero and $cA^{d-1}b \neq 0$, it follows that the input at any instant affects the output d instants later. In this case the output at time $(k+d)$ is determined by $x(k)$ and $u(k)$ as follows:

$$y(k+d) = cA^d x(k) + cA^{d-1} bu(k). \tag{10}$$

The right-hand side of (10) does not contain $u(k+1), u(k+2), \cdots$, or $u(k+d-1)$ since they do not affect $y(k+d)$. The integer d is called the relative degree of the system. It also represents the delay experienced before the input affects the output.

By successively expressing $y(k+1), y(k+2), \cdots, y(k+d-1)$ in (8) in terms of inputs and outputs at previous instants, a modified representation can be obtained for a linear time-invariant system with relative degree d. This has the general form

$$y(k+d) = \sum_{i=0}^{n-1} \bar{\alpha}_i y(k-i) + \sum_{j=0}^{n-1} \bar{\beta}_j u(k-j)$$

where $\bar{\alpha}_i$ and $\bar{\beta}_j\, (i, j = 0, 1, 2, \cdots, n-1)$ are the parameters of the system. In the identification and control of nonlinear systems, it is this equation which finds wide application.

In the case of a multivariable systems with m inputs and m outputs described by (9), there are m^2 transfer functions between inputs and outputs and hence a different relative degree can be defined for each of them. If d_{ij} represents the delay from the jth input to the ith output, the relative degree of the ith output is defined as d_i where

$$d_i = \min_j d_{ij}$$

or d_i is the smallest interval of time before an input can affect the output y_i. Defining $d_1, d_2, \cdots, d_m$ in a similar fashion we obtain the relative degree of all the outputs. Using the same arguments as before, it can be shown that the MIMO system can be described by the equation

$$Y(k+d) \stackrel{\text{def}}{=} \begin{bmatrix} y_1(k+d_1) \\ y_2(k+d_2) \\ \vdots \\ y_m(k+d_m) \end{bmatrix} = \begin{bmatrix} CA^{d_1}x(k) + E_1 u(k) \\ CA^{d_2}x(k) + E_2 u(k) \\ \vdots \\ CA^{d_m}x(k) + E_m u(k) \end{bmatrix} \tag{11}$$

where $E_i = c_i A^{d_i - 1} B$ and c_i is the ith row of C. E_i is a $(1 \times m)$ row vector and the matrix E which has E_i for its ith row is central to all control problems dealing with multivariable systems. It is evident from (11) that if E is a nonsingular matrix, $u(k)$ can be chosen so that $y_1(k+d_1), y_2(k+d_2), \cdots, y_m(k+d_m)$ have desired values.

4) Minimum Phase System: The polynomial $\beta_0 z^{n-1} + \beta_1 z^{n-2} + \cdots + \beta_{n-1}$ is called a stable polynomial, where $\beta_i \in \mathbb{R}$, if its roots lie in the interior of the unit circle. An SISO system is called a minimum phase system if its transfer function $W(z)$ [given by (3)] has a numerator polynomial which is stable, i.e.,

$$W(z) = \frac{\beta(z)}{\alpha(z)} = \frac{\beta_0 z^{n-1} + \cdots + \beta_{n-1}}{z^n + \alpha_0 z^{n-1} + \cdots + \alpha_{n-1}}$$

and $\beta(z)$ is a stable polynomial.

If an LTI system is nonminimum phase, an unbounded input can result in a bounded output or even one which tends to zero. For example if $W(z) = \frac{z-2}{(z-0.5)(z-0.750)}$, an unbounded input sequence $\{2^k\}$ results in an output which is zero. This introduces difficulties in adaptively controlling the plant.

Comment: The concepts described thus far in this section are important in the control of linear time-invariant systems as well as in the adaptive control of linear systems with unknown parameters. As will become evident in the following sections, similar concepts also play an important role in the nonlinear case and have to be taken into account while designing nonlinear controllers.

C. Some Nonlinear Control Problems

Having discussed some of the basic concepts involved in the analysis and synthesis of linear control systems in the previous section, we proceed to formulate some of the problems that arise in nonlinear dynamical systems.

Let Σ, defined by (1), be the process to be controlled. If the functions f and h are known, the problem belongs to the domain of nonlinear control. If f and h are unknown or partially known, the problem is one of nonlinear adaptive control. Most of the problems in which neural networks are currently finding application, are essentially nonlinear adaptive control problems. Naturally, as in linear adaptive control case, we will be interested first in determining how such systems are to be controlled when f and h in (1) are known and how such methods can be modified for the adaptive case.

In practice, uncertainty in dynamical systems may extend from almost complete knowledge on the one hand to almost complete ignorance on the other. The two classes of problems discussed earlier represent the two extreme cases. Quite often, when the system equations are derived from physical laws (or first principles), the function f and h may be known, but they may contain parameters that are unknown. In the adaptive control literature these are referred to as nonlinear adaptive control problem. Obviously such problems are of great interest and considerable theoretical work has been done on them. While neural networks are generally not used for such problems, nevertheless, they are of interest to us and we consider them briefly in Section IV for purposes of comparison.

Yet another factor that determines the complexity of the problem is whether or not the state of the system is accessible at every instant. If the state vector $x(k)$ can be measured, the problem may be substantially simplified. However, in most practical applications, control has to be effected using only IO measurements, and this in turn requires appropriate IO models to be developed for the system.

From a theoretical point of view, some of the most difficult questions encountered in the control (or adaptive control) of nonlinear systems are related to the stability of the overall system. In this paper, since our interest is in practical control, we shall assume that the system to be controlled is stable and that our primary objective is to improve performance.

The different types of control problems that are encountered in practice may now be classified conveniently as follows:

- nonlinear control problems (f and h known) and nonlinear adaptive control problems (f and h unknown);
- control based on the state vector and control based only on inputs and outputs;
- controller parameters chosen off-line and controller parameters determined on-line;
- time-invariant plants and time-varying plants. (P)

In all the above cases we will be interested in regulation and tracking problems. These problems can be formulated more precisely as follows: Given the dynamical system (1) whose equilibrium state is the origin (i.e., $f(0,0) = 0$), determine a control input u so that:

1) the equilibrium state $x = 0$ is asymptotically stable (stabilization)
2) the output y is maintained at a constant value (set point regulation), i.e., $\lim_{k\to\infty} y(k) = y^*$
3) the output tracks a specific desired signal $y^*(k)$ which is the bounded output of an unforced nonlinear system Σ_ω
4) the output $y(k)$ tracks an arbitrary bounded signal $y^*(k)$ so that $\lim_{k\to\infty} |y(k) - y^*(k)| = 0$.

In all the above cases $x(k) \in \Omega$ where Ω is a bounded neighborhood of the origin. The unforced nonlinear system in 3) is denoted by Σ_ω and is described by

$$\begin{aligned}\Sigma_\omega\colon \omega(k+1) &= S[\omega(k)]\\ y^*(k) &= q[\omega(k)].\end{aligned}$$

The linearization of Σ_ω around the equilibrium state is denoted by $\Sigma_{L\omega}$.

The general tracking problem in 4) can be stated in two different ways. In the first, known as asymptotic tracking, $y(k)$ is required to approach $y^*(k)$ asymptotically. In the second, known as exact tracking, $y(k)$ tracks $y^*(k)$ exactly after a finite number of steps, i.e., $y(k) = y^*(k)$, $k \geq N$.

Items 1)–3) are referred to as problems of regulation and 4) is known as the tracking problem. In addition to the above, we will also be interested in extending the results to multivariable systems as well as to systems which are subject to external disturbances. The robustness of the performance of the system in the latter case is essential if the theory developed is to be applicable to practical control.

The solutions to the above problems for the various cases included in (P) will provide an adequate theoretical framework for discussing practical applications.

In Section IV the control and adaptive control problems are discussed when the state of the system, $x(k)$, is accessible. In Section V the same problems are treated when all decisions have to be based on IO data. Due to space limitations, only the basic theoretical concepts and principal results are presented. For detailed derivations and proofs, the reader is referred to appropriate source papers.

IV. Identification and Control: State Vector Accessible

From the discussion in the previous section it is clear that the prior information concerning Σ to a large extent determines how the control problem is to be formulated, as well as what method should be used for its solution. A convenient starting point for our discussions is the case when all the state variables of Σ are accessible.

Let an SISO dynamical system Σ be defined by (1). We consider the problem of control for the following cases of increasing uncertainty:

1) the functions f and h are known;
2) $f = f(x, u, p)$; $h = h(x, p)$ where f and h are known and p is an unknown parameter vector;
3) the functions f and h are unknown.

In all cases we assume that $x(k)$ is accessible.

A. Case 1: f and h Known

The control of Σ in a neighborhood of the equilibrium state when f and h are known has been studied extensively [10], [11] and the principal results are summarized below.

1) Stabilization: If the linearization Σ_L is stabilizable, it is well known that Σ is also stabilizable by a linear controller. What is more relevant to us is that a smooth function $\gamma(\cdot)$ exists such that $u(k) = \gamma(x(k))$ makes the closed loop system $x(k+1) = f[x(k), \gamma(x(k))]$ finite time stable (i.e., $x(k)$ tends to zero in a finite number of steps). This in turn assures the existence of smooth feedback laws $u(k) = \bar{\gamma}(x(k))$ which make the equilibrium state asymptotically stable.

2) Set Point Regulation: If the linearized system Σ_L satisfies the conditions for set point regulation, then a controller of the form $u(k) = \gamma[x(k), y^*]$ exists, which assures that $\lim_{k\to\infty} y(k) = y^*$ (a constant defined earlier) for the system Σ.

3) Tracking the Output of an Unforced System: If the linearization Σ_L of Σ is stabilizable and satisfies the conditions for the corresponding linear tracking problem, Σ has a well defined relative degree, Σ_ω is a stable reference model, and the linearization $\Sigma_{L\omega}$ of Σ_ω has eigenvalues on the boundary of the unit circle, then a smooth function $\gamma(x(k), y^*(k))$ exists such that $u(k) = \gamma(x(k), y^*(k))$ results in $\lim_{k\to\infty} |y(k) - y^*(k)| = 0$.

4) The General Tracking Problem: Necessary and sufficient conditions for both the asymptotic tracking problem and the finite time tracking problem have been derived [11] under the assumption that Σ is analytic (or f and h are analytic functions). For the asymptotic tracking problem, the control law has the form

$$u^*(k) = \gamma[x(k), y^*(k), y^*(k+1), \cdots, y^*(k+N)]$$

for some constant N, while for the exact tracking problem

$$u^*(k) = \gamma[x(k), y^*(k+d)]$$

where d is the relative degree of Σ. It is shown [11] that if Σ has a well-defined relative degree and the zero dynamics of the system is asymptotically stable, the general tracking problem has a solution. It is further shown that the exact tracking problem and the asymptotic tracking problem are equivalent so that the existence of a solution in one case assures the existence of a solution in the other.

