Identification and Control of Dynamical Systems Using Neural Networks

KUMPATI S. NARENDRA FELLOW, IEEE, AND KANNAN PARTHASARATHY

Abstract—The paper demonstrates that neural networks can be used effectively for the identification and control of nonlinear dynamical systems. The emphasis of the paper is on models for both identification and control. Static and dynamic back-propagation methods for the adjustment of parameters are discussed. In the models that are introduced, multilayer and recurrent networks are interconnected in novel configurations and hence there is a real need to study them in a unified fashion. Simulation results reveal that the identification and adaptive control schemes suggested are practically feasible. Basic concepts and definitions are introduced throughout the paper, and theoretical questions which have to be addressed are also described.

I. INTRODUCTION

MATHEMATICAL systems theory, which has in the past five decades evolved into a powerful scientific discipline of wide applicability, deals with the analysis and synthesis of dynamical systems. The best developed aspect of the theory treats systems defined by linear operators using well established techniques based on linear algebra, complex variable theory, and the theory of ordinary linear differential equations. Since design techniques for dynamical systems are closely related to their stability properties and since necessary and sufficient conditions for the stability of linear time-invariant systems have been generated over the past century, well-known design methods have been established for such systems. In contrast to this, the stability of nonlinear systems can be established for the most part only on a system-by-system basis and hence it is not surprising that design procedures that simultaneously meet the requirements of stability, robustness, and good dynamical response are not currently available for large classes of such systems.

In the past three decades major advances have been made in adaptive identification and control for identifying and controlling linear time-invariant plants with unknown parameters. The choice of the identifier and controller structures is based on well established results in linear systems theory. Stable adaptive laws for the adjustment of parameters in these cases which assure the global stability of the relevant overall systems are also based on properties of linear systems as well as stability results that

are well known for such systems [1]. In this paper our interest is in the identification and control of nonlinear dynamic plants using neural networks. Since very few results exist in nonlinear systems theory which can be directly applied, considerable care has to be exercised in the statement of the problems, the choice of the identifier and controller structures, as well as the generation of adaptive laws for the adjustment of the parameters.

Two classes of neural networks which have received considerable attention in the area of artificial neural networks in recent years are: 1) multilayer neural networks and 2) recurrent networks. Multilayer networks have proved extremely successful in pattern recognition problems [2]–[5] while recurrent networks have been used in associative memories as well as for the solution of optimization problems [6]–[9]. From a systems theoretic point of view, multilayer networks represent static nonlinear maps while recurrent networks are represented by nonlinear dynamic feedback systems. In spite of the seeming differences between the two classes of networks, there are compelling reasons to view them in a unified fashion. In fact, it is the conviction of the authors that dynamical elements and feedback will be increasingly used in the future, resulting in complex systems containing both types of networks. This, in turn, will necessitate a unified treatment of such networks. In Section III of this paper this viewpoint is elaborated further.

This paper is written with three principal objectives. This first and most important objective is to suggest identification as well as controller structures using neural networks for the adaptive control of unknown nonlinear dynamical systems. While major advances have been made in the design of adaptive controllers for linear systems with unknown parameters, such controllers cannot be used for the global control of nonlinear systems. The models suggested consequently represent a first step in this direction. A second objective is to present a prescriptive method for the dynamic adjustment of the parameters based on back propagation. The term dynamic back propagation is introduced in this context. The third and final objective is to state clearly the many theoretical assumptions that have to be made to have well posed problems. Block diagram representations of systems commonly used in systems theory, as well as computer simulations, are included throughout the paper to illustrate the various concepts introduced. The paper is organized as follows:

Section II deals with basic concepts and notational details used throughout the paper. In Section III, multilayer and recurrent networks are treated in a unified fashion. Section IV deals with static and dynamic methods for the adjustment of parameters of neural networks. Identification models are introduced in Section V while Section VI deals with the problem of adaptive control. Finally, in Section VII, some directions are given for future work.

II. PRELIMINARIES, BASIC CONCEPTS, AND NOTATION

In this section, many concepts related to the problem of identification and control are collected and presented for easy reference. While only some of them are directly used in the procedures discussed in Sections V and VI, all of them are relevant for a broad understanding of the role of neural networks in dynamical systems.

A. Characterization and Identification of Systems

System characterization and identification are fundamental problems in systems theory. The problem of characterization is concerned with the mathematical representation of a system; a model of a system is expressed as an operator P from an input space $\mathcal{U}$ into an output space $\mathcal{Y}$ and the objective is to characterize the class $\mathcal{P}$ to which P belongs. Given a class $\mathcal{P}$ and the fact that $P \in \mathcal{P}$, the problem of identification is to determine a class $\hat{\mathcal{P}} \subset \mathcal{P}$ and an element $\hat{P} \in \hat{\mathcal{P}}$ so that $\hat{P}$ approximates P in some desired sense. In static systems, the spaces $\mathcal{U}$ and $\mathcal{Y}$ are subsets of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, while in dynamical systems they are generally assumed to be bounded Lebesgue integrable functions on the interval $[0, T]$ or $[0, \infty)$. In both cases, the operator P is defined implicitly by the specified input–output pairs. The choice of the class of identification models $\hat{\mathcal{P}}$, as well as the specific method used to determine $\hat{P}$, depends upon a variety of factors which are related to the accuracy desired, as well as analytical tractability. These include the adequacy of the model $\hat{P}$ to represent P, its simplicity, the ease with which it can be identified, how readily it can be extended if it does not satisfy specifications, and finally whether the $\hat{P}$ chosen is to be used off line or on line. In practical applications many of these decisions naturally depend upon the prior information that is available concerning the plant to be identified.

1. Identification of Static and Dynamic Systems: The problem of pattern recognition is a typical example of identification of static systems. Compact sets $U_i \subset \mathbb{R}^n$ are mapped into elements $y_i \in \mathbb{R}^m; (i = 1, 2, \cdots,)$ in the output space by a decision function P. The elements of U_i denote the pattern vectors corresponding to class y_i. In dynamical systems, the operator P defining a given plant is implicitly defined by the input–output pairs of time functions $u(t), y(t), t \in [0, T]$. In both cases the objective is to determine $\hat{P}$ so that

$$\| \hat{y} - y \| = \| \hat{P}(u) - P(u) \| \leq \epsilon, \quad u \in \mathcal{U} \quad (1)$$

for some desired $\epsilon > 0$ and a suitably defined norm (denoted by $\| . \|$) on the output space. In (1), $\hat{P}(u) = \hat{y}$ de-

notes the output of the identification model and hence $\hat{y} - y \triangleq e$ is the error between the output generated by $\hat{P}$ and the observed output y. A more detailed statement of the identification problem of dynamical systems is given in Section II-C.

2. The Weierstrass Theorem and the Stone–Weierstrass Theorem: Let $C([a, b])$ denote the space of continuous real valued functions defined on the interval $[a, b]$ with the norm of $f \in C([a, b])$ defined by

$$\| f \| = \sup_t \left\{ | f(t) | : t \in [a, b] \right\}.$$

The famous approximation theorem of Weierstrass states that any function in $C([a, b])$ can be approximated arbitrarily closely by a polynomial. Alternately, the set of polynomials is dense in $C([a, b])$. Naturally, Weierstrass's theorem and its generalization to multiple dimensions finds wide application in the approximation of continuous functions $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ using polynomials (e.g., pattern recognition). A generalization of Weierstrass's theorem due to Stone, called the Stone–Weierstrass theorem can be used as the starting point for all the approximation procedures for dynamical systems.

Theorem: (Stone–Weierstrass [10]): Let $\mathcal{U}$ be a compact metric space. If $\hat{\mathcal{P}}$ is a subalgebra of $C(\mathcal{U}, \mathbb{R})$ which contains the constant functions and separates points of $\mathcal{U}$ then $\hat{\mathcal{P}}$ is dense in $C(\mathcal{U}, \mathbb{R})$.

In the problems of interest to us we shall assume that the plant P to be identified belongs to the space $\mathcal{P}$ of bounded, continuous, time-invariant and causal operators [11]. By the Stone–Weierstrass theorem, if $\hat{\mathcal{P}}$ satisfies the conditions of the theorem, a model belonging to $\hat{\mathcal{P}}$ can be chosen which approximates any specified operator $P \in \mathcal{P}$.

A vast literature exists on the characterization of nonlinear functionals and includes the classic works of Volterra, Wiener, Barret, and Urysohn. Using the Stone–Weierstrass theorem it can be shown that a given nonlinear functional under certain conditions can be represented by a corresponding series such as the Volterra series or the Wiener series. In spite of the impressive theoretical work that these represent, very few have found wide application in the identification of large classes of practical nonlinear systems. In this paper our interest is mainly on representations which permit on-line identification and control of dynamic systems in terms of finite dimensional nonlinear difference (or differential) equations. Such nonlinear models are well known in the systems literature and are considered in the following subsection.

B. Input-State-Output Representation of Systems

The method of representing dynamical systems by vector differential or difference equations is currently well established in systems theory and applies to a fairly large class of systems. For example, the differential equations

$$\frac{dx(t)}{dt} \triangleq \dot{x}(t) = \Phi[x(t), u(t)] \quad t \in \mathbb{R}^+ \tag{2}$$
$$y(t) = \Psi[x(t)]$$

where $x(t) \triangleq [x_1(t), x_2(t), \cdots, x_n(t)]^T$, $u(t) \triangleq [u_1(t), u_2(t), \cdots, u_p(t)]^T$ and $y(t) \triangleq [y_1(t), y_2(t), \cdots, y_m(t)]^T$ represent a p input m output system of order n with $u_i(t)$ representing the inputs, $x_i(t)$ the state variables, and $y_i(t)$ the outputs of the system. Φ and Ψ are static nonlinear maps defined as $\Phi : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ and $\Psi : \mathbb{R}^n \to \mathbb{R}^m$. The vector $x(t)$ denotes the state of the system at time t and is determined by the state at time $t_0 < t$ and the input u defined over the interval $[t_0, t)$. The output $y(t)$ is determined completely by the state of the system at time t. Equation (2) is referred to as the input-state-output representation of the system. In this paper we will be concerned with discrete-time systems which can be represented by difference equations corresponding to the differential equations given in (2). These take the form

$$x(k + 1) = \Phi[x(k), u(k)]$$
$$y(k) = \Psi[x(k)] \qquad (3)$$

where $u(.)$, $x(.)$, and $y(.)$ are discrete time sequences. Most of the results presented can, however, be extended to continuous time systems as well. If the system described by (3) is assumed to be linear and time invariant, the equations governing its behavior can be expressed as

$$x(k + 1) = Ax(k), + Bu(k)$$
$$y(k) = Cx(k) \qquad (4)$$

where A, B, and C are $(n \times n)$, $(n \times p)$, and $(m \times n)$ matrices, respectively. The system is then parameterized by the triple $\{C, A, B\}$. The theory of linear time-invariant systems, when C, A, and B are known, is very well developed and concepts such as controllability, stability, and observability of such systems have been studied extensively in the past three decades. Methods for determining the control input $u(.)$ to optimize a performance criterion are also well known. The tractability of these different problems may be ultimately traced to the fact that they can be reduced to the solution of n linear equations in n unknowns. In contrast to this, the problems involving nonlinear equations of the form (3), where the functions Φ and Ψ are known, result in nonlinear algebraic equations for the solution of which similar powerful methods do not exist. Consequently, as shown in the following sections, several assumptions have to be made to make the problems analytically tractable.

C. Identification and Control

1. Identification: When the functions Φ and Ψ in (3), or the matrices A, B, and C in (4), are unknown, the problem of identification of the unknown system (referred to as the plant in the following sections) arises [12]. This can be formally stated as follows [1]:

The input and output of a time-invariant, causal discrete-time dynamical plant are $u(.)$ and $y_p(.)$, respectively, where $u(.)$ is a uniformly bounded function of time. The plant is assumed to be stable with a known parameterization but with unknown values of the parame-
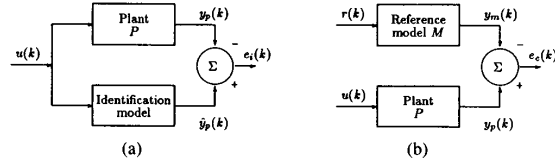

Fig. 1. (a) Identification. (b) Model reference adaptive control.

ters. The objective is to construct a suitable identification model (Fig. 1(a)) which when subjected to the same input $u(k)$ as the plant, produces an output $\hat{y}_p(k)$ which approximates $y_p(k)$ in the sense described by (1).

2. Control: Control theory deals with the analysis and synthesis of dynamical systems in which one or more variables are kept within prescribed limits. If the functions Φ and Ψ in (3) are known, the problem of control is to design a controller which generates the desired control input $u(k)$ based on all the information available at that instant k. While a vast body of frequency and time-domain techniques exist for the synthesis of controllers for linear systems of the form described in (4) with A, B, and C known, similar methods do not exist for nonlinear systems, even when the functions $\Phi(.,.)$ and $\Psi(.)$ are specified. In the last three decades there has been a great deal of interest in the control of plants when uncertainty exists regarding the dynamics of the plant [1]. To assure mathematical tractability, most of the effort has been directed towards the adaptive control of linear time-invariant plants with unknown parameters. Our interest in this paper lies primarily in the identification and control of unknown nonlinear dynamical systems.