Comment: The importance of the relative degree of a nonlinear system, as well as the asymptotic stability of its zero dynamics, become evident in the general tracking problem. Obviously, these are also relevant in practical applications. Qualitatively, the former requires that the delay from the input to the output be a constant. The latter corresponds to the minimum phase condition in linear time-invariant systems. A detailed treatment of these concepts is beyond the scope of this paper [11]. Even when no uncertainty is present and f and h are known precisely, it is seen from the discussion in this section that explicit solutions cannot, in general, be derived for the regulation and tracking problems. Instead, the conditions under which feedback control laws exist for the various problems were stated. We shall comment on this further in Section IV-C where we consider the case when f and h are unknown.

B. Case 2: $f[x, u, p]$ Known; p Unknown

In the previous case the functions f and h were assumed to be known but very little was assumed about their functional form. Even though extensive research is in progress in the area of nonlinear control, prescriptive methods for control do not exist for such a general class of systems. To obtain analytical solutions, the class of functions $\mathcal{F}$ to which f belongs must be restricted severely. The limiting case, when $\mathcal{F}$ is the class of linear maps is the one that has been extensively investigated. Hence, theoretical interest focusses on those classes of functions for which explicit solutions can be determined. In this context the following question attracted considerable attention:

> Given a dynamical system Σ described by (1), can it be made equivalent to a linear system $z(k+1) = Az(k) + bv(k)$ by 1) a change of coordinates $z = \Phi(x)$ and 2) a feedback law of the form $u(k) = \Psi[x(k), v(k)]$ where $v(k)$ is an external input?

If such transformations exist, the system is said to be feedback linearizable and the tools of linear control theory can be used to stabilize and control the system. Necessary and sufficient conditions for a system to be feedback linearizable were derived in [12] by which the existence of the maps Φ and Ψ is assured. In [10], methods of approximating Φ and Ψ using neural networks are described.

Following the above results in [12], there has been a great deal of interest in the adaptive community in adaptively controlling a class of feedback linearizable systems. The problem is best illustrated by considering the following class of continuous time systems which are feedback linearizable. The class considered represents a subset of the most general class of such systems considered in [13]

$$\begin{aligned}\Sigma\colon \dot{x}_1 &= f_1(x_1, x_2)\\ \dot{x}_2 &= f_2(x_1, x_2, x_3)\\ &\cdots\\ \dot{x}_n &= f_n(x_1, x_2, \cdots, x_n)\\ &\quad + \sum_{i=1}^{l} \alpha_i \gamma_i(x_1, x_2, \cdots, x_n) + \beta_0(x)u.\end{aligned}$$

Let the function $f_i (i = 1, 2, \cdots, n)$, $\gamma_j (j = 1, 2, \cdots, l)$, and $\beta_0(x)$ as well as the parameters α_i $(i = 1, 2, \cdots, l)$ be known. It can be readily shown that the system is feedback linearizable and hence u can be chosen to transform Σ into a linear system. If the parameters α_i are not known, we have a nonlinear adaptive control problem. For such a class of systems a large body of theory exists [14] by which adaptive laws can be derived using Lyapunov functions for global stability. The most important aspect of the result is that if the system is open-loop unstable it can be stabilized globally by the adaptive controller.

The fact that adaptive laws can be derived explicitly to assure global stability makes the above procedures theoretically attractive. It also implies that adaptation can be attempted with confidence. However, from the point of view of applications, the method is less attractive. The specific form of the equations, the fact that all the state variables have to be accessible, and most important of all that the parameters occur linearly, limit the applicability of the method.

Neural networks are not particularly suited to deal with problems of the above type where detailed information concerning the system is available. While stability results can also be derived using neural networks, our interest in this paper is in improving performance of systems that are stable. However, simulation studies on feedback linearizable systems using neural networks (assuming that f and h are unknown) when the system is stable are found to be far more robust than those obtained using the methods described in the adaptive control literature. This leads us to the consideration of systems Σ in which f and h are unknown.

C. Case 3: f and h Unknown

Our interest in this paper is primarily in those problems in which the function f and h are unknown, since many complex control problems belong to this class. Further, these are precisely the class of problems for which well-established techniques of control cannot be directly applied. The first step in the control of such systems is to identify them using the available signals.

The problem of identification consists of setting up a suitably parameterized identification model and adjusting the parameters of the model to optimize a performance criterion based on the error between the outputs of the plant and the model. If neural networks N_f and N_h are used to approximate functions f and h, respectively, the neural network model of the system is

$$\begin{aligned}\hat{x}(k+1) &= N_f[x(k), u(k)]\\ \hat{y}(k) &= N_h[x(k)].\end{aligned}$$

The inputs to N_f and N_h are, respectively, the vectors $[x(k)^T, u(k)]^T$ and $x(k)$ and the corresponding outputs are $\hat{x}(k+1)$ and $\hat{y}(k)$. Since $x(k)$ is accessible, the two networks can be trained independently. The parameters of N_f and N_h are adjusted using static backpropagation or any other equivalent learning algorithm to minimize $\|x(k) - \hat{x}(k)\|^2$ and $|y(k) - \hat{y}(k)|^2$, respectively. We do not consider at this stage the details of the identification procedure such as the choice of the network architectures (number of layers, number of nodes, etc.) or the learning algorithms, but merely assume that the resulting mappings approximate the maps of f and h as closely as desired.

Once N_c and N_f have been trained to approximate the functions f and h, there is essentially very little difference between the two cases discussed in Sections IV-A–C from a practical standpoint, i.e., between the cases when f and h are known and when they are unknown. The practical design of a controller proceeds exactly in the same fashion in the two cases.

The following points are worth noting:

- To identify the nonlinear plant, it must be assumed that the system is bounded-input bounded-output (BIBO) stable. This is a strong assumption and generally implies that a controller has already been designed to stabilize the system. In this case, the purpose of neurocontrol is to improve performance.
- The assumption above permits identification to be carried out either off-line or on-line. If the plant is unstable, identification and control have to be carried out simultaneously on-line and this is what is referred to as adaptive control. Proving the stability of the adaptive system is consequently a very difficult problem.
- In some cases, an approximate linear model of the plant may exist which is adequate when the domain of operation is Ω_L in the state space containing the equilibrium but becomes inadequate when, for operational reasons, the domain is extended to $\Omega \supset \Omega_L$. The neural networks N_f and N_h are then used to augment the linear model in the domain Ω.

1) Stabilization, Regulation, and Tracking: In Section IV-A conditions were given under which control laws can be shown to exist for the various control problems. In all cases, a control law $u(k) = \gamma[x(k), y^*(k)]$ results in $y(k) \to y^*(k)$ as $k \to \infty$. The theoretical existence of such a control law is crucial to the use of neural networks for control since the objective of the latter is to approximate the map γ. This basic philosophy also applies to the more general cases treated in Section V.

Since the system is unknown, the order of the system, the relative degree, the stability of the zero dynamics and the controllability and observability of the linearization Σ_L have to be deduced from the maps N_f and N_h rather than f and h.

In all the control problems described earlier, a neural network $N_\gamma[x(k), y^*(k)]$ is used to approximate the control law. Since the system can become unstable during the training process if N_γ is trained on-line, all training in practical problems is carried out off-line using computer simulations with N_f and N_h. Even in computer simulations, the fact that the controller is in a feedback loop makes parameter adjustments using gradient type algorithms computationally intensive. Hence, approximate methods are invariably used for parametric adaptation.

Many theoretical questions arise because N_f and N_h merely approximate the maps f and h but are not identical to them. Hence, the design of controllers based on N_f and N_h have to be demonstrated to be adequate for the plant.

V. Control Based on Inputs and Outputs

While the state vector representation of dynamical systems is particularly suited for theoretical developments, it is rarely that one has access to the entire state vector as was assumed in the previous section. As a consequence, state vector models of dynamical systems are hard to come by, unless they are derived from physical laws governing the system.

In contrast to the above, the behavior of a system as observed by an external observer is through its inputs and outputs. More relevant for the purposes of this paper, it is in situations of great uncertainty where the modeling of a system has to be carried out using only IO data that neural networks are found to be particularly attractive. Hence IO representation of systems have assumed great importance in neurocontrol problems.

For linear time-invariant systems corresponding to every state representation (2) there is an equivalent IO representation (8) and vice versa. For nonlinear systems the equivalence of the two representations is substantially harder to establish. A basic question in such cases is to decide whether a given IO operator admits a representation in state vector form, and whether a system Σ represented by (1) has an IO representation. The former is the domain of realization theory; the latter is the problem of direct interest to us.

A. The NARMA Model

Given the dynamical system Σ described by (1) we have

$$\begin{aligned} y(k) &= h[x(k)] \stackrel{\text{def}}{=} \Phi_1[x(k)] \\ y(k+1) &= h[f[x(k), u(k)]] \stackrel{\text{def}}{=} \Phi_2[x(k), u(k)] \\ &\cdots \\ y(k+n-1) &= h \cdot f^{n-1}[\cdots] \\ &\stackrel{\text{def}}{=} \Phi_n[x(k), u(k), u(k+1), \cdots, u(k+n-2)] \end{aligned} \tag{12}$$

where $f^{n-1}[\cdots]$ is an $(n-1)$ times iterated composition of f. Using the notation introduced earlier in Section III-B2, $Y_n(k) = \Phi[x(k), U_{n-1}(k)]$, if the Jacobian $\frac{\partial \Phi}{\partial x(k)}|_{(0,0)}$ is nonsingular, it follows from the implicit function theorem that $x(k)$ can be expressed locally in terms of $Y_n(k)$ and $U_{n-1}(k)$. However, it is known (by definition of the state) that $x(k+n)$ depends upon $x(k)$ and the inputs $u(k), u(k+1), \cdots, u(k+n-1)$. This yields the well known local IO representation of a single-input single-output nonlinear dynamical system as [15], [16]

$$\begin{aligned} y(k+1) = \mathcal{F}_1[&y(k), y(k-1), \cdots, y(k-n+1), \\ &u(k), u(k-1), \cdots, u(k-n+1)] \end{aligned} \tag{13}$$

which is called the nonlinear autoregressive moving average (NARMA) representation of the system. As shown in (13), the input $u(k)$ affects the output at time $k+1$ (i.e., the system has relative degree unity). However, if the system has a well defined relative degree d in the domain of interest, the input $u(k)$ at time k, for any initial condition, has an effect on the output at time $k+d$. For such a system, using the same procedure that was adopted for linear systems in Section III, we obtain the IO representation

$$\begin{aligned} y(k+d) = \mathcal{F}[&y(k), y(k-1), \cdots, y(k-n+1), \\ &u(k), u(k-1), \cdots, u(k-n+1)]. \end{aligned} \tag{14}$$

B. Identification

The representation (14) forms the starting point for all the investigations concerning the control of nonlinear dynamical systems using neural networks. Since $\mathcal{F}$ is known to exist, neural networks can be used to approximate it using IO data. In such a case, the identification model has the form

$$\begin{aligned} \hat{y}(k+d) = N_{\mathcal{F}}[&y(k), y(k-1), \cdots, y(k-n+1), \\ &u(k), u(k-1), \cdots, u(k-n+1)] \end{aligned} \tag{15}$$

where $\hat{y}(k)$ is the output of the neural network at time k. The model represented by (15) is generally referred to as the NARMA model of the system. A block diagram representation of the plant and the model when the relative degree is unity is shown in Fig. 3.

1) Extended NARMA Models: The NARMA model was derived using the implicit function theorem under the assumption that Σ_L is observable (the nonsingularity of $\frac{\partial \Phi}{\partial x(k)}$ in (12) is equivalent to the observability of Σ_L). Hence, such a representation is local in nature. Attempts have been made in the past by Aeyels [17], Sontag [18], and more recently by Levin and Narendra [19] to determine globally valid IO models. Aeyels considered the dynamical system $\dot{x} = f(x)$, $y = h(x)$ and showed that almost any such system will be observable if $2n+1$ measurements of the output are taken. Sontag studied the existence of global IO models when f and h in (1) are polynomial functions. In [19] transversality conditions were used to determine conditions similar to those of Aeyels for nonlinear systems of (1). For state invertible systems (for any $u(k)$, f is invertible so that $x(k)$ can be determined from $x(k+1)$), it

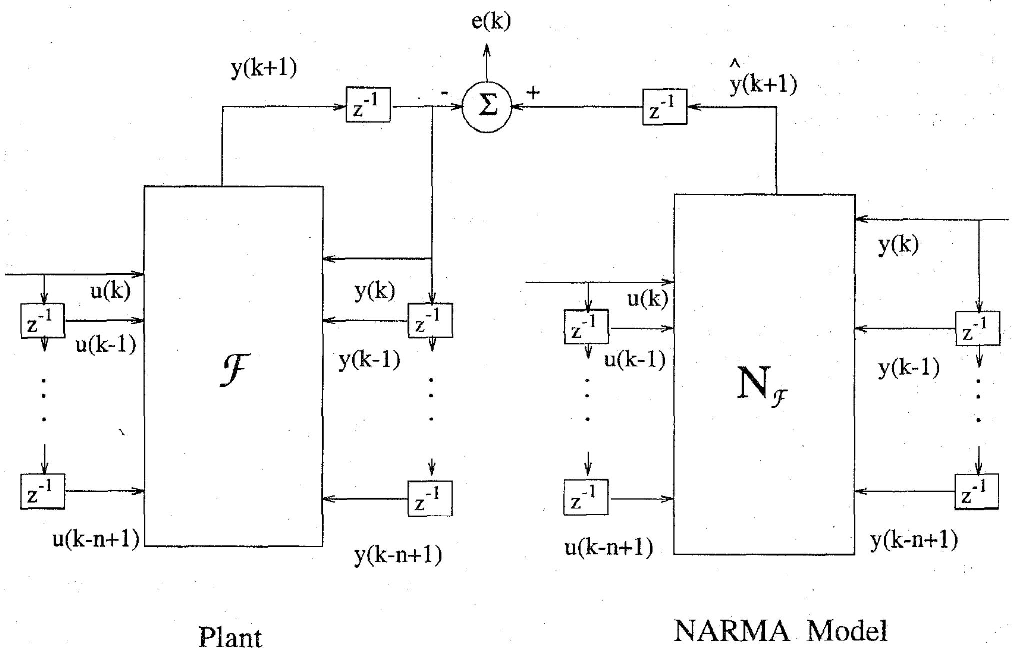


Fig. 3. Identification using IO data.

was shown that $2n+1$ past values of inputs and outputs are adequate generically to determine the output at time $k+1$. This is the extended NARMA model which is generically globally valid.

Comment: Quite often, in practical applications, the order of the system is not known *a priori*. Hence the number of past values needed to identify the nonlinear system using a NARMA model is not known. However, from the previous discussion on the extended NARMA model, it follows that if the system is state invertible, a globally valid model exists generically with a finite number of delays. This provides the rationale for increasing the number of elements in the input vector of the network to achieve better approximation.

C. Control Using IO Representation

In Section IV-A, it was shown that when the state vector $x(k)$ is accessible, exact tracking of a general output $y^*(k)$ could be achieved using a control law

$$u(k) = \gamma[x(k), y^*(k+d)]. \quad (16)$$

While deriving the NARMA model it was shown that, if the system Σ is observable, $x(k)$ can be expressed as a function of $y(k), y(k-1), \cdots, y(k-n+1)$ and $u(k-1), u(k-2), \cdots, u(k-n+1)$. Hence, it follows from (16) that $u(k)$ can also be expressed in terms of its past values $u(k-1), \cdots, u(k-n+1)$, as well as $y(k), \cdots, y(k-n+1)$ and $y^*(k+d)$. If $r^*(k) = y^*(k+d)$, $r^*(k)$ represents the information that is needed at the instant k to implement the control law. In such a case

$$u(k) = \Phi[y(k), y(k-1), \cdots, y(k-n+1), r^*(k), u(k-1), \cdots, u(k-n+1)]. \quad (17)$$

Since the existence of a control law (17) has been established, it can be realized using a neural network N_Φ with $2n$ inputs.

Comment: As stated in the previous section, the neural controller is part of a feedback loop (i.e., recurrent network) and hence the adjustment of its parameters to minimize a performance index is not as straightforward as in the static case and is generally computationally intensive. In view of this, approximate methods are generally used in the optimization procedures. These include the use of static gradients in the place of dynamic gradients and Jacobians derived from the model in the place of those from the plant. This has motivated the development of approximate NARMA models for dynamical systems which simplify the control problem. These are discussed in the following section.

1) Approximate NARMA Models: If, in solving the control problem of a dynamical system, one is inevitably faced with the prospect of making approximations, the question naturally arises as to when such approximations should be made. Is an exact controller for an approximate model better than an approximate controller for an exact model such as the NARMA model? This question is discussed in [20] where two nonlinear approximations are suggested. These are called the NARMA-L1 and NARMA-L2 models and are based on two different Taylor series expansions of the right hand side of (14). Due to space limitations we present here only the models [20].

2) NARMA-L1: The first model is described by

$$y(k+d) = f_0[y(k), \cdots, y(k-n+1)] + \sum_{i=0}^{n-1} g_i[y(k), \cdots, y(k-n+1)]u(k-i)$$

where $f_0 = \mathcal{F}[y(k), \cdots, y(k-n+1), 0, \cdots, 0]$ and

$$g_i = (\partial\mathcal{F})/(\partial u(k-i))|_{y(k),\cdots,y(k-n+1),0,\cdots,0}$$

and is obtained by expanding $\mathcal{F}$ in (14) around $(y(k), \cdots, y(k-n+1), 0, \cdots, 0)$.

3) NARMA-L2: The second model is described by the equation

$$\begin{aligned} y(k+d) = F_1[&y(k), \cdots, y(k-n+1), \\ &u(k-1), \cdots, u(k-n+1)] \\ &+ F_2[y(k), \cdots, y(k-n+1), u(k-1), \\ &\cdots, u(k-n+1)]u(k) \end{aligned}$$

and is obtained by expanding $\mathcal{F}$ in Taylor series around $[y(k), \cdots, y(k-n+1), 0, u(k-1), \cdots, u(k-n+1)]$.

Error bounds for the two models are given in [20] in terms of the maximum matrix norm M_1 of the Hessian matrix of $\mathcal{F}$ with respect to $[u(k), \cdots, u(k-n+1)]^T$ for NARMA-L1 and $(\partial^2\mathcal{F})/(\partial u(k)^2)$ for NARMA-L2 when evaluated over a neighborhood of the equilibrium state.

The great advantage of the two approximate models for the control problem is that, when they are valid, they make the generation of a control input substantially simpler. In fact, $u(k)$ can be computed algebraically in the two cases so that the computational control problem is completely bypassed and the principal problem becomes one of identification.

Comment: The analytical tractability of models proposed and their effectiveness in practical contexts generally determine their acceptance by the scientific community. The NARMA-L1 and NARMA-L2 models have been found to perform at least as well as the NARMA model in numerous simulation studies. In view of this the author believes that they will find a wide following in the neural network community.

Example: A plant is described by the first order equation

$$y(k+1) = \sin(y(k)) + u(k)[5 + \cos(u(k)y(k))].$$

A reference output signal is given as

$$y^*(k) = 2\sin\frac{2\pi k}{50} + 2\sin\frac{2\pi k}{100}.$$

A linear model, the NARMA model, and the NARMA-L1 model (in this case it is the same as the NARMA-L2 model) were used to identify the system and used the corresponding controllers to follow the desired output. The control errors are shown in Fig. 4(a)–(c). While the linear controller performs very poorly, the NARMA model and the NARMA-L1 model result in much better performance. The controller design, as discussed earlier, is substantially more involved if the NARMA model is used.

D. Disturbance Rejection in Control Systems

In all the problems discussed thus far, no external disturbances acted on the system. Further, it was also assumed that the plant parameters were constant so that, at least in theory, the adaptive control system could track the desired signal $y^*(k)$ exactly. However, in practice, external disturbances are invariably present and the system contains dynamics not included in the identification model. More importantly, the *raison d'étre* of adaptive control is to adapt the control input to changes in the system to be controlled. All these imply that the success of adaptive control must be judged in situations where external or internal perturbations are present. A large literature exists on methods for making linear adaptive control systems robust in the presence of bounded external disturbances, unmodeled dynamics, and time-varying parameters. By robustness we mean that small perturbations of the type described above will also result in small deviations in the values of the signals from their nominal values in the unperturbed case. Robustness in linear adaptive control is achieved by suitably modifying the adaptive control laws. In this section we discuss some of the recent attempts made by the author and his colleagues toward extending the results in linear adaptive control to one of the important problems encountered in practice, i.e., the rejection of external disturbances [21], [22].

1) Statement of the Problem: A single-input single-output dynamical system Σ is described by the equations

$$\begin{aligned} \Sigma: x(k+1) &= f[x(k), u(k), v(k)] \\ y(k) &= h[x(k)] \end{aligned} \tag{18}$$

where $x(k) \in \mathbb{R}^n$ is the state of the system, $u(k), y(k) \in \mathbb{R}$ are respectively the input and the output, $v(k)$ is a scalar disturbance which is the output of an unforced stable dynamical system Σ_v described by the equation

$$\begin{aligned} \Sigma_v: x_v(k+1) &= g[x_v(k)] \\ v(k) &= d[x_v(k)] \end{aligned} \tag{19}$$

where $x_v(k) \in \mathbb{R}^m$. The state $x(k)$ of the system Σ, as well as the disturbance $v(k)$, are not accessible. The objective of the control is to track a desired bounded output $y^*(k)$ exactly, even in the presence of the disturbance, by the application of a suitable control $u(k)$.

Since $v(k)$ is the output of an unforced stable system, the value of $v(k+1)$ can be computed if the model of the disturbance (i.e., Σ_v) and $x_v(k)$ are known at instant k. In such a case the appropriate control input $u(k)$ can be determined which will exactly cancel the effect of the disturbance. The procedure described below realizes such a control input asymptotically.