Adaptive systems which make explicit use of models for control have been studied extensively. Such systems are commonly referred to as model reference adaptive control (MRAC) systems. The implicit assumption in the formulation of the MRAC problem is that the designer is sufficiently familiar with the plant under consideration so that he can specify the desired behavior of the plant in terms of the output of a reference model. The MRAC problem can be qualitatively stated as follows (Fig. 1(b)).

a. Model reference adaptive control: A plant P with an input-output pair $\{u(k), y_p(k)\}$ is given. A stable reference model M is specified by its input-output pair $\{r(k), y_m(k)\}$ where $r : N \to \mathbb{R}$ is a bounded function. The output $y_m(k)$ is the desired output of the plant. The aim is to determine the control input $u(k)$ for all $k \geq k_0$ so that

$$\lim_{k \to \infty} |y_p(k) - y_m(k)| \leq \epsilon$$

for some specified constant $\epsilon \geq 0$.

As described earlier, the choice of the identification model (i.e., its parameterization) and the method of adjusting its parameters based on the identification error $e_i(k)$ constitute the two principal parts of the identification problem. Determining the controller structure, and adjusting its parameters to minimize the error between the

output of the plant and the desired output, represent the corresponding parts of the control problem. In Section II-C-3, some well-known methods for setting up an identification model and a controller structure for a linear plant as well as the adjustment of identification and control parameters are described. Following this, in Section II-C-4, the problems encountered in the identification and control of nonlinear dynamical systems are briefly presented.

3. Linear Systems: For linear time-invariant plants with unknown parameters, the generation of identification models are currently well known. For a single-input single-output (SISO) controllable and observable plant, the matrix A and the vectors B and C in (4) can be chosen in such a fashion that the plant equation can be written as

$$y_p(k + 1) = \sum_{i=0}^{n-1} \alpha_i y_p(k - i) + \sum_{j=0}^{m-1} \beta_j u(k - j) \quad (5)$$

where α_i and β_j are constant unknown parameters. A similar representation is also possible for the multi-input multi-output (MIMO) case. This implies that the output at time $k + 1$ is a linear combination of the past values of both the input and the output. Equation (5) motivates the choice of the following identification models:

$$\hat{y}_p(k + 1) = \sum_{i=0}^{n-1} \hat{\alpha}_i(k)\hat{y}_p(k - i)$$

$$+ \sum_{j=0}^{m-1} \hat{\beta}_j(k)u(k - j) \quad (6)$$

(Parallel model)

$$\hat{y}_p(k + 1) = \sum_{i=0}^{n-1} \hat{\alpha}_i(k)y_p(k - i)$$

$$+ \sum_{j=0}^{m-1} \hat{\beta}_j(k)u(k - j)$$

(Series—parallel model) (7)

where $\hat{\alpha}_i(i = 0, 1, \cdots, n - 1)$ and $\hat{\beta}_j(j = 0, 1, \cdots, m - 1)$ are adjustable parameters. The output of the parallel identification model (6) at time $k + 1$ is $\hat{y}_p(k + 1)$ and is a linear combination of its past values as well as those of the input. In the series-parallel model, $\hat{y}_p(k + 1)$ is a linear combination of the past values of the input and output of the plant. To generate stable adaptive laws, the series-parallel model is found to be preferable. In such a case, a typical adaptive algorithm has the form

$$\hat{\alpha}_i(k + 1)$$

$$= \hat{\alpha}_i(k) - \eta$$

$$\cdot \frac{e(k + 1)y_p(k - i)}{1 + \sum_{i=0}^{n-1} y_p^2(k - i) + \sum_{j=0}^{m-1} u^2(k - j)}$$

(8)

where $\eta > 0$ determines the step size. In the following discussions, the constant vector of plant parameters $[\alpha_0,$

$\cdots, \alpha_{n-1}, \beta_0, \cdots, \beta_{m-1}]^T$ will be denoted by p and that of the identification model $[\hat{\alpha}_0, \cdots, \hat{\alpha}_{n-1}, \hat{\beta}_0, \cdots, \hat{\beta}_{m-1}]^T$ by $\hat{p}$.

Linear time-invariant plants which are controllable can be shown to be stabilizable by linear state feedback. This fact has been used to design adaptive controllers for such plants. For example, if an upper bound on the order of the plant is known, the control input can be generated as a linear combination of the past values of the input and output respectively. If $\theta(k)$ represents the control parameter vector, it can be shown that a constant vector θ^* exists such that when $\theta(k) \equiv \theta^*$ the plant together with the controller has the same input–output characteristics as the reference model. Adaptive algorithms for adjusting $\theta(k)$ in a stable fashion are now well known and have the general form shown in (8).

4. Nonlinear Systems: The importance of controllability and observability in the formulation of the identification and control problems for linear systems is evident from the discussion in Section II-C-3. Other well-known results in linear systems theory are also called upon to choose a reference model as well as a suitable parameterization of the plant and to assure the existence of a desired controller. In recent years a number of authors have addressed issues such as controllability, observability, feedback stabilization, and observer design for nonlinear systems [13]-[16]. In spite of such attempts constructive procedures, similar to those available for linear systems, do not exist for nonlinear systems. Hence, the choice of identification and controller models for nonlinear plants is a formidable problem and successful identification and control has to depend upon several strong assumptions regarding the input–output behavior of the plant. For example, if a SISO system is represented by the equation (3), we shall assume that the state of the system can be reconstructed from n measurements of the input and output. More precisely, $y_p(k) = \Psi[x(k)], y_p(k + 1) = \Psi[\Phi[x(k), u(k)]], \cdots, y_p(k + n - 1) = \Psi[\Phi[\cdots \Phi[\Phi[x(k), u(k)], u(k + 1)], \cdots, u(k + n - 2)]]$ yield n nonlinear equations in n unknowns $x(k)$ if $u(k), \cdots, u(k + n - 2), y_p(k), \cdots y_p(k + n - 1)$ are specified and we shall assume that for any set of values of $u(k)$ in a compact region in $\mathcal{U}$, a unique solution to the above problem exists. This permits identification procedures to be proposed for nonlinear systems along lines similar to those in the linear case.

Even when the function Φ is known in (3) and the state vector is accessible, the determination of $u(.)$ for the plant to have a desired trajectory is an equally difficult problem. Hence, for the generation of the control input, the existence of suitable inverse operators have to be assumed. If a controller structure is assumed to generate the input $u(.)$, further assumptions have to be made to assure the existence of a constant control parameter vector to achieve the desired objective. All these indicate that considerable progress in nonlinear control theory will be needed to obtain rigorous solutions to the identification and control problems.

In spite of the above comments, the linear models described in Section II-C-3 motivate the choice of structures for identifiers and controllers in the nonlinear case. It is in these structures that we shall incorporate neural networks as described in Sections V and VI. A variety of considerations discussed in Section III reveal that both multilayer neural networks as well as recurrent networks, which are currently being extensively studied, will feature as subsystems in the design of identifiers and controllers for nonlinear dynamical systems.

III. MULTILAYER AND RECURRENT NETWORKS

The assumptions that have to be made to assure well posed problems using models suggested in Sections V and VI are closely related to the properties of multilayer and recurrent networks. In this section, we describe briefly the two classes of neural networks and indicate why a unified treatment of the two may be warranted to deal with more complex systems in the future.

A. Multilayer Networks

A typical multilayer network with an input layer, an output layer, and two hidden layers is shown in Fig. 2. For convenience we denote this in block diagram form as shown in Fig. 3 with three weight matrices W^1, W^2, and W^3 and a diagonal nonlinear operator Γ with identical sigmoidal elements γ [i.e., $\gamma(x) = 1 - e^{-x}/1 + e^{-x}$] following each of the weight matrices. Each layer of the network can then be represented by the operator

$$N_i[u] = \Gamma[W^i u] \qquad (9)$$

and the input–output mapping of the multilayer network can be represented by

$$y = N[u] = \Gamma[W^3 \Gamma[W^2 \Gamma[W^1 u]]] = N_3 N_2 N_1[u]. \qquad (10)$$

In practice, multilayer networks have been used successfully in pattern recognition problems [2]–[5]. The weights of the network W^1, W^2, and W^3 are adjusted as described in Section IV to minimize a suitable function of the error e between the output y of the network and a desired output y_d. This results in the mapping function $N[u]$ realized by the network, mapping vectors into corresponding output classes. Generally a discontinuous mapping such as a nearest neighbor rule is used at the last stage to map the input sets into points in the range space corresponding to output classes. From a systems theoretic point of view, multilayer networks can be considered as versatile nonlinear maps with the elements of the weight matrices as parameters. In the following sections we shall use the terms "weights" and "parameters" interchangeably.

B. Recurrent Networks

Recurrent networks, introduced in the works of Hopfield [6] and discussed quite extensively in the literature, provide an alternative approach to pattern recognition. One version of the network suggested by Hopfield con-
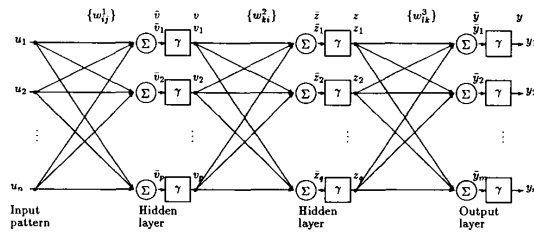

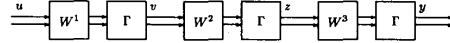Fig. 2. A three layer neural network.

Fig. 3. A block diagram representation of a three layer network.

sists of a single layer network N_1, included in feedback configuration, with a time delay (Figs. 4 and 5). Such a network represents a discrete-time dynamical system and can be described by

$$x(k + 1) = N_1[x(k)], \qquad x(0) = x_0.$$

Given an initial value x_0, the dynamical system evolves to an equilibrium state if N_1 is suitably chosen. The set of initial conditions in the neighborhood of x_0 which converge to the same equilibrium state is then identified with that state. The term "associative memory" is used to describe such systems. Recently, both continuous-time and discrete-time recurrent networks have been studied with constant inputs [17]. The inputs rather than the initial conditions represent the patterns to be classified in this case. In the continuous-time case, the dynamic system in the feedback path has a diagonal transfer matrix with identical elements $1/(s + \alpha)$ along the diagonal. The system is then represented by the equation

$$\dot{x} = -\alpha x + N_1[x] + I \qquad (11)$$

so that $x(t) \in \mathbb{R}^n$ is the state of the system at time t, and the constant vector $I \in \mathbb{R}^n$ is the input.

C. A Unified Approach

In spite of the seeming differences between the two approaches to pattern recognition using neural networks, it is clear that a close relation exists between them. Recurrent networks with or without constant inputs are merely nonlinear dynamical systems and the asymptotic behavior of such systems depends both on the initial conditions as well as the specific input used. In both cases, this depends critically on the nonlinear map represented by the neural network used in the feedback loop. For example, when no input is used, the equilibrium state of the recurrent network in the discrete case is merely the fixed point of the mapping N_1. Thus the existence of a fixed point, the conditions under which it is unique, the maximum number of fixed points that can be achieved in a given network are all relevant to both multilayer and recurrent networks. Much of the current literature deals with such problems [18] and for mathematical tractability most of them as-
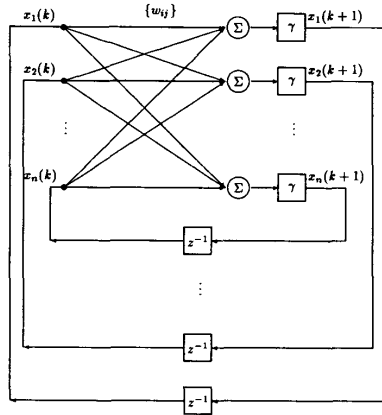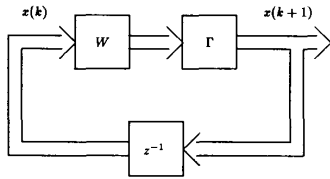
Fig. 4. The Hopfield network.

Fig. 5. Block diagram representation of the Hopfield network.

Fig. 6. (a) Representation 1. (b) Representation 2. (c) Representation 3. (d) Representation 4.

sume that recurrent networks contain only single layer networks (i.e., $N_1[.]$). As mentioned earlier, inputs when they exist are assumed to be constant. Recently, two layer recurrent networks have also been considered [19] and more general forms of recurrent networks can be constructed by including multilayer networks in the feedback loop [20]. In spite of the interesting ideas that have been presented in these papers, our understanding of such systems is still far from complete. In the identification and control problems considered in Sections V and VI, multilayer networks are used in cascade and feedback configurations and the inputs to such models are functions of time.