The linearized equations of system (18) and (19) can be written as

$$\begin{aligned} x(k+1) &= Ax(k) + b_1u(k) + b_2v(k) \\ x_v(k+1) &= A_vx_v(k) \\ y(k) &= cx(k) \\ v(k) &= c_vx_v(k) \end{aligned}$$

or alternately as

$$\begin{aligned} x_0(k+1) &= A_ox_o(k) + bu(k) \\ y(k) &= \bar{c}x_o(k) \end{aligned}$$

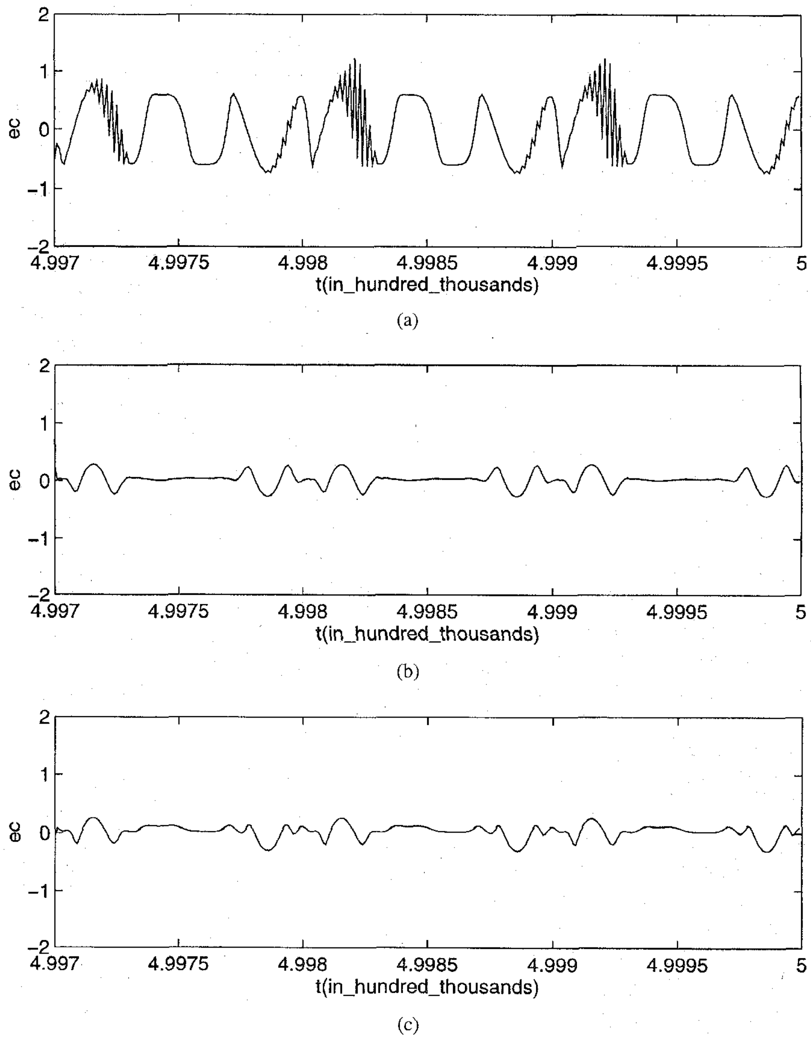


Fig. 4. Control of a first order plant using different IO models: The control errors.

where $x_o(k) = \begin{bmatrix} x(k) \\ x_v(k) \end{bmatrix}$, $A_o = \begin{bmatrix} A & b_2 c_v \\ 0 & A_v \end{bmatrix}$, $b = \begin{bmatrix} b_1 \\ 0 \end{bmatrix}$, and $\bar{c} = [c, 0]$.

If the pairs (c, A) and (c_v, A_v) are observable, it can be shown [21], [22] that the overall system is observable through the output $y(k)$ if the eigenvalues of A_v are different from the zeros of the transfer function from the disturbance to the output, i.e., $c[zI - A]^{-1}b_2$. In such a case the overall linearized system can be represented by an ARMA model of dimension $(n + m)$, i.e.,

$$y(k+1) = \sum_{i=0}^{n+m-1} a_i y(k-i) + \sum_{j=0}^{n+m-1} b_j u(k-j)$$

which is independent of the disturbance. The input $u(k)$ at time k can then be chosen as in the disturbance free case.

The same approach can also be used in the nonlinear problem discussed earlier. Under the same conditions on the linearized systems, it can be shown using the implicit function theorem that a NARMA model of the form

$$\begin{aligned} y(k+1) = \mathcal{F}[&y(k), y(k-1), \cdots, y(k-(m+n)+1), \\ &u(k), \cdots, u(k-(m+n)+1)] \end{aligned}$$

exists relating the input $u(k)$ and the output $y(k)$. If such a NARMA model is realized, the control problem can be solved using the approaches described in the preceding sections.

The following example illustrates that disturbance rejection using the above method can be very effective in practical situations.

2) Example for Disturbance Rejection: A second-order linear time-invariant plant is described by the state equations

$$\begin{aligned} x_1(k+1) &= 0.5x_2(k) \\ x_2(k+1) &= 0.5x_1(k) + 0.1x_2(k) + (u(k) + v(k)) \\ y(k) &= x_1(k) + x_2(k) \end{aligned} \tag{20}$$

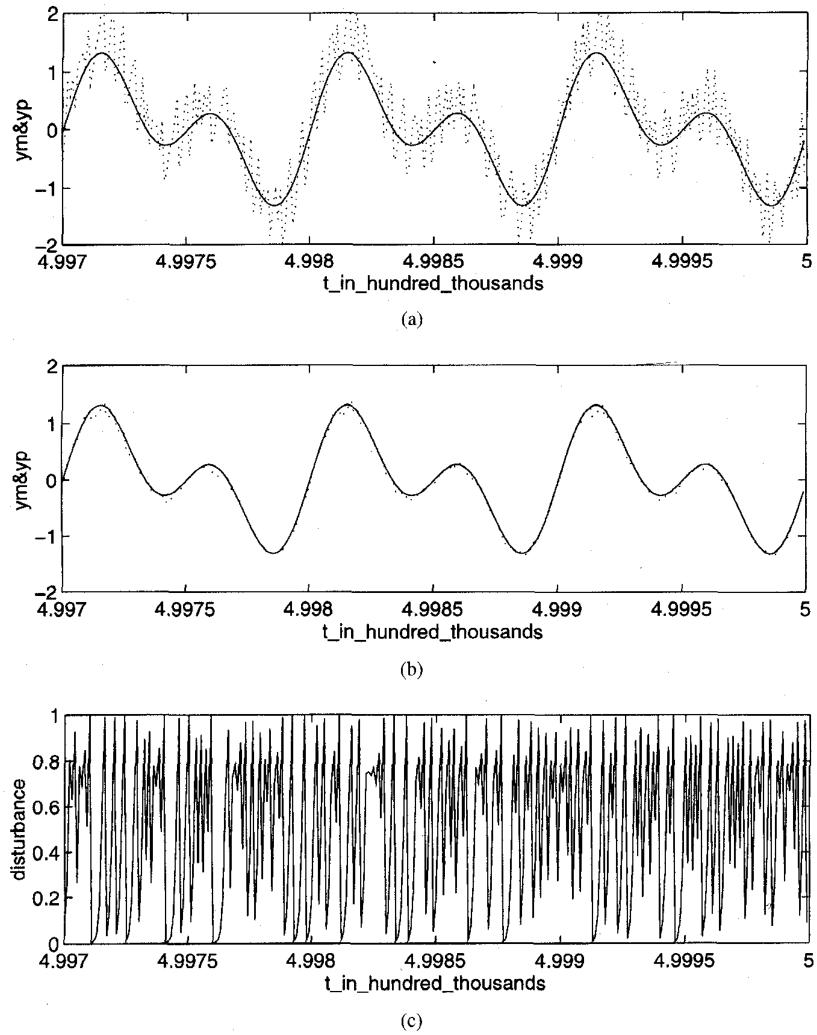


Fig. 5. Disturbance rejection for a linear plant using neural networks.

where the disturbance $v(k)$ is assumed to be generated as a chaotic time-series described by the difference equation $v(k+1) = 4v(k)(1 - v(k))$. The resulting disturbance with initial condition $v(0) \in (0, 1)$ has irregular chaotic behavior [See Fig. 5(c)].

From the theory of linear adaptive control, it is well known that the plant without the disturbance can be effectively controlled by a linear adaptive controller with two delayed values of inputs and outputs. When the disturbance is present, the linear controller results in extremely poor performance even when its dimension is increased by one (since the disturbance satisfies a first order difference equation) [See Fig. 5(a)]. The poor performance can be attributed to the distinctly nonlinear nature of the disturbance dynamics. To account for the nonlinear disturbance, a nonlinear controller was then designed using the method described in this section. This involved the use of a NARMA identification model defined by a neural network N_i, and a controller defined by a neural network N_c as shown below.

$$\text{Identifier: } \hat{y}(k+1) = N_i[y(k), y(k-1), y(k-2), u(k), u(k-1), u(k-2)]$$

$$\text{Controller: } u(k) = N_c[y(k), y(k-1), y(k-2), y^*(k+1), u(k-1), u(k-2)]$$

where $y^*(k)$ is the desired output. The resulting performance [Fig. 5(b)] after training shows dramatic improvement and an almost complete rejection of the disturbance.

E. Nonlinear Multivariable Systems

Since most practical systems have multiple inputs and multiple outputs, the efficacy of neural networks as practical adaptive controllers will eventually be judged by the applicability of the methodology discussed in the previous sections to the multivariable case. Simple heuristic methods which have proved successful for SISO systems cannot be readily extended to MIMO systems and the need for a well developed theory becomes very apparent in such cases. Even when the system is linear with known parameters, the problem of multivariable control is a difficult one due to the coupling that exists between the various inputs and outputs. The representation problem is also not simple, as described in Section III, because of the different delays that can exist between each IO pair. In the 1980's, the adaptive control problem of linear multivariable systems was extensively investigated, and sufficient conditions were derived for the outputs of an MIMO plant with unknown parameters to follow the outputs of a reference model. If the plant is nonlinear and multivariable and has unknown characteristics we have a nonlinear multivariable adaptive control problem. Constructive procedures for designing satisfactory controllers for such problems were practically nonexistent before the advent of neural networks. We present in this section an outline of how such systems can be represented in a convenient fashion and identified using neural networks, and how the identification model can be used for control purposes. It is in the adaptive control of nonlinear multivariable systems that the power of neural networks is particularly evident. As in the SISO problems, the linearization Σ_L of the nonlinear system, along with the implicit function theorem, results in an IO representation.

1) Problem Statement: Let a dynamical system Σ with m inputs and m outputs be described by (1). The two problems which have attracted considerable attraction in the linear case are 1) decoupling and 2) tracking. Our interest here is to solve the same problems with the nonlinear system Σ in some neighborhood of the equilibrium state. If Σ_L is the linearization of Σ around the origin and is defined by (2), it was shown in Section III that Σ_L has an input-state-output representation of the form given in (11). Further, if E is a nonsingular matrix, $u(k)$ can be chosen to be a linear combination of $x(k)$ and the desired output $Y^*(k + d) = R^*(k)$ where the elements of $R^*(k)$ are known at time k.