D. Generalized Neural Networks

From the above discussion, it follows that the basic elements in a multilayer network is the mapping $N_1[.] = \Gamma[W^1.]$, while the addition of the time delay element z^{-1} in the feedback path (Fig. 5) results in a recurrent network. In fact, general recurrent networks can be constructed composed of only the basic operations of 1) delay, 2) summation, and 3) the nonlinear operator $N_i[.]$. In continuous-time networks, the delay operator is replaced by an integrator. In some cases (as in (11)) multiplication by a constant is also allowed. Hence such networks are nonlinear feedback systems which consist only of elements $N_1[.]$, in addition to the usual operations found in linear systems.

Since arbitrary linear time-invariant dynamical systems can be constructed using the operations of summation,
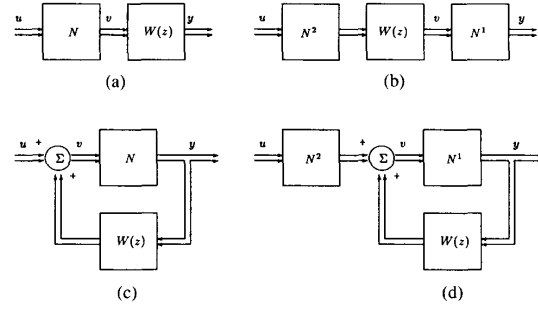
multiplication by a constant and time delay, the class of nonlinear dynamical systems that can be generated using generalized neural networks can be represented in terms of transfer matrices of linear systems [i.e., $W(z)$] and nonlinear operators $N[.]$. Fig. 6 shows these operators connected in cascade and feedback in four configurations which represent the building blocks for more complex systems. The superscript notation N^i is used in the figures to distinguish between different multilayer networks in any specific representation.

From the discussion of generalized neural networks, it follows that the mapping properties of $N_i[.]$ and consequently $N[.]$ (as defined in (10)) play a central role in all analytical studies of such networks. It has recently been shown in [21], using the Stone–Weierstrass theorem, that a two layer network with an arbitrarily large number of nodes in the hidden layer can approximate any continuous function $f \in C(\mathbb{R}^n, \mathbb{R}^m)$ over a compact subset of $\mathbb{R}^n$. This provides the motivation to assume that the class of generalized networks described is adequate to deal with a large class of problems in nonlinear systems theory. In fact, all the structures used in Section V and VI for the construction of identification and controller models are generalized neural networks and are closely related to the configurations shown in Fig. 6. For ease of discussion in the rest of the paper, we shall denote the class of functions generated by a network containing N layers by the symbol $\mathfrak{N}^N_{i_1, i_2, \cdots, i_{N+1}}$. Such a network has i_1 inputs, i_{N+1} outputs and $(N - 1)$ sets of nodes in the hidden layers, each containing $i_2, i_3, \cdots, i_N$ nodes, respectively.

IV. BACK PROPAGATION IN STATIC AND DYNAMIC SYSTEMS

In both static identification (e.g., pattern recognition) and dynamic system identification of the type treated in this paper, if neural networks are used, the objective is to determine an adaptive algorithm or rule which adjusts the parameters of the network based on a given set of input–output pairs. If the weights of the networks are considered as elements of a parameter vector θ, the learning process involves the determination of the vector θ^* which optimizes a performance function J based on the output error. Back propagation is the most commonly used method for

this purpose in static contexts. The gradient of the performance function with respect to θ is computed as $\nabla_\theta J$ and θ is adjusted along the negative gradient as

$$\theta = \theta_{\text{nom}} - \eta \nabla_\theta J \big|_{\theta = \theta_{\text{nom}}}$$

where η, the step size, is a suitably chosen constant and θ_{nom} denotes the nominal value of θ at which the gradient is computed. In this section, a diagrammatic representation of back propagation is first introduced. Following this, a method of extending this concept to dynamical systems is described and the term dynamic back propagation is defined. Prescriptive methods for the adjustment of weight vectors are suggested which can be used in the identification and control problems of the type discussed in Sections V and VI.

In the early 1960's, when the adaptive identification and control of linear dynamical systems were extensively studied, sensitivity models were developed to generate the partial derivatives of the performance criteria with respect to the adjustable parameters of the system. These models were the first to use sensitivity methods for dynamical systems and provided a great deal of insight into the necessary adaptive system structure [22]–[25]. Since conceptually the above problem is identical to that of determining the parameters of neural networks in identification and control problems, it is clear that back-propagation can be extended to dynamical systems as well.

A. A Diagrammatic Representation of Back Propagation

In this section we introduce a diagrammatic representation of back propagation. While the diagrammatic and algorithmic representations are informationally equivalent, their computational efficiency is different since the former preserves information about topological and geometric relations. In particular, the diagrammatic representation provides a better visual understanding of the entire process of back propagation, lends itself to modifications which are computationally more efficient and suggests novel modifications of the existing structure to include other functional extensions.

In the three layered network shown in Fig. 2, $u^T \triangleq [u_1, u_2, \cdots, u_n]$ denotes the input pattern vector while $y^T \triangleq [y_1, y_2, \cdots, y_m]$ is the output vector. $v^T \triangleq [v_1, v_2, \cdots, v_p]$ and $z^T \triangleq [z_1, z_2, \cdots, z_q]$ are the outputs at the first and the second hidden layers, respectively. $\{w_{ij}^1\}_{p \times n}$, $\{w_{ki}^2\}_{q \times p}$ and $\{w_{lk}^3\}_{m \times q}$ are the weight matrices associated with the three layers as shown in Fig. 2. The vectors $\bar{v} \in \mathbb{R}^p$, $\bar{z} \in \mathbb{R}^q$ and $\bar{y} \in \mathbb{R}^m$ are as shown in Fig. 2 with $\gamma(\bar{v}_i) = v_i$, $\gamma(\bar{z}_k) = z_k$ and $\gamma(\bar{y}_l) = y_l$ where $\bar{v}_i, \bar{z}_k$, and $\bar{y}_l$ are elements of $\bar{v}$, $\bar{z}$ and $\bar{y}$ respectively. If $y_d^T = [y_{d1}, y_{d2}, \cdots, y_{dm}]$ is the desired output vector, the output error vector for a given input pattern u is defined as $e \triangleq y - y_d$. The performance criterion J is then defined as

$$J = \sum_s \|e\|^2$$

where the summation is carried out over all patterns in a given set S. If the input patterns are assumed to be presented at each instant of time, the performance criterion J may be interpreted as the sum squared error over an interval of time. It is this interpretation which is found to be relevant in dynamic systems. In the latter case, the inputs and outputs are time sequences and the performance criterion J has the form $(1/T) \sum_{i=k-T+1}^{k} e^2(i)$, where T is a suitably chosen integer.

While strictly speaking the adjustment of the parameters should be carried out by determining the gradient of J in parameter space, the procedure commonly followed is to adjust it at every instant based on the error at that instant and a small step size η. If θ_j represents a typical parameter, $\partial e / \partial \theta_j$ has to be determined to compute the gradient as $e^T (\partial e / \partial \theta_j)$. The back propagation method is a convenient method of determining this gradient.

Fig. 7 shows the diagrammatic representation of back propagation for the three layer network shown in Fig. 2. The analytical method of deriving the gradient is well known in the literature and will not be repeated here. Fig. 7 merely shows how the various components of the gradient are realized. In our example, it is seen that signals u, v, and z and $\gamma'(\bar{v})$, $\gamma'(\bar{z})$, and $\gamma'(\bar{y})$, as well as the error vector, are used in the computation of the gradient (where $\gamma'(x)$ is the derivative of $\gamma(x)$ with respect to x). qm, pq, and np multiplications are needed to compute the partial derivatives with respect to the elements of W^3, W^2, and W^1, respectively. The structure of the weight matrices in the network used to compute the derivatives is seen to be identical to that in the original network while the signal flow is in the opposite direction, justifying the use of the term ''back propagation.'' For further details regarding the diagrammatic representation, the reader is referred to [26] and [27]. The advantages of the diagrammatic representation mentioned earlier are evident from Fig. 7. More relevant to our purpose is that the same representation can be readily modified for the dynamic case. In fact, the diagrammatic representation was used extensively in all the simulation studies described in Sections V and VI.

B. Dynamic Back Propagation

In a causal dynamical system the change in a parameter at time k will produce a change in the output $y(t)$ for all $t \geq k$. For example, given a nonlinear dynamical system $x(k + 1) = \Phi[x(k), u(k), \theta]$; $y(k) = \Psi[x(k)]$ where θ is a parameter, u is the input and x is the state vector defined in (3), the partial derivative of $y(k)$ with respect to θ can be obtained by solving the linear state equations

$$z(k + 1) = A(k)z(k) + v(k), \quad z(k_0) = 0$$

$$w(k) = C(k)z(k) \tag{12}$$

where $z(k) = \partial x(k)/\partial \theta \in \mathbb{R}^n$, $A(k) = \Phi_x(k) \in \mathbb{R}^{n \times n}$, $v(k) = \Phi_\theta(k) \in \mathbb{R}^n$, $w(k) = \partial y(k)/\partial \theta \in \mathbb{R}^m$ and $C(k)$
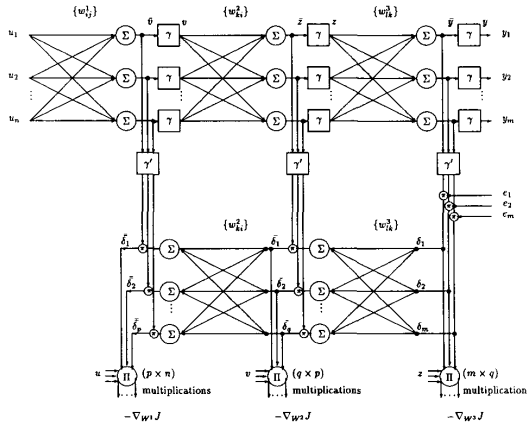
Fig. 7. Architecture for back propagation.

$= \Psi_x(k) \in \mathbb{R}^{m \times n}$. Φ_x and Ψ_x are Jacobian matrices and the vector Φ_θ represents the partial derivative of Φ with respect to θ. Equation (12) represents the linearized equations of the nonlinear system around the nominal trajectory and input. If $A(k)$, $v(k)$, and $C(k)$ can be computed, $w(k)$, the partial derivative of y with respect to θ can be obtained as the output of a dynamic sensitivity model.

In the previous section, generalized neural networks were defined and four representations of such networks with dynamical systems and multilayer neural networks connected in series and feedback were shown in Fig. 6. Since complex dynamical systems can be expressed in terms of these four representations, the back-propagation method can be extended to such systems if the partial derivative of the outputs with respect to the parameters can be determined for each of the representations. In the following we indicate briefly how (12) can be specialized to these four cases. In all cases it is assumed that the partial derivative of the output of a multilayer neural network with respect to one of the parameters can be computed using static back propagation and can be realized as the output of the network in Fig. 7.

In representation 1, the desired output $y_d(k)$ as well as the error $e(k) \triangleq y(k) - y_d(k)$ are functions of time. Representation 1 is the simplest situation that can arise in dynamical systems. This is because

$$\frac{\partial e(k)}{\partial \theta_j} = \frac{\partial y(k)}{\partial \theta_j} = W(z)\frac{\partial v}{\partial \theta_j}$$

where θ_j is a typical parameter of the network N. Since $\partial v/\partial \theta_j$ can be computed at every instant using static back propagation, $\partial e(k)/\partial \theta_j$ can be realized as the output of a dynamical system $W(z)$ whose inputs are the partial derivatives generated.

In representation 2, the determination of the gradient is rendered more complex by the presence of neural network N^1. If θ_j is a typical parameter of N^1, the partial derivative

$\partial e(k)/\partial \theta_j$ is computed by static back propagation. However, if θ_j is a typical parameter of N^2

$$\frac{\partial y_i}{\partial \theta_j} = \sum_l \frac{\partial y_i}{\partial v_l}\frac{\partial v_l}{\partial \theta_j}.$$

Since $\partial v/\partial \theta_j$ can be computed using the method described in representation 1 and $\partial y_i/\partial v$ can be obtained by static back propagation, the product of the two yield the partial derivative of the signal y_i with respect to the parameter θ_j.