From (1) it follows, exactly as in the linear case that the output $(y_1(k + d_1), \cdots, y_m(k + d_m)$ can be expressed in terms of the state $x(k)$ and the input $u(k)$ as

$$R(k) = \begin{bmatrix} y_1(k+d_1) \\ \cdots \\ y_m(k+d_m) \end{bmatrix} = \begin{bmatrix} \Phi_1[x(k), u(k)] \\ \Phi_2[x(k), u(k)] \\ \cdots \\ \Phi_m[x(k), u(k)] \end{bmatrix}.$$

If Σ_L is controllable, it also follows that $u(k) = \gamma(x(k), R^*(k))$ exists which can follow a desired output $R^*(k)$ exactly after a finite number of steps. If system (1) is observable, $x(k)$ can be expressed as a nonlinear function of $y(k)$ and its $(n-1)$ past values as well as the $(n-1)$ past values of $u(k)$. This yields the following IO representation for the plant as well as the controller.

$$\begin{aligned}\text{Plant: } R(k) &= \mathcal{F}[y(k), \cdots, y(k-n+1), \\ &\qquad u(k), \cdots, u(k-n+1)] \\ \text{Controller: } u(k) &= \Gamma[y(k), \cdots, y(k-n+1), R^*(k), \\ &\qquad u(k-1), \cdots, u(k-n+1)]\end{aligned}$$

where $R^*(k)$ is the desired value of $R(k)$. Since $\mathcal{F}$ and Γ are known to exist, they can be approximated using neural networks. Further if, as in the case of SISO systems, an approximate model (e.g., the multivariable version of the NARMA-L2 model) is used to identify the plant, it has the general form

$$\begin{aligned} &y_i(k+d_i) \\ &= f_i[y(k), \cdots, y(k-n+1), u(k-1), \cdots, u(k-n+1)] \\ &\quad + \sum_{j=1}^{m} g_{ij}[y(k), \cdots, y(k-n+1), u(k-1), \cdots, \\ &\qquad\qquad u(k-n+1)]u_j(k), \quad i = 1, 2, \cdots, m. \end{aligned} \tag{21}$$

In this case the control input $u(k)$ can be computed algebraically to either decouple the system or follow a reference input exactly. In the latter case

$$\begin{aligned} u(k) = G^{-1}[R^*(k) - f(y(k), \cdots, y(k-n+1), \\ u(k-1), \cdots, u(k-n+1))] \end{aligned}$$

where G is the $(m \times m)$ matrix $\{g_{ij}\}$ whose elements are given in (21). Further, for the state vector $x(k)$ to be bounded, the zero dynamics of the nonlinear system, as in the SISO case, has to be asymptotically stable. For simulation studies on nonlinear multivariable adaptive systems, see [23], [24].

F. Adaptive Control Using Multiple Models

An area of active research at the present time is the control of complex systems which operate in multiple environments. Sudden changes in parameter values, failures of sensors or subsystems, and external disturbances of the form discussed in Section V-D, can be considered as different "environments" control system may be called upon to cope with. In such cases, the need to use multiple models arises naturally, since a different mathematical model may be needed to represent the behavior of the plant in each of the environmants.

1) The Need for Multiple Models: The need for multiple models in the control of dynamic systems may arise for a wide variety of reasons, some of which are given below.

- Many physical systems can be represented by interpolating between local models. The control paradigm known as gain scheduling is based upon this concept.
- Multiple models may be needed to detect different changes in the plant mentioned earlier and initiate the appropriate control action.
- In some cases, all the information concerning the plant, such as the order or the relative degree, may not be available to compute the input. Multiple models may be needed to obtain the appropriate information.
- An important reason for using multiple models is to combine their advantages. One model may assure stability, while another designed heuristically may demonstrate better performance in computer simulations. A proper combination of the two may result in a stable system with improved performance.

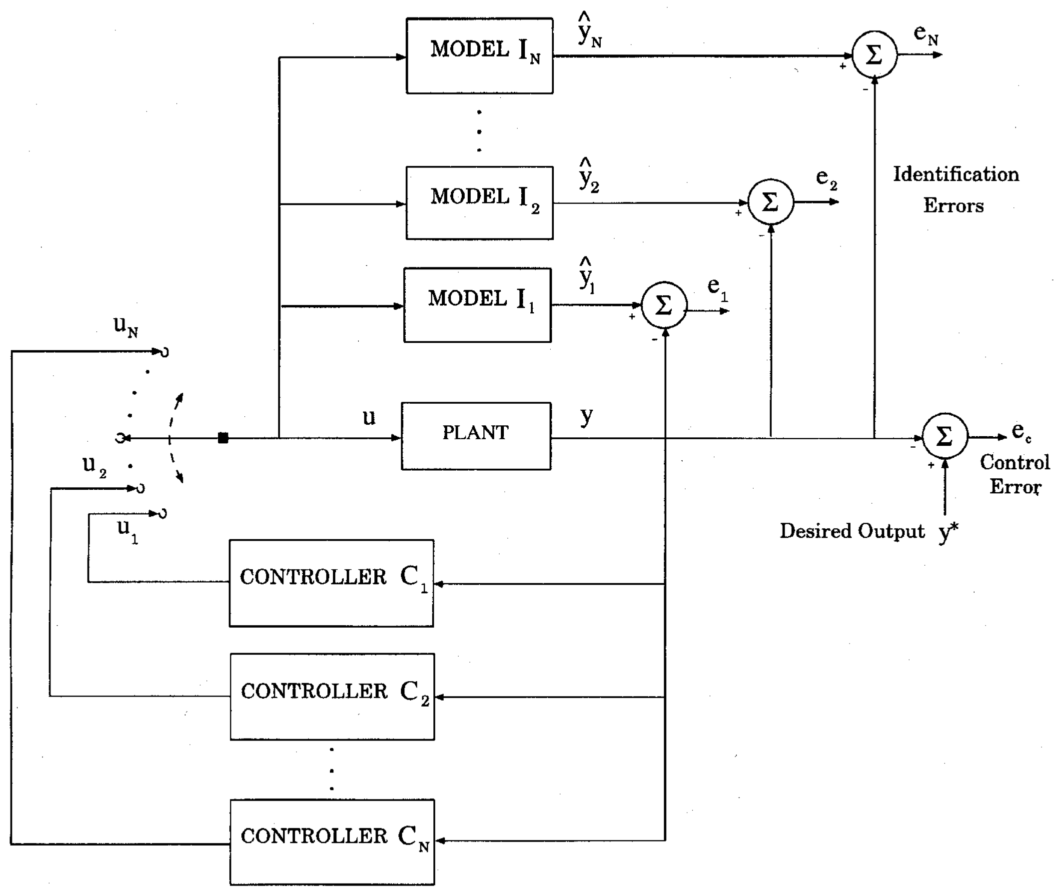


Fig. 6. Structure of the multiple model control with switching and tuning.

If multiple models are used for any of the reasons mentioned above, the aim of the control is to detect which model exists at any instant and service it appropriately. This in turn leads to "switching" and "tuning" techniques for the control of complex systems [24]–[26]. The former is to take rapid measures to avoid catastrophic failures, while the latter corresponds to more leisurely steps taken to improve performance within the new configuration.

2) The Structure of the Controller: The architecture of the switching and tuning system is shown in Fig. 6. $I_1, I_2, \cdots, I_n$ are N predictive models of the plant which have been obtained by the designer after observing the system over a long period of time. $C_1, C_2, \cdots, C_N$ are the corresponding controllers (which have been stored after being obtained using the methods described in the proceeding sections). If the plant output is $y(k)$ and the output of the model I_j is $\hat{y}_j(k)$, the output error is defined as $e_j(k) = \hat{y}_j(k) - y(k)$. Based on a performance index $J(e_j)$, evaluated for $j = 1, 2, \cdots, N$, the model to be used at any instant is chosen. If $J_i(k) = \min_j J(e_j(k))$, the model I_i and the corresponding controller C_i are chosen at instant k. This corresponds to the switching part of the adaptive scheme.

In Fig. 6, the identification models I_i and the controllers C_i can be linear or nonlinear, fixed or adaptive. If the system is nonlinear and neural networks are used for its control, tuning of the controllers is needed for good performance. Hence switching may be considered as necessary for the choice of the initial conditions of the controllers, and tuning for the subsequent adaptive behavior. The following simple example illustrates the improvement in performance that can result by switching and tuning, as compared to a scheme using only tuning.

3) Example: A plant which can operate in three different environments (Environment 1: nominal; Environment 2: sensor failure; Environment 3: nominal with an input disturbance) is described by the state equation

$$\begin{aligned} x_1(k+1) &= .5x_2(k) + u(k) \\ x_2(k+1) &= .5x_1(k) + \frac{.5\sin(x_2(k)+u(k))}{1+.5\sin(x_2(k)+u(k))}. \end{aligned}$$

In Environment 1, the output $y(k) = \alpha_1 \sin(2\pi x_1(k))$, $\alpha_1 \in (-.6, 1.4)$. In Environment 2, $y(k) = \alpha_1 \tanh(2\pi x_1(k))$, $\alpha_1 \in (-.6, 1.4)$. Environment 3 is the same as Environment 1, but u in (22) is replaced by $u + .1\sin\frac{2\pi k}{10}$.

Nominal identifiers and controllers were designed for the three environments using the methods outlined in this section. During the operation of the system, the plant switches randomly between the three environments. Fig. 7 shows the behavior of the overall system for a specific switching sequence:

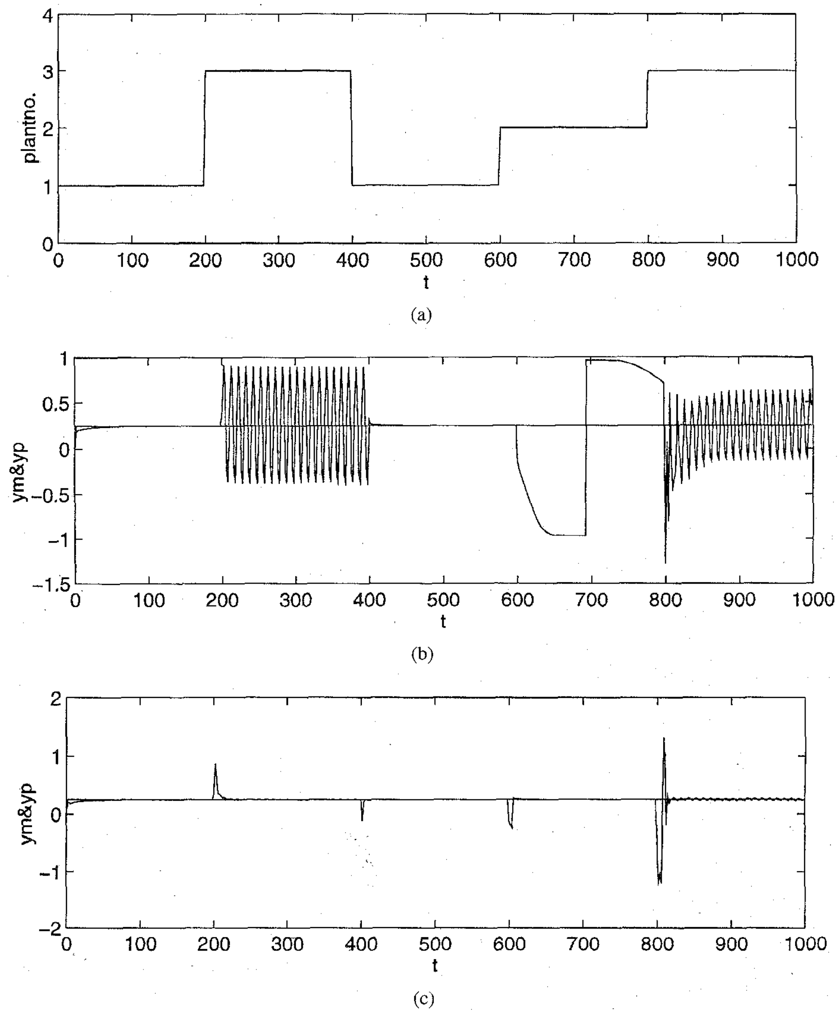


Fig. 7. Switching and tuning for a nonlinear plant.