Representation 3 shows a neural network connected in feedback with a transfer matrix $W(z)$. The input to the nonlinear feedback system is a vector $u(k)$. If θ_j is a typical parameter of the neural network, the aim is to determine the derivatives $\bar{\partial} y_i(k)/\bar{\partial}\theta_j$ for $i = 1, 2, \cdots, m$ and all $k \geq 0$. We observe here for the first time a situation not encountered earlier, in that $\bar{\partial} y_i(k)/\bar{\partial}\theta_j$ is the solution of a difference equation, i.e., $\bar{\partial} y_i(k)/\bar{\partial}\theta_j$ is affected by its own past values

$$\frac{\bar{\partial} y}{\bar{\partial}\theta_j} = \frac{\partial N[v]}{\partial v} W(z)\frac{\bar{\partial} y}{\bar{\partial}\theta_j} + \frac{\partial N[v]}{\partial \theta_j}. \quad (13)$$

In (13), $\bar{\partial} y/\bar{\partial}\theta_j$ is a vector and $\partial N[v]/\partial v$ and $\partial N[v]/\partial \theta_j$ are the Jacobian matrix and a vector, respectively, which are evaluated around the nominal trajectory. Hence it represents a linearized difference equation in the variables $\bar{\partial} y_i/\bar{\partial}\theta_j$. Since $\partial N[v]/\partial v$ and $\partial N[v]/\partial \theta_j$ can be computed at every instant of time, the desired partial derivatives can be generated as the output of a dynamical system shown in Fig. 8(a) (the bar notation $\bar{\partial} y/\bar{\partial}\theta_j$ is used in (13) to distinguish between $\partial y/\partial \theta_j$ and $\partial N[v]/\partial \theta_j$).

In the final representation, the feedback system is preceded by a neural network N^2. The presence of N^2 does not affect the computation of the partial derivatives of the output with respect to the parameters of N^1. However, if θ_j is a typical parameter of N^2, it can be shown that $\partial y/\partial \theta_j$ can be obtained as

$$\frac{\partial y}{\partial \theta_j} = \frac{\partial N^1[v]}{\partial v}\left[\frac{\partial N^2[u]}{\partial \theta_j} + W(z)\frac{\partial y}{\partial \theta_j}\right]$$

or alternately it can be represented as the output of the dynamical system shown in Fig. 8(b) whose inputs can be computed at every instant of time.

In all the problems of identification and control that we will be concerned with in the following sections, the matrix $W(z)$ is diagonal and consists only of elements of the form z^{-d_i} (i.e., a delay of d_i units). Further since dynamic back propagation is considerably more involved than static back propagation, the structure of the identification models is chosen, wherever possible, so that the latter can be used. The models of back propagation developed here can be applied to general control problems where neural networks and linear dynamical systems are interconnected in arbitrary configurations and where static back propagation cannot be justified. For further details the reader is
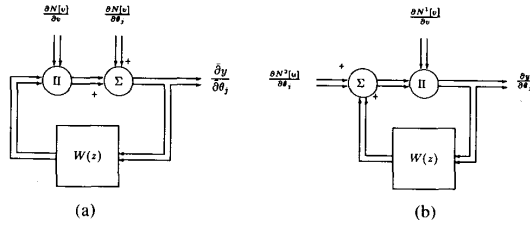
Fig. 8. (a) Generation of gradient in representation 3. (b) Generation of gradient in representation 4.

referred to [27]. A paper based on [27] but providing details concerning the implementation of the algorithms in practical applications is currently under preparation.

V. IDENTIFICATION

As mentioned in Section III, the ability of neural networks to approximate large classes of nonlinear functions sufficiently accurately make them prime candidates for use in dynamic models for the representation of nonlinear plants. The fact that static and dynamic back-propagation methods, as described in Section IV, can be used for the adjustment of their parameters also makes them attractive in identifiers and controllers. In this section four models for the representation of SISO plants are introduced which can also be generalized to the multivariable case. Following this, identification models are suggested containing multilayer neural networks as subsystems. These models are motivated by the models which have been used in the adaptive systems literature for the identification and control of linear systems and can be considered as their generalization to nonlinear systems.

A. Characterization

The four models of discrete-time plants introduced here can be described by the following nonlinear difference equations:

Model I: $y_p(k + 1)$

$$= \sum_{i=0}^{n-1} \alpha_i y_p(k - i)$$

$$+ g[u(k), u(k - 1), \cdots, u(k - m + 1)]$$

Model II: $y_p(k + 1)$

$$= f[y_p(k), y_p(k - 1), \cdots, y_p(k - n + 1)]$$

$$+ \sum_{i=0}^{m-1} \beta_i u(k - i)$$

Model III: $y_p(k + 1)$

$$= f[y_p(k), y_p(k - 1), \cdots, y_p(k - n + 1)]$$

$$+ g[u(k), u(k - 1), \cdots, u(k - m + 1)]$$

Model IV: $y_p(k + 1)$

$$= f[y_p(k), y_p(k - 1), \cdots, y_p(k - n + 1);$$

$$u(k), u(k - 1), \cdots, u(k - m + 1)]$$

$$(14)$$

where $[u(k), y_p(k)]$ represents the input–output pair of the SISO plant at time k, and $m \leq n$. The block diagram representation of the various models are shown in Fig. 9. The functions $f: \mathbb{R}^n \rightarrow \mathbb{R}$ in Models II and III and $f: \mathbb{R}^{n+m} \rightarrow \mathbb{R}$ in Model IV, and $g: \mathbb{R}^m \rightarrow \mathbb{R}$ in (14) are assumed to be differentiable functions of their arguments. In all the four models, the output of the plant at the time $k + 1$ depends both on its past n values $y_p(k - i)$ ($i = 0, 1, \cdots, n - 1$) as well as the past m values of the input $u(k - j)$ ($j = 0, 1, \cdots, m - 1$). The dependence on the past values $y_p(k - i)$ is linear in Model I while in Model II the dependence on the past values of the input $u(k - j)$ is assumed to be linear. In Model III, the nonlinear dependence of $y_p(k + 1)$ on $y_p(k - i)$ and $u(k - j)$ is assumed to be separable. It is evident that Model IV in which $y_p(k + 1)$ is a nonlinear function of $y_p(k - i)$ and $u(k - j)$ subsumes Models I-III. If a general nonlinear SISO plant can be described by an equation of the form (3) and satisfies the stringent observability condition discussed in Section II-C-4, it can be represented by such a model. In spite of its generality, Model IV is, however, analytically the least tractable and hence for practical applications some of the other models are found to be more attractive. For example, as will be apparent in the following section, Model II is particularly suited for the control problem.

From the results given in Section III, it follows that under fairly weak conditions on the function f and/or g in (14), multilayer neural networks can be constructed to approximate such mappings over compact sets. We shall assume for convenience that f and/or g belong to a known class $\mathfrak{N}_{i1, i2, \cdots, iN+1}^N$ in the domain of interest, so that the plant can be represented by a generalized neural network as discussed in Section III. This assumption motivates the choice of the identification models and allows the statement of well posed identification problems. In particular, the identification models have the same structure as the plant but contain neural networks with adjustable parameters.

Let a nonlinear dynamic plant be represented by one of the four models described in (14). If such a plant is to be identified using input–output data, it must be further assumed that it has bounded outputs for the class of permissible inputs. This implies that the model chosen to represent the plant also enjoys this property. In the case of Model I, this implies that the roots of the characteristic equation $z^n - \alpha_0 z^{n-1} - \cdots - \alpha_{n-2} z - \alpha_{n-1} = 0$ lie in the interior of the unit circle. In the other three cases no such simple algebraic conditions exist. Hence the study of the stability properties of recurrent networks containing multilayer networks represents an important area of research.
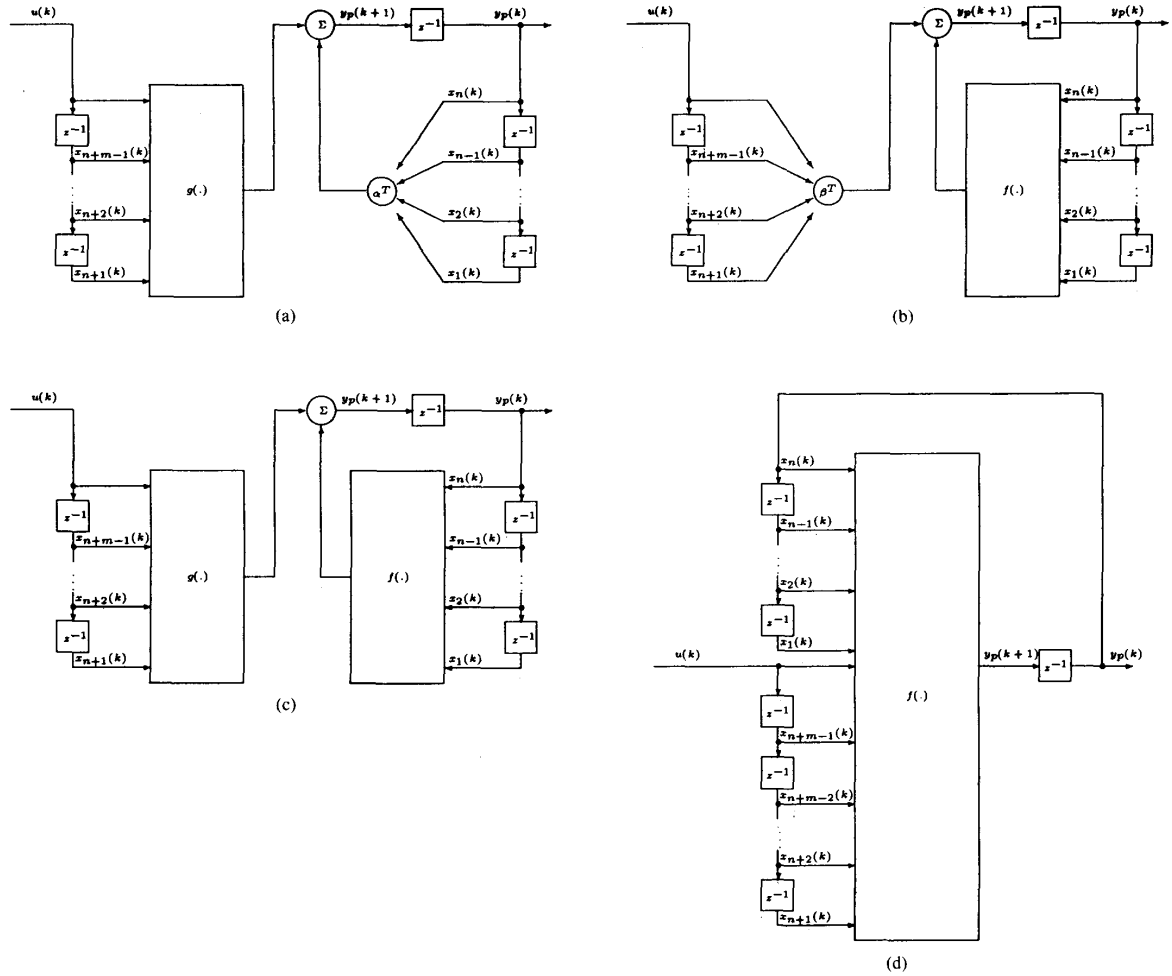
Fig. 9. Representation of SISO plants. (a) Model I. (b) Model II. (c) Model III. (d) Model IV.

The models described thus far are for the representation of discrete-time plants. Continuous-time analogs of these models can be described by differential equations, as stated in Section II. While we shall deal exclusively with discrete-time systems, the same methods also carry over to the continuous time case.

B. Identification

The problem of identification, as described in Section II-C, consists of setting up a suitably parameterized identification model and adjusting the parameters of the model to optimize a performance function based on the error between the plant and the identification model outputs. Since the nonlinear functions in the representation of the plant are assumed to belong to a known class $\mathfrak{N}_{i_1,i_2,\cdots,i_{N+1}}^N$ in the domain of interest, the structure of the identification model is chosen to be identical to that of the plant. By assumption, weight matrices of the neural networks in the identification model exist so that, for the same initial conditions, both plant and model have the same output for any specified input. Hence the identification procedure consists in adjusting the parameters of the neural networks in the model using the method described in Section IV based on the error between the plant and model outputs. However, as shown in what follows, suitable precautions have to be taken to ensure that the procedure results in convergence of the identification model parameters to their desired values.

1. Parallel Identification Model: Fig. 10(a) shows a plant which can be represented by Model I with $n = 2$ and $m = 1$. To identify the plant one can assume the structure of the identification model shown in Fig. 10(a) and described by the equation

$$\hat{y}_p(k + 1) = \hat{\alpha}_0\hat{y}_p(k) + \hat{\alpha}_1\hat{y}_p(k - 1) + N[u(k)].$$
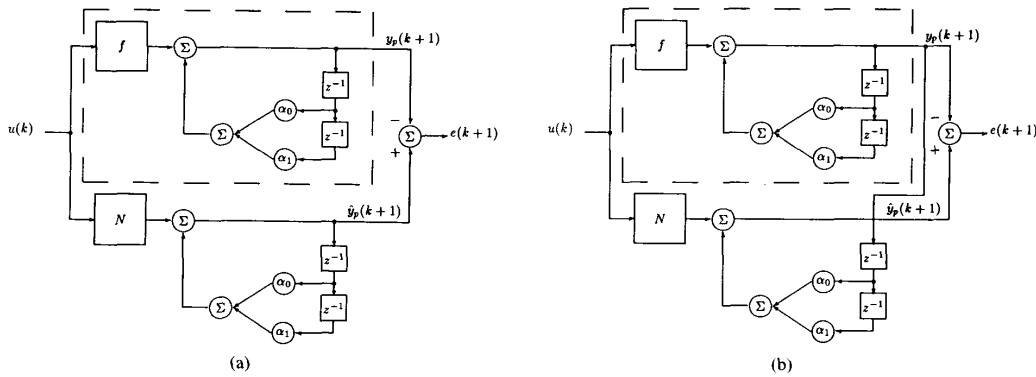
Fig. 10. (a) Parallel identification model. (b) Series-parallel identification model.