1) when only tuning is used from the nominal plant [Fig. 7(b)];
2) when both switching and tuning are used (Fig. 7(c)].

The response in the latter case is seen to be far superior to that in the former.

The simplicity of the example belies the large number of theoretical and practical difficulties encountered in the control of nonlinear systems, but it serves to illustrate how the method proposed can be used.

VI. Applications

A recent computer search revealed that during 1990–1995, 9955 articles were published in the engineering literature with titles containing the words "neural network." Among these, over 8000 dealt with function approximation, and pattern recognition, i.e., static systems; 1960 articles were on problems related to control using neural networks, among which only 353 were in the area of applications. Even among these, 45% were related to theory. Of the remaining, approximately 28% described computer simulation studies, 23% were related to experiments in the laboratory, and a mere 4% (or 14 articles) were concerned with real applications (probably, many successful industrial applications do not appear in these lists, for proprietary reasons.) The above statistics indicates that most of the current interest in the application of neural networks is in static systems, particularly in pattern recognition, where they have been very successful. The problem of control is a considerably more difficult one, and the presence of a feedback loop implies that stability problems are invariably present. Even though such stability questions will eventually become very important in industrial applications, the neurocontrol problems being attempted at the present time are those in which improving performance rather than assuring stability is the main consideration.

It is a truism in control practice that the simplest controller which satisfies all the constraints while meeting performance specifications, is the one most likely to be chosen in any application. This is because simplicity generally correlates strongly with robustness as well as with low cost. Neural network based controllers, on the other hand, contain a large number of parameters and are invariably complex. Hence, simpler controllers such as constant gain controllers (PI and PID), state feedback controllers, and linear adaptive controllers must all be tried and found inadequate in some sense before neural controllers are considered seriously in applications.

In the initial stages of any field, when theory is not well developed, it is natural for heuristic techniques to be used in the solution of practical problems. There are numerous examples of such periods in the history of automatic control. Before 1868 automatic control systems were designed through intuition and invention. Efforts to increase the accuracy of the system led to problems of instability and the field had to await Maxwell's theory for an explanation. In the 1950's and 1960's, when the area of adaptive control was in its initial stages, ingenious schemes were proposed with little theoretical justification. History repeated itself in the 1980's, when a number of approximate schemes, based on simple concepts, were proposed in the neurocontrol literature for the output y of a plant to track a desired output y^* with a small error. The basic idea of the above schemes was to determine the inverse of an operator P (which represents the plant) and use it as a controller C, so that the plant together with the controller (i.e., PC) would approximate a unit operator over the range of values of the reference input. The different methods of computing the inverse gave rise to different control architectures, some of which are shown in Fig. 8. In "Direct Inverse Control" [Fig. 8(a)], the input to the controller is y and the controller (a neural network) is trained to approximate the input to the plant. In "Feedforward Inverse Control" [Fig. 8(b)], a neural network NN_2 is trained so that its output $\hat{y}$ approximates the output of the plant. A controller NN_1 is then trained so that $(\mathrm{NN}_2)(\mathrm{NN}_1)$ approximates a unit operator. In internal model control, the error e between model and plant is fed back to the input so that NN_1, NN_2, and P form part of a feedback loop [Fig. 8(c)]. This model has been found to be very successful in practical applications in chemical engineering.

All the above approaches are approximate, since the plant is a dynamic rather than a static map. The latter raises many mathematical questions such as existence of solutions, left versus right inverses of operators, as well as those related to the domains and range spaces of the operators being approximated. This accounts for the fact that we have not discussed them in earlier sections. Nevertheless, we mention them here because successful heuristic techniques are the precursors to good theory and the latter is quite often merely a precise way of stating intuitive concepts.

The problem of nonlinear control is very complex and even at the present time a great deal of heuristic reasoning is used in applications. Simple schemes, such as those discussed earlier, have yielded impressive results in simulation studies as well as in some applications with long time constants. But they do not offer guarantees about their applicability under very different operating conditions. Theoretical control methods become increasingly relevant to provide such guarantees. This is not to imply that such

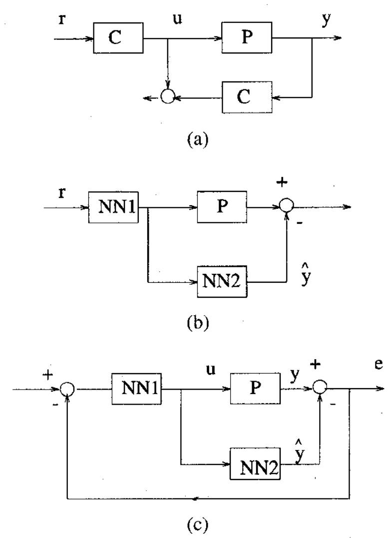


Fig. 8. Control architectures.

methods can be directly applied to practical problems. Practical control system design invariably involves compromises and approximations, but a sound theory provides a sound basis for such engineering choices.

In this section we present a few applications selected from the large number mentioned earlier to provide the reader with some idea of the status of neurocontrol in practical applications. These include computer simulation studies, laboratory experiments, and real applications and range from the control of fairly simple electromechanical systems to complex ill-defined processes. They indicate the wide variety of situations where neural networks are proving successful and also provide a sense of the difficulties involved.

A. Robotics

Neural networks have been used extensively in robotics and the areas where neurocontrollers have found application can be conveniently classified as: 1) manipulator control, 2) contact control, 3) coordination, grasping, and manipulation, 4) locomotion, and 5) planning and navigation.

In manipulator control, the use of neural networks has been strongly motivated by the enormous complexity of computing kinematics in real time. Since adaptive laws with kinematics terms computed based on desired rather than actual trajectories had been shown to be stable, neural networks could be trained off-line to learn the dynamics of the robot.

This has been the main approach used in this area starting with the work of Kawato [27] to the recent work of Zomoya and Nabhan [28]. In such work, the authors have shown that satisfactory tracking can be achieved for unknown payloads and that neural networks can adapt rapidly to changing payloads on-line. With few exceptions [29]–[31], most work on robot control has been in the form of simulation studies where dynamic model parameter values have been obtained from experimental studies.

Learning impedance relationships for the purpose of transferring human skills to robots in tool hanling, and learning environmental models for overall system stability and performance are two areas where some effort has been made to apply neural networks to contact control. Asada [32] has applied neural networks for learning skill from real force and motion data. Learning of a skill in many cases implies the learning of the combined impedance of the human and the tool, and this is accomplished using a neural network. Transferring the skill to a robot implies that the robot must be able to change its impedance through control to match what was learned in the neural network. Using recurrent networks, Venkataraman [33] has developed a robotic system that has the ability to autonomously acquire models of soil and rock samples found on planetary surfaces.

In view of the inherent complexity of coordination, grasping, and manipulation, neural networks have not been used to any appreciable extent in this area. A notable exception is Hwang [34] who has considered the problem of cooperative control of two robots that grasp and carry an object. Two neural networks (one for each arm) are trained with data from the system. Another problem area where neural networks have been used effectively is due to Hanes [35] and is related to control of contact forces between the fingers of a robot hand and the object that it grasps. The problem of determining grasp locations that will result in a desired force distribution is complex from the computational viewpoint and neural networks are ideally suited for it. The computational efficiency of the latter is demonstrated by the authors by comparing the approach based on neural networks with a linear programming approach to solve the problem. Neural networks were found to yield stable grasp configurations over a wide range of object sizes and clinch levels.

In the area of locomotion, Miller [36] has been systematically using CMAC networks to control biped walking robots. In recent years he and his co-workers have designed, implemented, and tested an adaptive dynamic balance scheme on such an experimental robot. According to them, while the control problems for dynamic walking are more complicated than with static balance, dynamic walking promises higher walking speeds, improved walking structures, and greater efficiency.

The application of neural networks in planning and navigation is considered in the following subsection.

B. Autonomous Navigation

A truly interesting and novel application of neural adaptive control is described by Pomerleau in [37] and is the area of vision-based autonomous driving. Based on images from an on-board video camera, a robot van, equipped with motors for the steering wheel, braking, and the acceleration pedal, determines its own trajectory. The most noteworthy feature of the system is the supervisory control method used to train it. The neural network is taught to imitate the driving reactions of a person. As the person drives, the neural network is trained using back propagation. The input to the neural network is a 30×32 unit video image and the output layer consists of 30 nodes each of which corresponds to a steering direction.

To facilitate generalizations to new situations, shifting and rotating the camera image in software is resorted to. This permits the network to learn the desired behavior without requiring the human trainer to stray away from the center of the road. A van trained in the above fashion was able to travel at 55 mi/h under different lighting and weather conditions, which, according to the author, is five times as fast as any nonconnectionist system has done using comparable hardware.

Another application which clearly borrows from the work of Pomerleau is a laboratory experiment which attempts to drive a car into a U-shaped parking space from an arbitrary starting position in front of the U [38]. As in [37], an expert driver is used to teach the vehicle the steering commands to be used in different situations. A multilayer perceptron (MLP) maps a video image into steering commands. The initial image with 621×286 pixels is processed by an edge detector and fed into the MLP as an input. During the training process, the expert driver drives at constant velocity into the U-shaped space. Thirteen training trajectories were used to train the MLP.

A very exciting application in a somewhat different direction is by Beer and his colleagues who believe that neural network architectures abstracted from biological systems can be directly applied to the control of autonomous agents [39]. Since they have evolved over a long periods of time, even simple animals are capable of feats of sensorimotor control that are far superior to the most sophisticated robots. In [40], a fully distributed neural network architecture for controlling the locomotion of a hexaped robot is described. It is shown that the controller can be utilized to direct the robust locomotion of an actual six-legged robot to achieve a wide range of gaits.