As mentioned in Section II-C-3, this is referred to as a parallel model. Identification then involves the estimation of the parameters $\hat{\alpha}_i$ as well as the weights of the neural network using dynamic back propagation based on the error $e(k)$ between the model output $\hat{y}_p(k)$ and the actual output $y_p(k)$.

From the assumptions made earlier, the plant is bounded-input bounded-output (BIBO) stable in the presence of an input (in the assumed class). Hence, all the signals in the plant are uniformly bounded. In contrast to this, the stability of the identification model as described here with a neural network cannot be assured and has to be proved. Hence if a parallel model is used, there is no guarantee that the parameters will converge or that the output error will tend to zero. In spite of two decades of work, conditions under which the parallel model parameters will converge even in the linear case are at present unknown. Hence, for plant representations using Models I–IV, the following identification model, known as the series-parallel model, is used.

2. Series-Parallel Model: In contrast to the parallel model described above, in the series-parallel model the output of the plant (rather than the identification model) is fed back into the identification model as shown in Fig. 10(b). This implies that in this case the identification model has the form

$$\hat{y}_p(k + 1) = \hat{\alpha}_0 y_p(k) + \hat{\alpha}_1 y_p(k - 1) + N[u(k)].$$

We shall use the same procedure with all the four models described earlier. The series-parallel identification model corresponding to a plant represented by Model IV has the form shown in Fig. 11. TDL in Fig. 11 denotes a tapped delay line whose output vector has for its elements the delayed values of the input signal. Hence the past values of the input and the output of the plant form the input vector to a neural network whose output $\hat{y}_p(k)$ corresponds to the estimate of the plant output at any instant of time k. The series-parallel model enjoys several advan-

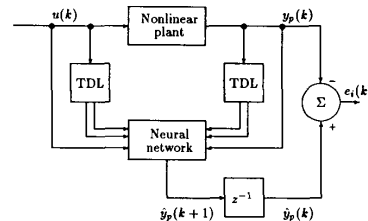Fig. 11. Identification of nonlinear plants using neural networks.

tages over the parallel model. Since the plant is assumed to be BIBO stable, all the signals used in the identification procedure (i.e., inputs to the neural networks) are bounded. Further, since no feedback loop exists in the model, static back propagation can be used to adjust the parameters reducing the computational overhead substantially. Finally, assuming that the output error tends to a small value asymptotically so that $y_p(k) \approx \hat{y}_p(k)$, the series-parallel model may be replaced by a parallel model without serious consequences. This has practical implications if the identification model is to be used off line. In view of the above considerations the series-parallel model is used in all the simulations in this paper.

C. Simulation Results

In this section simulation results of nonlinear plant identification using the models suggested earlier are presented. Six examples are presented where the prior information available dictates the choice of one of the Models I–IV. Each example is chosen to emphasize a specific point. In the first five examples, the series-parallel model is used to identify the given plant and static back-propagation is used to adjust parameters of the neural networks. A final example is used to indicate how dynamic back propagation may be used in identification problems. Due to space limitations, only the principal results are presented here. The reader interested in further details is referred to [27]–[29].
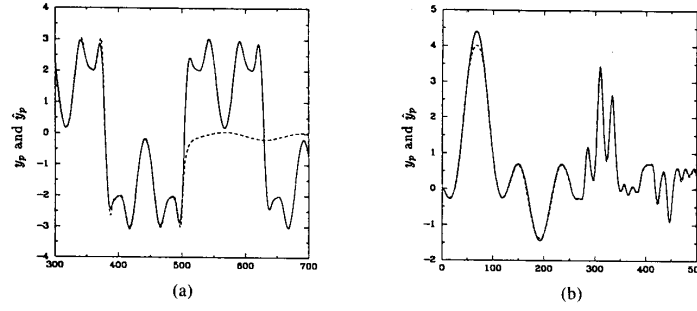
Fig. 12. Example 1: (a) Outputs of the plant and identification model when adaptation stops at $k = 500$. (b) Response of plant and identification model after identification using a random input.

1. Example 1: The plant to be identified is governed by the difference equation

$$y_p(k + 1) = 0.3y_p(k) + 0.6y_p(k - 1) + f[u(k)]$$

$$(15)$$

where the unknown function has the form $f(u) = 0.6 \sin(\pi u) + 0.3 \sin(3\pi u) + 0.1 \sin(5\pi u)$. From (15), it is clear that the unforced linear system is asymptotically stable and hence any bounded input results in a bounded output. In order to identify the plant, a series-parallel model governed by the difference equation

$$\hat{y}_p(k + 1) = 0.3y_p(k) + 0.6y_p(k - 1) + N[u(k)]$$

was used. The weights in the neural network were adjusted at every instant of time ($T_i = 1$) using static back propagation. The neural network belonged to the class $\mathfrak{N}_{1,20,10,1}^3$ and the gradient method employed a step size of $\eta = 0.25$. The input to the plant and the model was a sinusoid $u(k) = \sin(2\pi k/250)$. As seen from Fig. 12(a), the output of the model follows the output of the plant almost immediately but fails to do so when the adaptation process is stopped at $k = 500$, indicating that the identification of the plant is not complete. Hence the identification procedure was continued for 50 000 time steps using a random input whose amplitude was uniformly distributed in the interval $[-1, 1]$ at the end of which the adaptation was terminated. Fig. 12(b) shows the outputs of the plant and the trained model. The nonlinear function in the plant in this case is $f[u] = u^3 + 0.3u^2 - 0.4u$. As can be seen from the figure, the identification error is small even when the input is changed to a sum of two sinusoids $u(k) = \sin(2\pi k/250) + \sin(2\pi k/25)$ at $k = 250$.

2. Example 2: The plant to be identified is described by the second-order difference equation

$$y_p(k + 1) = f[y_p(k), y_p(k - 1)] + u(k)$$

where

$$f[y_p(k), y_p(k - 1)]$$
$$= \frac{y_p(k) y_p(k - 1)[y_p(k) + 2.5]}{1 + y_p^2(k) + y_p^2(k - 1)}. \quad (16)$$

This corresponds to Model II. A series-parallel identifier of the type discussed earlier is used to identify the plant from input–output data and is described by the equation

$$\hat{y}_p(k + 1) = N[y_p(k), y_p(k - 1)] + u(k) \quad (17)$$

where N is a neural network with $N \in \mathfrak{N}_{2,20,10,1}^3$. The identification process involves the adjustment of the weights of N using back propagation.

Some prior information concerning the input–output behavior of the plant is needed before identification can be undertaken. This includes the number of equilibrium states of the unforced system and their stability properties, the compact set $\mathfrak{U}$ to which the input belongs and whether the plant output is bounded for this class of inputs. Also, it is assumed that the mapping N can approximate f over the desired domain.

a. Equilibrium states of the unforced system: The equilibrium states of the unforced system $y_p(k + 1) = f[y_p(k), y_p(k - 1)]$ with f as defined in (16) are $(0, 0)$ and $(2, 2)$, respectively, in the state space. This implies that while in equilibrium without an input, the output of the plant is either the sequence $\{0\}$ or the sequence $\{2\}$. Further, for any input $|u(k)| \leq 5$, the output of the plant is uniformly bounded for initial conditions $(0, 0)$ and $(2, 2)$ and satisfies the inequality $|y_p(k)| \leq 13$.

Assuming different initial conditions in the state space and with zero input, the weights of the neural network were adjusted so that the error $e(k + 1) = y_p(k + 1) - N[y_p(k), y_p(k - 1)]$ is minimized. When the weights converged to constant values, the equation $\hat{y}_p(k + 1) = N[\hat{y}_p(k), \hat{y}_p(k - 1)]$ was simulated for initial conditions within a radius of 4. The identified system was found to have the same trajectories as the plant for the same initial conditions. The behavior of the plant and the identified model for different initial conditions are shown in Fig. 13. It must be emphasized here that in practice the initial conditions of the plant cannot be chosen at the discretion of the designer and must be realized only by using different inputs to the plant.

b. Identification: While the neural network realized above can be used in the identification model, a separate
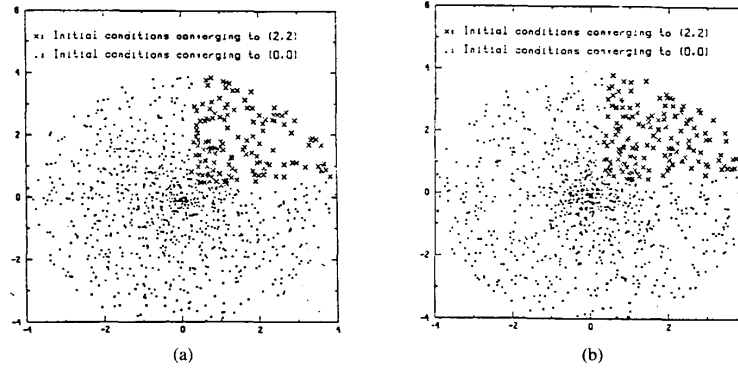
Fig. 13. Example 2: Behavior of the unforced system. (a) Actual plant.
(b) Identified model of the plant.

simulation was carried out using both inputs and outputs
and a series-parallel model. The input $u(k)$ was assumed
to be an i.i.d. random signal uniformly distributed in the
interval $[-2, 2]$ and a step size of $\eta = 0.25$ was used in
the gradient method. The weights in the neural network
were adjusted at intervals of five steps using the gradient
of $\Sigma_{i=k-4}^{k} e^2(i)$. Fig. 14 shows the outputs of the plant
and the model after the identification procedure was ter-
minated at $k = 100\,000$.

3. Example 3: In Example 2, the input is seen to occur
linearly in the difference equation describing the plant. In
this example the plant is described by Model III and has
the form

$$y_p(k + 1) = \frac{y_p(k)}{1 + y_p(k)^2} + u^3(k)$$

which corresponds to $f[y_p(k)] = y_p(k)/(1 + y_p(k)^2)$
and $g[u(k)] = u^3(k)$ in (14). A series-parallel identifi-
cation model consists of two neural networks N_f and N_g
belonging to $\mathfrak{N}_{1,20,10,1}^3$ and can be described by the differ-
ence equation

$$\hat{y}_p(k + 1) = N_f[y_p(k)] + N_g[u(k)].$$

The estimates $\hat{f}$ and $\hat{g}$ are obtained by using neural net-
works N_f and N_g. The weights in the neural networks were
adjusted at every instant of time using a step size of $\eta = 0.1$ and was continued for $100\,000$ time steps. Since the
input was a random input in interval $[-2, 2]$, $\hat{g}$ approx-
imates g only over this interval. Since this in turn results
in the variation of y_p over the interval $[-10, 10]$, $\hat{f}$ ap-
proximates f over the latter interval. The functions $\hat{f}$ and
$\hat{g}$ as well as f and g over their respective domains are
shown in Fig. 15(a) and (b). In Fig. 15(c), the outputs of
the plant as well as the identification model for an input
$u(k) = \sin(2\pi k/25) + \sin(2\pi k/10)$ are shown and are
seen to be indistinguishable.

4. Example 4: The same methods used for identifica-
tion of plants in examples 1–3 can be used when the un-
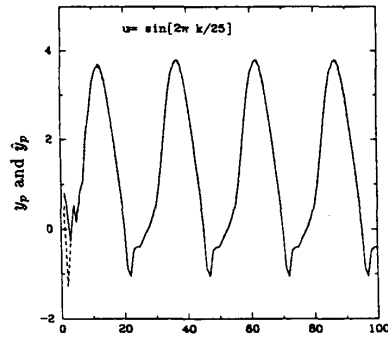known plants are known to belong to Model IV. In this

Fig. 14. Example 2: Outputs of the plant and the identification model.

example, the plant is assumed to be of the form

$$y_p(k + 1) = f[y_p(k), y_p(k - 1),$$
$$y_p(k - 2), u(k), u(k - 1)]$$

where the unknown function f has the form

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1 x_2 x_3 x_5(x_3 - 1) + x_4}{1 + x_3^2 + x_2^2}.$$

In the identification model, a neural network N belonging
to the class $\mathfrak{N}_{5,20,10,1}^3$ is used to approximate the function
f. Fig. 16 shows the output of the plant and the model
when the identification procedure was carried out for
$100\,000$ steps using a random input signal uniformly dis-
tributed in the interval $[-1, 1]$ and a step size of $\eta = 0.25$. As mentioned earlier, during the identification pro-
cess a series-parallel model is used, but after the identi-
fication process is terminated the performance of the
model is studied using a parallel model. In Fig. 16, the
input to the plant and the identified model is given by
$u(k) = \sin(2\pi k/250)$ for $k \leq 500$ and $u(k) = 0.8 \sin(2\pi k/250) + 0.2 \sin(2\pi k/25)$ for $k > 500$.

5. Example 5: In this example, it is shown that the
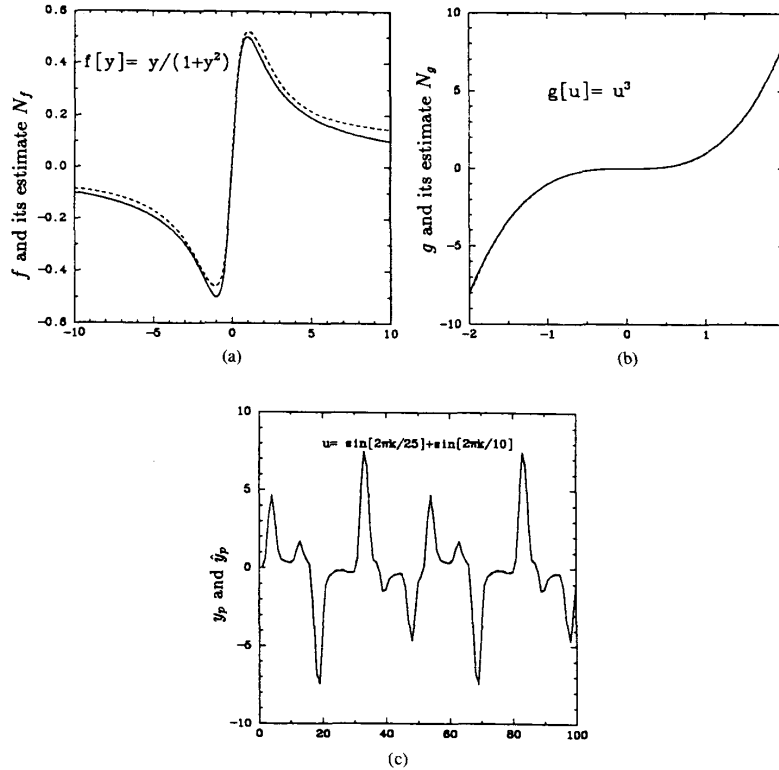same methods used to identify SISO plants can be used to

Fig. 15. Example 3: (a) Plots of the functions f and $\hat{f}$. (b) Plots of the functions g and $\hat{g}$. (c) Outputs of the plant and the identification model.
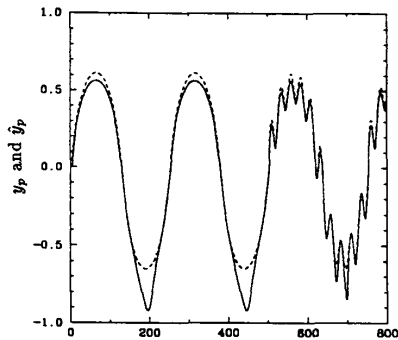

Fig. 16. Example 4: Identification of Model IV.

identify MIMO plants as well. The plant is described by the equations

$$\begin{bmatrix} y_{p1}(k+1) \\ y_{p2}(k+1) \end{bmatrix} = \begin{bmatrix} \dfrac{y_{p1}(k)}{1 + y_{p2}^2(k)} \\ \dfrac{y_{p1}(k)y_{p2}(k)}{1 + y_{p2}^2(k)} \end{bmatrix} + \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}. \quad (18)$$

This corresponds to the multivariable version of Model II. The series-parallel identification model consists of two neural networks N^1 and N^2 and is described by the equation

$$\begin{bmatrix} \hat{y}_{p1}(k+1) \\ \hat{y}_{p2}(k+1) \end{bmatrix} = \begin{bmatrix} N^1[y_{p1}(k), y_{p2}(k)] \\ N^2[y_{p1}(k), y_{p2}(k)] \end{bmatrix} + \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}.$$

The identification procedure was carried out for 100 000 time steps using a step size of $\eta = 0.1$ with random inputs $u_1(k)$ and $u_2(k)$ uniformly distributed in the interval $[-1, 1]$. The responses of the plant and the identification model for a vector input $[\sin(2\pi k/25), \cos(2\pi k/25)]^T$ are shown in Fig. 17.

Comment: In examples 1, 3, 4, and 5 the adjustment of the parameters was carried out by computing the gradient of $e^2(k)$ at instant k while in example 2 adjustments were based on the gradient of an error function evaluated over an interval of length 5. While from a theoretical point of view it is preferable to use a larger interval to define the error function, very little improvement was observed in the simulations. This accounts for the fact that in examples 3, 4, and 5 adjustments were based on the instantaneous rather than an average error signal.

6. Example 6: In examples 1-5, a series-parallel identification model was used and hence the parameters of the neural networks were adjusted using the static back-propagation method. In this example, we consider a simple
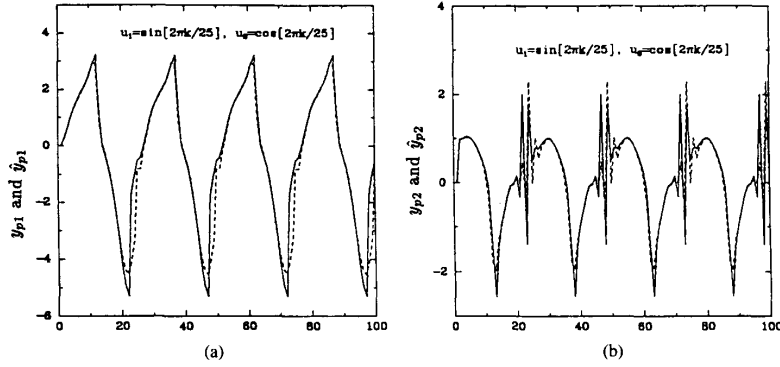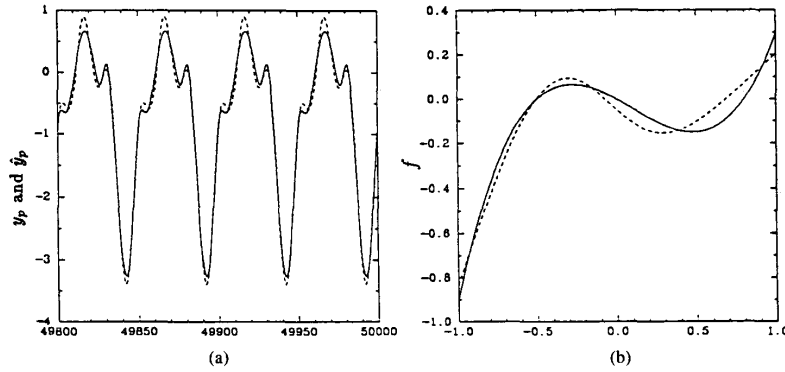
Fig. 17. Example 5: Responses of the plant and the identification model.

Fig. 18. Example 6: (a) Outputs of plant and identification model. (b) $f[u]$
and $N[u]$ for $u \in [-1, 1]$.

first order nonlinear system which is identified using the dynamic back-propagation method discussed in Section IV. The nonlinear plant is described by the difference equation

$$y_p(k + 1) = 0.8y_p(k) + f[u(k)]$$

where the function $f[u] = (u - 0.8)u(u + 0.5)$ is unknown. However, it is assumed that f can be approximated to the desired degree of accuracy by a multilayer neural network.

The identification model used is described by the difference equation

$$\hat{y}_p(k + 1) = 0.8\hat{y}_p(k) + N[u(k)]$$

and the neural network belonged to the class $\mathfrak{N}_{1,20,10,1}^3$. The model chosen corresponds to representation 1 in Section IV (refer to Fig. 6(a)). The objective is to adjust a total of 261 weights in the neural network so that $e(k) \triangleq \hat{y}_p(k) - y_p(k) \to 0$ asymptotically. Defining the performance criterion to be minimized as $J = (1/2T) \sum_{i=k-T+1}^{k} e^2(i)$, the partial derivative of J with respect to a weight θ_j in the neural network can be computed as $(\partial J / \partial \theta_j) = (1/T) \sum_{i=k-T+1}^{k} e(i) (\partial e(i) / \partial \theta_j)$. The quantity $(\partial e(i) / \partial \theta_j)$ can be computed in a dynamic fash-

ion using the method discussed in Section IV and used in the following rule to update θ:

$$k - T + 1 = \theta k - T + 1 - \eta \left[\frac{1}{T} \sum_{i=\theta(k+1)}^{k} e(i) \frac{\partial e(i)}{\partial \theta} \right]$$

where η is the step size in the gradient procedure.

Fig. 18(a) shows the outputs of the plant and the identification model when the weights in the neural network were adjusted after an interval of 10 time steps using a step size of $\eta = 0.01$. The input to the plant (and the model) was $u(k) = \sin(2\pi k/25)$. In Fig. 18(b), the function $f(u) = (u - 0.8)u(u + 0.5)$, as well as the function realized by the three layer neural network after 50 000 steps for $u \in [-1, 1]$, are shown. As seen from the figure, the neural network approximates the given function quite accurately.

VI. CONTROL OF DYNAMICAL SYSTEMS

As mentioned in Section II, for the sake of mathematical tractability most of the effort during the past two decades in the model reference adaptive control theory has been directed towards the control of linear time-invariant plants with unknown parameters. Much of the theoretical work in the late 1970's was aimed at determining adaptive

laws for the adjustment of the control parameter vector $\theta(k)$ which would result in stable overall systems. In 1980 [30]–[33], it was conclusively shown that for both discrete-time and continuous-time systems, such stable adaptive laws could be determined provided that some prior information concerning the plant transfer function was available. Since that time much of the research in the area has been directed towards determining conditions which assure the robustness of the overall system under different types of perturbations.

In contrast to the above, very little work has been reported on the global adaptive control of plants described by nonlinear difference or differential equations. It is in the control of such systems that we are primarily interested in this section. Since, in most problems, very little theory exists to guide the analysis, one of the aims is to indicate precisely how the nonlinear control problem differs from the linear one and the nature of the theoretical questions that have to be answered.

Algebraic and Analytic Parts of Adaptive Control Problems: In conventional adaptive control theory, two stages are generally distinguished in the adaptive process. In the first, referred to as the *algebraic part*, it is first shown that the controller has the necessary degrees of freedom to achieve the desired objective. More precisely, if some prior information regarding the plant is given, it is shown that a controller parameter vector θ^* exists for every value of the plant parameter vector p, so that the output of the controlled plant together with the controller approaches the output of the reference model asymptotically. The *analytic part* of the problem is then to determine stable adaptive laws for adjusting $\theta(k)$ so that $\lim_{k \to \infty} \theta(k) = \theta^*$ and the output error tends to zero.

Direct and Indirect Control: For over 20 years, two distinct approaches have been used [1] to control a plant adaptively. These are 1) *direct control* and 2) *indirect control*. In direct control, the parameters of the controller are directly adjusted to reduce some norm of the output error. In indirect control, the parameters of the plant are estimated as the elements of a vector $\hat{p}(k)$ at any instant k and the parameter vector $\theta(k)$ of the controller is chosen assuming that $\hat{p}(k)$ represents the true value p of the plant parameter vector. Even when the plant is assumed to be linear and time invariant, both direct and indirect adaptive control result in overall nonlinear systems. Figs. 19 and 20 represent the structure of the overall adaptive system using the two methods for the adaptive control of a linear time-invariant plant [1].

A. Adaptive Control of Nonlinear Systems Using Neural Networks

For a detailed treatment of direct and indirect control systems the reader is referred to [1]. The same approaches which have proved successful for linear plants can also be attempted when nonlinear plants have to be adaptively controlled. The structure used for the identification model as well as the controller are strongly motivated by those
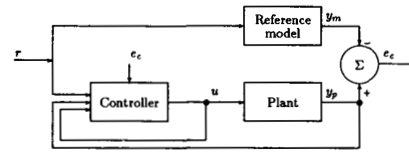

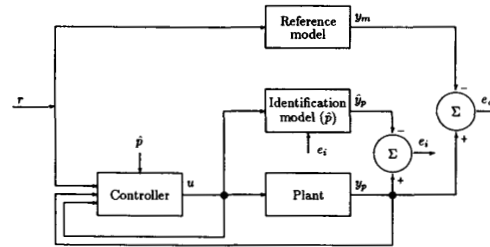Fig. 19. Direct adaptive control.

Fig. 20. Indirect adaptive control.

used in the linear case. However, in place of the linear gains, nonlinear neural networks are used.

Methods for identifying nonlinear plants using delayed values of both plant input and output were discussed in the previous section and Fig. 11 shows a general identification model. Fig. 21 shows a controller whose output is the control input to the plant and whose inputs are the delayed values of the plant input and output, respectively.