C. Neural Networks in Aeronautics

Another area where neural networks are finding application is aeronautics. For an excellent review of the problems that arise in such systems and the manner in which neural networks may be used, see [41].

Driven by cost and operational requirements operating envelopes of aircraft are being extended toward regimes governed by unsteady fluid mechanics. These, in turn, are bringing in their wake complex stability and control problems. In rotor craft, high intermittent blade torque causes rotor fatigue requiring replacement of critical components. Interactions between rotor blades and blade tip vortex can cause vibrations of the retreating rotor blade. Under such operating conditions, to maintain safety and operational readiness, new types of control systems are needed. Neural network technologies are being explored for fault diagnosis, control reconfiguration, identification of nonlinear dynamics and adaptive control. The methods described in Section IV for the identification and control of nonlinear multivariable systems are obviously relevant for these problems.

The use of the theory and technology of artificial networks for problems of identification, diagnosis, and control in large complex space systems is proposed by Rauch and Schaecter [42]. Improvement of performance without interrupting the control loop, evaluation of the output sensors to determine the existence of spurious structural vibrations and implementation of health monitoring which allows the system to recognize immediate faults as well as long term degradation, are some of the problems considered in this article.

The idea of using stored information to react rapidly to changing environments was proposed in 1992 by Narendra and Mukhopadhyay [42]. This eventually evolved to the general methodology of using multiple models for adaptive control using neural networks described in Section V-F. This idea is being increasingly used in aircraft control problems. Models representing specific failure modes are described by Rauch. At Lockheed, a failure detection and isolation procedure has been implemented on a six degree-of-freedom computer simulation of the F/A-18 fighter aircraft to detect control surface faults [44]. In [44], a similar scheme is suggested for determining the reference models of an aircraft system. The latter is provided with both the anticipated vehicle dynamics and the expected sensor measurements, and the error between predicted and measured values could be used for fault detection and adaptive control. In a recent paper [45], the ideas of multiple models and switching are applied to the adaptive control of a high speed ship. Control and actuator laws are developed as a function of sea state and vessel heading and validated during sea trials. Later, during operation, the adaptive controller senses the actual sea state and heading and chooses the appropriate control law.

Another interesting feature of aeronautical systems described in [41] is the possibility of using thousands of sensors and actuators based on microelectronic mechanical systems. Faller and Schreck [41] and Rauch and Schaechter [42] believe that one of the major advantages of neural network is the tremendous computational speed achieved by massively parallel hardware.

D. Fed-Batch Fermentation Processes

Biofermentation processes, in which microorganizms grown on a suitable medium (substrate) synthesize a desired substance, are widely used in the production of a large number of useful products including amino acids, antibiotics, fuels, and various foods and beverages. Recent advances in genetic engineering have increased the importance of such processes. In fed-batch fermentation processes, the process starts with some initial volume, concentration of microorganisms and substrate, and as the process evolves the substrate is continuously fed into the fermenter. The feeding rate of the substrate and the influent substrate concentration are the control variables. In qualitative terms the objective of control is to maximize the output of micro-organizms while keeping other outputs with a deleterious effect (such as ethanol or acetic acid) at low values. The state variables and the control inputs as well as the state equations for a typical process are shown in Fig. 9.

The control objective corresponding to the optimization objective stated earlier is to force $x_1(t)$ to follow a desired profile $x_1^*(t)$ even while regulating $x_2(t)$ around a constant value.

The control of the above process is an important industrial problem. It is typical of complex processes where a small improvement in performance results in substantial economic benefits. The control process is distinctly nonlinear, and its dynamics is not well known. Changes in initial conditions, parameter variations, input saturation, external disturbances, and unmodeled dynamics are all encountered in practice, making the control problem very difficult.

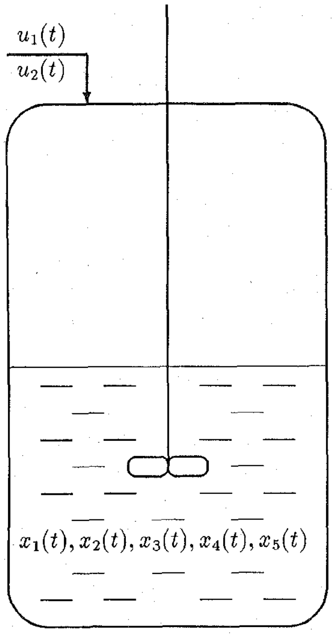
The adaptive control of such a nonlinear time-varying plant is discussed in detail in [46], and using simulation studies, the performance of linear, linear-adaptive, nonlinear, and nonlinear-adaptive control methods are compared. It is concluded in [46] that when the nonlinearities are not known accurately, linear adaptive and nonlinear adaptive controllers using neural networks are the only two viable alternatives. Even among these two, if accuracy and robustness are critical issues, neural networks are distinctly preferable.

E. Neural Control of Steel Rolling Mill

The problem of controlling the strip thickness of a steel rolling mill in simulation studies was reported by Sbarbaro–Hofer *et al.* [51]. The complex nonlinear nature of the problem as well as the varying delays, make the use of neural networks attractive in this case. The authors used predictive control of the type described in Section IV, as well as internal model control described earlier in this section, to design neural networks. The authors attribute the greater accuracy of neurocontrollers as compared to conventional PI and PID controllers to the markedly different behavior of the plant in different operating regions. Since neural networks require more computational effort than conventional controllers, an attempt was made to combine the two types of controllers. The best results from an engineering point of view were obtained by using an approximate inverse of the plant in parallel with a simple integral controller. According to the authors, the viability of the neural network scheme was established by the simulation studies. Additional technical aspects such as memory and computational requirements should be studied before attempting hardware implementation.

F. Arc Furnace

One of the most impressive industrial applications of neurocontrol encountered by the author is the Intelligent Arc Furnace designed by Neural Applications Corporation in Coralville, IA. Electric arc furnaces use electric power to melt steel. A typical furnace is 15–30 ft in diameter and melts 55–150 tons of steel in about an hour. The amount of power delivered to the scrap metal is controlled by positioning three large electrodes each 12–24 in. in



u_1 - substrate feeding rate.

u_2 - influent concentration.

$$\dot{x}_1 = \mu_m \frac{x_1 x_2}{k_s + x_2} \cdot \frac{1}{1 + x_3^2} - \frac{x_1}{x_5} u_1 + + k_{xe} x_1 \xi_1(x),$$
(x_1 - Cell Concentration)

$$\dot{x}_2 = -\frac{\mu_m}{k_y} \frac{x_1 x_2}{k_s + x_2} \cdot \frac{1}{1 + x_3^2} + \frac{u_2 - x_2}{x_5} u_1 - -k_m x_1 + \frac{1}{k_{es}} x_1 \xi_2(x),$$
(x_2 - Substrate Concentration)

$$\dot{x}_3 = k_1 x_1 + k_2 \mu_m \frac{x_1 x_2}{k_s + x_2} \cdot \frac{1}{1 + x_3^2} - \frac{x_3}{x_5} u_1,$$
(x_3 - Inhibitory Substance Concentration)

$$\dot{x}_4 = x_1 \{\xi_2(x) - \xi_1(x)\} - \frac{x_4}{x_5} u_1,$$
(x_4 - By-product Concentration)

$$\dot{x}_5 = u_1. \quad (x_5 \text{ - Fermenter Volume})$$

Fig. 9. Control of a fed-batch fermentation process.

diameter. Historically, electric arc furnaces have consisted of three independent controllers, one for each electrode. The controllers move the electrodes up and down for set point regulation around a constant value of the current. However, the system is complex and nonlinear in which the interaction between the three inputs has to be taken into account to optimize performance. Mathematical representation of the entire multivariable system is practically impossible. This makes the problem ideally suited for control using ANN's. The Neural Applications Corporation has developed neural network controllers by continuously updating a multivariable model of the system [47]. The controller can also be trained to reduce oscillations in the current, and to minimize line voltage fluctuations called "flicker."

Apart form being complex and nonlinear, the system is also time-varying. Furnace performance is a function of a variety of factors such as electrode mass, power characteristics, and scrap content so that set points cannot be predetermined but have to be predicted on-line, making it a tracking problem. Neural network based systems have been developed for these and related control problems, which are substantially improving overall furnace operation.

VII. Comments and Conclusion

Applications in new technologies such as robotics, manufacturing, space technology, and medical instrumentation, as well as those in older technologies such as process control and aircraft control, are creating a wide spectrum of control problems in which nonlinearities, uncertainties, and complexity play a major role. For the solution of many of these problems techniques based on ANN's are beginning to complement conventional control techniques, and in some cases they are emerging as the only viable alternatives. From a control theoretic point of view, ANN's may be considered as tractable parametrized families of nonlinear maps. As such they have found wide application in pattern recognition problems, which require nonlinear decision surfaces. With the introduction of dynamics and feedback, the scope of such networks as identifiers and controllers in nonlinear dynamical systems has increased significantly.

The theoretical principles for the design of identifiers and controllers using neural networks proposed in this paper are based on well established results in linear adaptive control and nonlinear control. Using the implicit function theorem, the properties of the linearized system around the equilibrium state can be extended to the nonlinear domain. This in turn permits most of the results derived in linear control theory such as disturbance rejection, decoupling of multivariable systems, and adaptation using multiple models to be extended to nonlinear dynamical systems.

Extensive simulation studies carried out in many fields have been remarkably successful, revealing that neural networks have great potential in practical control problems. The dynamical systems in these studies are distinctly nonlinear, for which standard linear methods are not applicable. The most impressive feature from a practical standpoint that has become evident from these investigations is the remarkable ability of neural networks to identify and control complex multi-input multi-output nonlinear systems using only IO data.

In both identifiers and controllers, the parameters are adjusted using back propagation or equivalent gradient based methods. While static back propagation is adequate for identification, the adjustment of controller parameters in the feedback loop requires dynamic gradient methods which are considerably more complex and computationally intensive [48]–[50]. This in turn has led to the development of approximate identification models which simplify the control problem substantially. It is the author's opinion that such approximate models will find wide application in the future in practical control problems.

Even as studies in academia and industry have proceeded, a better understanding of the strength and limitations of neurocontrollers is beginning to emerge. Successes of simulation studies on synthetic problems are encouraging researchers to shift to real problems and exploit the vast amounts of experimental data that have been collected in industries and research laboratories. This, in turn, is making the latter more receptive to ideas of using neural networks for real time identification and control in practical systems. In many such applications neural networks will be used to augment, rather than replace, linear identifiers and controllers that are already in place.

One of the most attractive features of neural networks is their massive parallelism. However, most of the simulation studies are carried out at present on serial computers. When VLSI implementation of neural networks becomes common, the full potential of such networks can be exploited to realize computationally fast solutions. This is bound to have a major impact on neurocontrol technology.

Practical systems design is driven by both cost and operational requirements. Anyone familiar with industrial problems is only too aware that bridging the gap between theoretical principles on the one hand, and development, testing, and implementation on the other, is a slow process. However, on the basis of the advances made thus far and the great successes that have already been achieved in some cases, there is every reason to believe that neural network based control systems will be developed in many industries in the next decade to address a wide range of complex control problems.