1. Indirect Control: At present, methods for directly adjusting the control parameters based on the output error (between the plant and the reference model outputs) are not available. This is because the unknown nonlinear plant in Fig. 21 lies between the controller and the output error e_c. Hence, until such methods are developed, adaptive control of nonlinear plants has to be carried out using indirect methods. This implies that the methods described in Section V have to be first used on line to identify the input–output behavior of the plant. Using the resulting identification model, which contains neural networks and linear dynamical elements as subsystems, the parameters of the controller are adjusted. This is shown in Fig. 22. It is this procedure of identification followed by control that is adopted in this section. Dynamic back propagation through a system consisting of only neural networks and linear dynamic elements was discussed in Section IV to determine the gradient of a performance index with respect to the adjustable parameters of a system. Since identification of the unknown plant is carried out using only neural networks and tapped delay lines, the identification model can be used to compute the partial derivatives of a performance index with respect to the controller parameters.

B. Simulation Results

The procedure adopted to adaptively control a nonlinear plant depends largely on the prior information available regarding the unknown plant. This includes knowl-
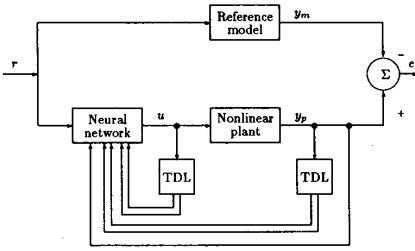
Fig. 21. Direct adaptive control of nonlinear plants using neural networks.
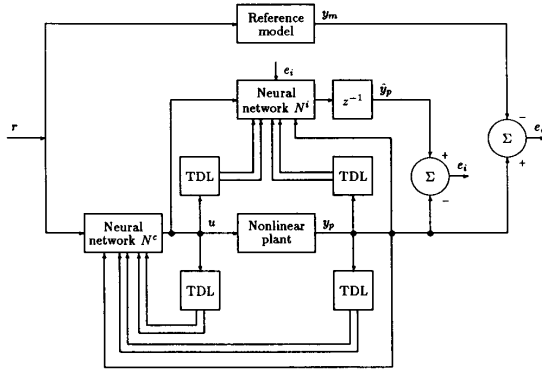

Fig. 22. Indirect adaptive control using neural networks.

edge of the number of equilibrium states of the unforced system, their stability properties, as well as the amplitude of the input for which the output is also bounded. For example, if the plant is known to have a bounded output for all inputs u belonging to some compact set $\mathcal{U}$, then the plant can be identified off line using the methods outlined in Section V. During identification, the weights in the identification model can be adjusted at every instant of time ($T_i = 1$) or at discrete time intervals ($T_i > 1$). Once the plant has been identified to the desired level of accuracy, control action can be initiated so that the output of the plant follows the output of a stable reference model. It must be emphasized that even if the plant has bounded outputs for bounded inputs, feedback control may result in unbounded solutions. Hence, for on-line control, identification and control must proceed simultaneously. The time intervals T_i and T_c, respectively, over which the identification and control parameters are to be updated have to be judiciously chosen in such a case.

Five examples, in which nonlinear plants are adaptively controlled, are included below and illustrate the ideas discussed earlier. As in the previous section, each example is chosen to emphasize a specific point.

1. Example 7: We consider here the problem of controlling the plant discussed in example 2 which is described by the difference equation

$$y_p(k + 1) = f[y_p(k), y_p(k - 1)] + u(k)$$

where the function

$$f[y_p(k), y_p(k - 1)]$$

$$= \frac{y_p(k)y_p(k - 1)[y_p(k) + 2.5]}{1 + y_p^2(k) + y_p^2(k - 1)} \qquad (19)$$

is assumed to be unknown. A reference model is described by the second-order difference equation

$$y_m(k + 1) = 0.6y_m(k) + 0.2y_m(k - 1) + r(k)$$

where $r(k)$ is a bounded reference input. If the output error $e_c(k)$ is defined as $e_c(k) \triangleq y_p(k) - y_m(k)$, the aim of control is to determine a bounded control input $u(k)$ such that $\lim_{k \to \infty} e_c(k) = 0$. If the function $f[.]$ in (19) is known, it follows directly that at stage k, $u(k)$ can be computed from a knowledge of $y_p(k)$ and its past values as

$$u(k) = -f[y_p(k), y_p(k - 1)] + 0.6y_p(k)$$
$$+ 0.2y_p(k - 1) + r(k) \qquad (20)$$

resulting in the error difference equation $e_c(k + 1) = 0.6e_c(k) + 0.2e_c(k - 1)$. Since the reference model is asymptotically stable, it follows that $\lim_{k \to \infty} e_c(k) = 0$ for arbitrary initial conditions. However, since $f[.]$ is unknown, it is estimated on line as $\hat{f}$ as discussed in example 2 using a neural network N and the series-parallel method.

The control input to the plant at any instant k is computed using $N[.]$ in place of f as

$$u(k) = -N[y_p(k), y_p(k - 1)] + 0.6y_p(k)$$
$$+ 0.2y_p(k - 1) + r(k). \qquad (21)$$

This results in the nonlinear difference equation

$$y_p(k + 1)$$
$$= f[y_p(k), y_p(k - 1)] - N[y_p(k), y_p(k - 1)]$$
$$+ 0.6y_p(k) + 0.2y_p(k - 1) + r(k) \qquad (22)$$

governing the behavior of the plant. The structure of the overall system is shown in Fig. 23.

In the first stage, the unknown plant was identified off line using random inputs as described in example 2. Following this, (21) was used to generate the control input. The response of the controlled system with a reference input $r(k) = \sin(2\pi k/25)$ is shown in Fig. 24(b).

In the second stage, both identification and control were implemented simultaneously using different values of T_i and T_c. The asymptotic response of the system when identification and control start at $k = 0$ with $T_i = T_c = 1$ is shown in Fig. 25(a). Since it is desirable to adjust the control parameters at a slower rate than the identification parameters, the experiment was repeated with $T_i = 1$ and $T_c = 3$ and is shown in Fig. 25(b). Since the identification process is not complete for small values of k, the control can be theoretically unstable. However, this was not observed in the simulations. If the control is initiated at time
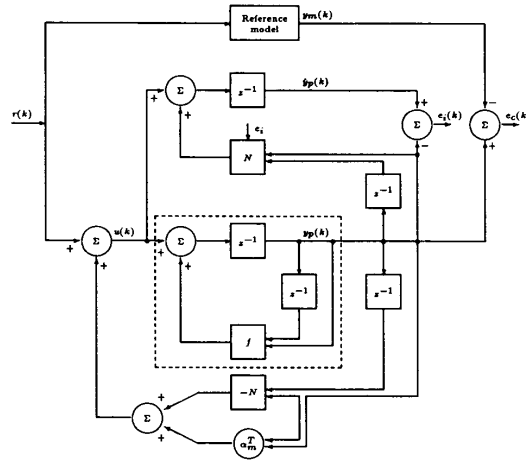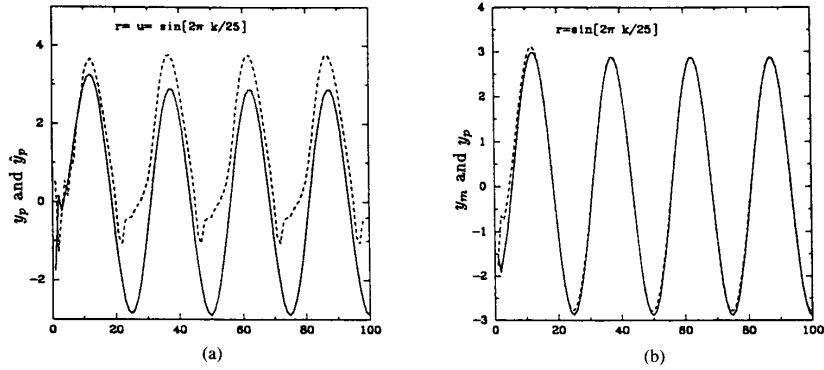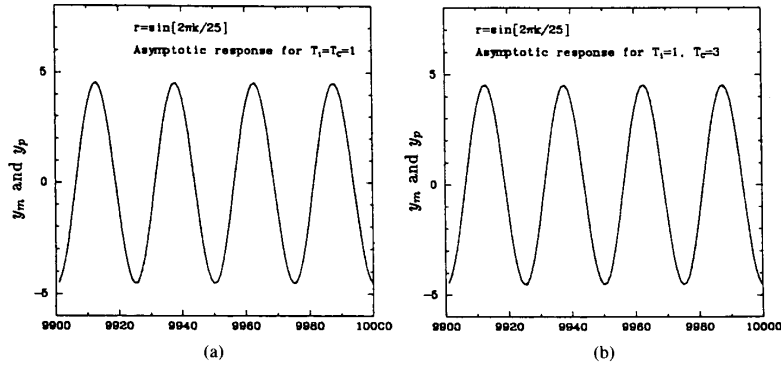
Fig. 23. Example 7: Structure of the overall system.

Fig. 24. Example 7: (a) Response for no control action. (b) Response for
$r = \sin(2\pi k/25)$ with control.

Fig. 25. Example 7: (a) Response when control is initiated at $k = 0$ with
$T_i = T_c = 1$. (b) Response when control is initiated at $k = 0$ and $T_i = 1$ and $T_c = 3$.

$k = 0$ using nominal values of the parameters of the neural network with $T_i = T_c = 10$, the output of the plant was seen to increase in an unbounded fashion as shown in Fig. 26.

The simulations reported above indicate that for stable and efficient on-line control, the identification must be sufficiently accurate before control action is initiated and hence T_i and T_c should be chosen with care.
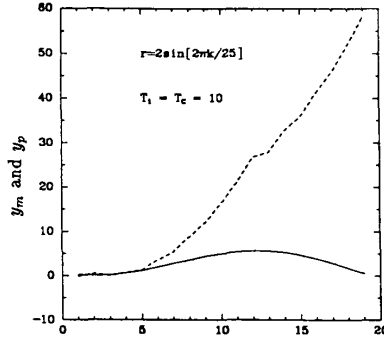
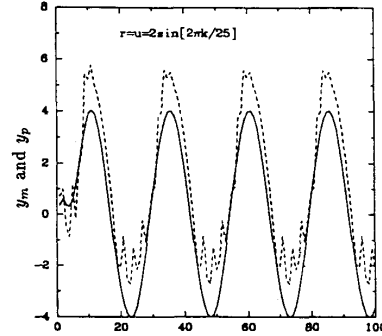Fig. 26. Example 7: Response when control is initiated at $k = 0$ with $T_i = T_c = 10$.

Fig. 27. Example 8: Responses of the reference model and the plant when no control action is taken.

2. Example 8: The unknown plant in this case corresponds to Model II and can be described by a difference equation of the form

$$y_p(k + 1)$$

$$= f[y_p(k), y_p(k - 1), \cdots, y_p(k - n + 1)]$$

$$+ \sum_{j=0}^{m-1} \beta_j u(k - j), \quad m \leq n.$$

It is assumed that the parameters β_j ($j = 0, 1, \cdots, m - 1$) are unknown, but that β_0 is nonzero with a known sign. The specific plant used in the simulation study was

$$y_p(k + 1) = \frac{5y_p(k)y_p(k - 1)}{1 + y_p^2(k) + y_p^2(k - 1) + y_p^2(k - 2)}$$

$$+ u(k) + 0.8u(k - 1). \quad (23)$$

The output of the stable reference model is described by

$$y_m(k + 1) = 0.32y_m(k) + 0.64y_m(k - 1)$$

$$- 0.5y_m(k - 2) + r(k)$$

where r is the uniformly bounded reference input. The responses of the reference model and the plant when $r(k) = u(k) = \sin(2\pi k/25)$ are shown in Fig. 27. While the output of the reference model is also a sinusoid of the same frequency, the response of the plant is seen to contain higher harmonics. It is assumed that sgn $\beta_0 = +1$ and that $\beta_0 \geq 0.1$. This enables a projection type algorithm to be used in the identification procedure so that the estimate $\hat{\beta}_0$ of β_0 satisfies the inequality $\hat{\beta}_0 \geq 0.1$. The control input any instant of time k is generated as

$$u(k) = \frac{1}{\hat{\beta}_0} \left[-\hat{f}_k[y_p(k), y_p(k - 1), y_p(k - 2)] \right.$$

$$- \hat{\beta}_1 u(k - 1) + 0.32y_p(k) + 0.64y_p(k - 1)$$

$$\left. - 0.5y_p(k - 2) + r(k) \right]. \quad (24)$$

In Fig. 28, the plant is identified over a period of 50 000 time steps using an input which is random and distributed

uniformly over the interval $[-2, 2]$. At the end of this interval, the control is implemented as given in (24). The response of the plant as well as the reference model are shown in Fig. 28. In Fig. 28(a) the reference input is $r(k) = \sin(2\pi k/25)$, while in Fig. 28(b) the reference input is $r(k) = \sin(2\pi k/25) + \sin(2\pi k/10)$. In both cases the control system is found to perform satisfactorily. Since the plant is identified over a sufficiently long time with a general input, the parameters $\hat{\beta}_0$ and $\hat{\beta}_1$ are found to converge to 1.005 and 0.8023, respectively, which are close to the true values of 1 and 0.8.