ACKNOWLEDGMENT

The author would like to thank all the graduate students and colleagues who collaborated with him on different aspects of the research reported here. He gratefully acknowledges the help provided by J. Boskovic and the critical and insightful comments of S. T. Venkataraman. His special thanks go to S. Mukhopadhyay for excellent suggestions and inspiring discussions and L. Chen for his valuable and patient assistance throughout the preparation of this paper.

REFERENCES

[1] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neur. Networks*, vol. 1, pp. 4–27, Mar. 1990.
[2] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neur. Networks*, vol. 2, pp. 183–192, 1989.
[3] G. Cybenko, "Approximation by superposition of a sigmoidal function," *Mathematics Contr., Signals, and Syst.*, vol. 2, no. 4, pp. 303–314, 1989.
[4] K. Hornik, M. Stinchcombe, and H. White, "Multi-layer feedforward networks are universal approximators," *Neur. Networks*, vol. 2, pp. 359–366, 1989.
[5] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares algorithm for radial basis function networks," *IEEE Trans. Neur. Networks*, vol. 2, pp. 302–309, Mar. 1991.
[6] A. R. Barron, "Neural net approximation," in *Proc. 7th Yale Workshop on Adaptive and Learning Syst.*, 1992, pp. 68–72; "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Inform. Theory*, vol. 39, pp. 930–945, Mar. 1993.
[7] W. Rudin, *Principles of Mathematical Analysis*. New York: McGraw-Hill, 1976.
[8] C. T. Chen, *Introduction to Linear System Theory*. New York: Holt, Rinehart, and Winston, 1970.
[9] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1980.
[10] A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks: Controllability and stabilization," *IEEE Trans. Neur. Networks*, Mar. 1993.
[11] J. B. D. Cabrera and K. S. Narendra, "On regulation and tracking in nonlinear discrete-time systems—Part I: State variables accessible; Part II: Nonlinear ARMA models, Tech. Rep. No. 9515 and 9602, Center for Syst. Sci., Yale Univ.
[12] B. Jakubczyk, "Feedback linearization of discrete-time systems," *Syst. Contr. Lett.*, vol. 9, pp. 411–416, 1987.
[13] D. Seto, A. M. Annaswamy, and J. Baillieul, "Adaptive control of nonlinear systems with a triangular structure," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1411–1428, July 1994.
[14] M. Krstić, I. Kanellakopoulos, and P. V. Kokotović, *Nonlinear and Adaptive Control Design*. New York: Wiley, 1995.
[15] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for nonlinear systems, Part I: Deterministic nonlinear systems," *Int. J. Contr.*, vol. 41, no. 2, pp. 303–328, 1985.
[16] A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks: Part II—Identification," Tech. Rep. no. 9116, Center for Syst. Sci., Yale Univ.
[17] D. Aeyels, "Generic observability of differentiable systems," *SIAM J. Contr. and Opt.*, vol. 19, pp. 595–603, 1981.
[18] E. D. Sontag, "On the observability of polynomial systems," *SIAM J. Contr. and Opt.*, vol. 17, pp. 139–151, 1979.
[19] A. U. Levin and K. S. Narendra, "Recursive identification using feedforward neural networks," *Int. J. Contr.*, vol. 61, pp. 533–547, 1995.
[20] K. S. Narendra and S. Mukhopadhyay, "Adaptive control using neural networks and approximate models," Tech. Rep. No. 9603, Neuroengineering and Neurosci. Ctr., Yale Univ., Jan. 1996.
[21] S. Mukhopadhyay and K. S. Narendra, "Disturbance rejection in nonlinear systems using neural networks," *IEEE Trans. Neur. Networks*, pp. 63–72, Jan. 1993.
[22] K. P. Krishna, "Disturbance rejection in nonlinear dynamical systems," Ph.D. dissertation, Yale Univ., 1996.
[23] K. S. Narendra and S. Mukhopadhyay, "Adaptive control of nonlinear multivariable systems using neural networks," *Neur. Networks*, vol. 7, no. 5, pp. 737–752, 1994.
[24] ——, "Intelligent control using neural networks," in *Intelligent Control Systems*, M. M. Gupta and N. K. Sinha, Eds. Piscataway, NJ: IEEE Press, 1995, pp. 151–186.
[25] K. S. Narendra, J. Balakrishnan, and K. Cılız, "Adaptation and learning using multiple models, switching and tuning," *IEEE Contr. Syst. Mag.*, pp. 37–51, June 1995.

[26] K. S. Narendra and J. Balakrishnan, "Improving transient response of adaptive control systems using multiple models and switching, *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1861–1866, Sept. 1994; "Adaptive control using multiple models," *IEEE Trans. Automat. Contr.*, 1996.
[27] M. Kawato, "Computational schemes and neural network models for formation and control of multi-joint arm trajectory," *Neural Networks for Control*, T. Miller, R. Sutton, and P. Werbos, Eds. Cambridge, MA: MIT Press, 1990, pp. 197–228.
[28] A. Y. Zomoya and T. M. Nabhan, "Centralized and decentralized neuro-adaptive controllers," *Neur. Networks*, vol. 6, no. 2, pp. 223–244, 1993.
[29] M. B. Leahy, M. A. Johnson, and S. K. Rogers, "Neural network payload estimation for adaptive robot control," *IEEE Trans. Neur. Networks*, vol. 2, pp. 93–100, Jan. 1991.
[30] M. Kuperstien, "Neural network model for adaptive hand-eye coordination," *Sci.*, vol. 239, pp. 1308–1311, 1988.
[31] D. A. Handelman, S. H. Lane, and J. J. Gelfand, "Integrating neural networks and knowledge-based systems for intelligent robotic control," *IEEE Contr. Syst. Mag.*, vol. 10, no. 3, pp. 77–87, 1990.
[32] S. Liu and H. Asada, "Transferring manipulative skills to robots—Representation and acquisition of tool manipulative skills using a process dynamics model," *ASME J. Dynamic Syst., Measure. and Contr.*, vol. 114, no. 2, pp. 220–228, 1992.
[33] S. T. Venkataraman, S. Gulati, J. Barhen, and N. Toomarian, "Compliance control with neuromorphic identification of environmental dynamics," in *Proc. IEEE Conf. on Decision and Contr.*, Tucson, AZ, 1992, pp. 3475–3480.
[34] Y. Y. Hwang and I. Todo, "Cooperative control of 2 direct-drive robots using neural networks (grasping and movement of an object)," *JSMEC*, vol. 37, no. 2, pp. 335–341, 1994.
[35] M. K. Hanes, S. C. Ahalt, K. Mirza, and D. E. Orin "Power grasp force distribution control using neural networks," *J. Robotic Syst.*, vol. 9, no. 5, pp. 635–661, 1992.
[36] W. T. Miller, "Real-time neural network control of a biped walking robot," *IEEE Contr. Syst. Mag.*, vol. 14, no. 1, pp. 41–48, 1994.
[37] D. A. Pomerleau, "Efficient training of artificial neural networks for autonomous navigation," *Neur. Computation*, vol. 3, pp. 88–97, 1991.
[38] W. A. Daxwanger and G. K. Schmidt, *Skill-based Visual Parking Control Using Neural and Fuzzy Networks.*
[39] R. D. Beer, *Intelligence as Adaptive Behavior: An Experiment in Computational Neuroethology.* San Diego: Academic, 1990.
[40] R. D. Beer *et al.*, "A distributed neural network architecture for hexaped robot locomotion," *Neur. Computation*, vol. 4, pp. 356–365, 1992.
[41] W. E. Faller and S. J. Schreck, "Neural networks: Applications and opportunities in aeronautics," *Progress in Aerospace Sciences*, 1996.
[42] H. E. Rauch and D. B. Schaecter, "Neural networks for control, identification, and diagnosis," in *Proc. World Space Congress*, 1992, pp. F4.4–M1.06.
[43] K. S. Narendra and S. Mukhopadhyay, "Intelligent control using neural networks," *IEEE Contr. Syst. Mag.*, pp. 11–18, Apr. 1992.
[44] H. E. Rauch, "Intelligent fault diagnosis and control reconfiguration," *IEEE Contr. Syst. Mag.*, June 1994.
[45] H. E. Rauch, T. Schmidt, and B. Natoni, "Adaptive control of a high-speed ship," in *Proc. 9th Yale Workshop on Adaptive and Learning Syst.*, 1996.
[46] J. D. Boskovic and K. S. Narendra, "Comparison of linear, nonlinear, and neural network based adaptive controllers for a class of fed-batch fermentation processes," *Automatica*, vol. 31, no. 6, pp. 817–840, 1995.
[47] W. E. Staib and R. B. Staib, "The intelligent arc furnace^{TM} controller: A neural network electrode position optimization system for the electric arc furnace," in *Proc. Int. Joint Conf. on Neur. Networks*, 1992, pp. III1–III9.
[48] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, pp. 1550–1560, Oct. 1990.
[49] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neur. Computation*, vol. 1, no. 2, pp. 270–280, 1989.
[50] K. S. Narendra and K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Trans. Neur. Networks*, vol. 2, pp. 252–262, Mar. 1991.
[51] D. Sbarbaro–Hofer, D. Neumerkel, and K. J. Hunt, "Neurocontrol of a steel rolling mill," *IEEE Contr. Syst. Mag.*, vol. 13, no. 3, pp. 69–75
[52] K. S. Narendra and S.-M. Li, "Neural networks in control systems," in *Mathematical Perspectives on Neural Networks*, Smolensky, Mozer, and Rumelhard, Eds. Hillsdale, NJ: L. Erlbaum Assoc.

Kumpati S. Narendra (Fellow, IEEE) received the Ph.D. degree from Harvard University, Cambridge, MA, in 1959, and the honorary D.Sc. degree from Anna University, Madras, India, in 1995.

At present, he is Professor of Electrical Engineering and Director of the Center for Systems Science at Yale University. He was also appointed Distinguished Visiting Scientist by the Jet Propulsion Laboratory during 1994–1995. He is the co-author of three books, *Frequency Domain Criteria for Absolute Stability* (with J. H. Taylor), *Stable Adaptive Systems* (with A. M. Annaswamy), and *Learning Automata—An Introduction* (with M. A. L. Thathachar). He is also the editor of four books. His research interests include stability theory, adaptive systems, learning automata, and the control of complex systems using neural networks.

Dr. Narendra is a member of Sigma Xi and a Fellow of the American Association for the Advancement of Science. He received the 1972 Franklin V. Taylor Award of the IEEE Systems, Man, and Cybernetics Society, the George S. Axelby Best Paper Award of the IEEE Control Systems Society in 1988, the Education Award of the American Automatic Control Council in 1990, and the Outstanding Paper Award of the Neural Networks Council in 1991. In 1994, he received the Neural Networks Leadership Award of the International Neural Networks Society. He received the Bode Prize of the Control Systems Society in 1995.