In Fig. 29 the response of the controlled plant to a reference input $r(k) = \sin(2\pi k/25)$ is shown, when identification and control are initiated at $k = 0$. Since the input is not sufficiently general, $\hat{\beta}_0(k)$ and $\hat{\beta}_1(k)$ tend to values 4.71 and 3.59 so that the asymptotic values of the parameter errors are large. In spite of this, the output error is seen to tend to zero for values of k greater than 9900. This example reveals that good control may be possible without good parameter identification.

3. Example 9: In this case, the plant is described by the same equation as in (23) with $0.8u(k - 1)$ replaced by $1.1u(k - 1)$ and the same procedure is adopted as in example 8 to generate the control input. It is found that the output error is bounded and even tends to zero while the control input grows in an unbounded fashion (Fig. 30). This is a phenomenon which is well known in adaptive control theory and arises due to the presence of zeros of the plant transfer function lying outside the unit circle. In the present context $u(k) + 1.1u(k - 1)$ can be zero even as $u(k) = (-1.1)^k$ tends to ∞ in an oscillatory fashion. The same phenomenon can also occur in systems where the dependence of y_p on u in nonlinear.

4. Example 10: The control of the nonlinear multivariable plant with two inputs and two outputs, discussed in example 5, is considered in this example and the plant is described by (18). The reference model is linear and is described by the difference equations

$$\begin{bmatrix} y_{m1}(k + 1) \\ y_{m2}(k + 1) \end{bmatrix} = \begin{bmatrix} 0.6 & 0.2 \\ 0.1 & -0.8 \end{bmatrix} \begin{bmatrix} y_{m1}(k) \\ y_{m2}(k) \end{bmatrix} + \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix}$$
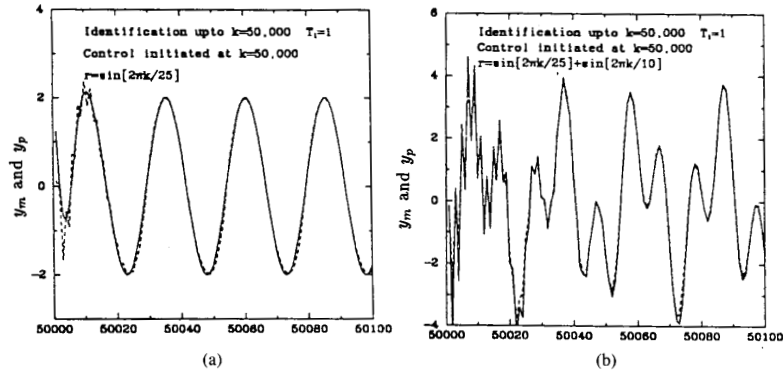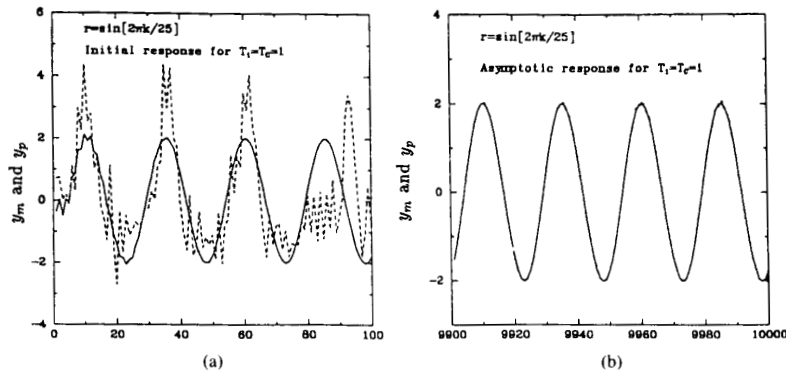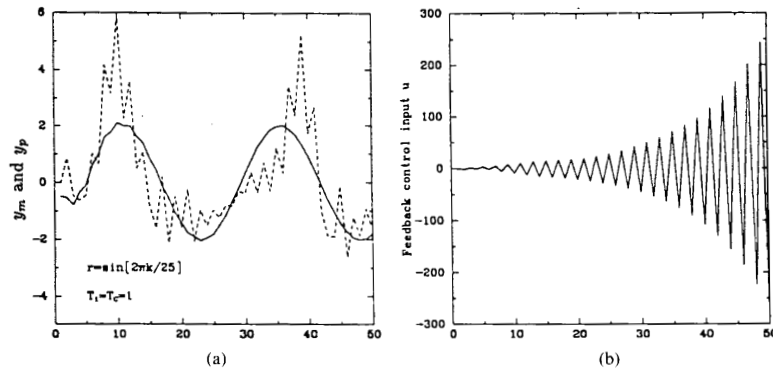
Fig. 28. Example 8: Identification followed by control.

Fig. 29. Example 8: Initial response when control action is taken at $k = 0$ with $T_i = T_c = 1$. (b) Asymptotic response with $T_i = T_c = 1$.

Fig. 30. Example 9: (a) Outputs of the reference model and the plant when control is initiated at $k = 0$. (b) The feedback control input u.

where r_1 and r_2 are bounded reference inputs. The plant is identified as in example 5 and control is initiated after the identification process is complete. The responses of the plant as compared to the reference model for the same inputs are shown in Fig. 31. The improvement in the responses, when the neural networks in the identification model are used to generate the control input to the plant

are evident from the figure. The outputs of the controlled plant and the reference model are shown and indicate that the output error is almost zero.

5. *Example 11:* In examples 7–10, the output of the plant depends linearly on the control input. This makes the computation of the latter relatively straightforward. In this example the plant is described by Model III and has
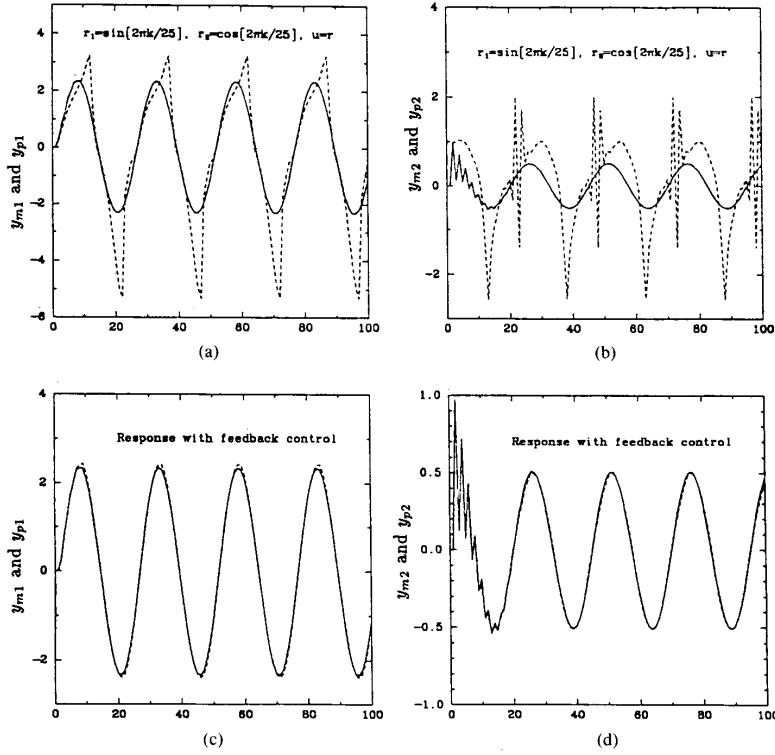
Fig. 31. Example 10: (a), (b) Outputs of the reference model and the plant
when no control action is taken. (c), (d) Outputs of the reference model
and the plant with feedback control.

the form

$$y_p(k + 1) = \frac{y_p(k)}{1 + y_p(k)^2} + u^3(k)$$

which was identified successfully in example 3. Choosing
the reference model as

$$y_m(k + 1) = 0.6y_m(k) + r(k)$$

the aim once again is to choose $u(k)$ so that $\lim_{k \to \infty}$
$|y_p(k) - y_m(k)| = 0$. If $f[y_p] = y_p/(1 + y_p^2)$ and $g[u]$
$= u^3$, the control input in this case is chosen as

$$u(k) = \widehat{g^{-1}}\left[-\hat{f}[y_p(k)] + 0.6y_p(k) + r(k)\right] \quad (25)$$

where $\hat{f}$ and $\widehat{g^{-1}}$ are the estimates of f and g^{-1}, respec-
tively. The estimates $\hat{f}$ and $\hat{g}$ are obtained as described
earlier using neural networks N_f and N_g. Since $\hat{g}[u]$ has
been realized as the output of a neural network N_g, the
weights of a neural network $N_c \in \mathfrak{N}_{1,20,10,1}^3$ (shown in Fig.
32) can be adjusted so that $N_g[N_c(r)] \approx r$ as $r(k)$ varies
over the interval $[-4, 4]$. The range $[-4, 4]$ was chosen
for $r(k)$ since this assures that the input to the identifi-
cation model varies over the same range for which the
estimates $\hat{f}$ and $\hat{g}$ are valid. In Fig. 33 $N_g[N_c(r)]$ is plot-
ted against r and is seen to be unity over the entire range.
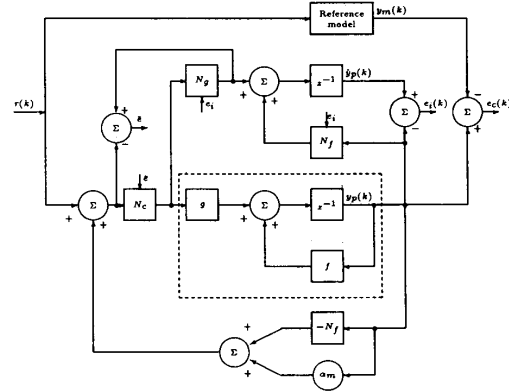

Fig. 32. Example 11: Structure of the overall system.

The determination of N_c was carried out over 25 000 time
steps using a random input uniformly distributed in the
interval $[-4, 4]$ and a step size of $\eta = 0.01$. Since the
plant nonlinearities f and g as well as g^{-1} have been es-
timated using neural networks N_f, N_g, and N_c, respec-
tively, the control input to the plant can be determined
using (25). The output of the plant to a reference input
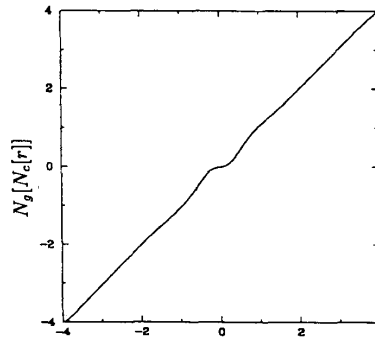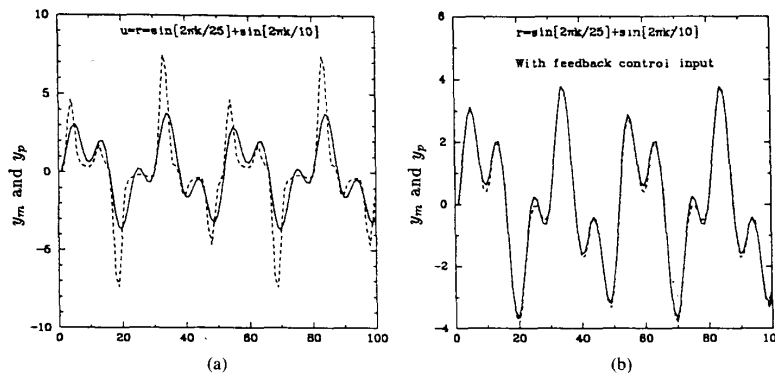$r(k) = \sin(2\pi k/25) + \sin(2\pi k/10)$ is shown in Fig.

Fig. 33. Example 11: Plot of the function $N_g[N_c(r)]$.

Fig. 34. Example 11: (a) Outputs of the reference model and plant without a feedback controller. (b) Outputs of the reference model and plant with a feedback controller.

34(a) when a feedback controller is not used; the response with a controller is shown in Fig. 34(b). The response in Fig. 34(b) is identical to that of the reference model and is almost indistinguishable from it. Hence, from this example we conclude that it may be possible in some cases to generate a control input to an unknown plant so that almost perfect model following is achieved.

VII. COMMENTS AND CONCLUSIONS

In this paper models for the identification and control of nonlinear dynamic systems are suggested. These models, which include multilayer neural networks as well as linear dynamics, can be viewed as generalized neural networks. In the specific models given, the delayed values of relevant signals in the system are used as inputs to multilayer neural networks. Methods for the adjustment of parameters in generalized neural networks are treated and the concept of dynamic back propagation is introduced in this context to generate partial derivatives of a performance index with respect to adjustable parameters on line. However, in many identifiers and controllers it is shown that by using a series-parallel model, the gradient can be obtained with the simpler static back-propagation method.

The simulation studies on low order nonlinear dynamic systems reveal that identification and control using the methods suggested can be very effective. There is every reason to believe that the same methods can also be used successfully for the identification and control of multivariable systems of higher dimensions. Hence, they should find wide application in many areas of applied science.

Several assumptions were made concerning the plant characteristics in the simulation studies to achieve satisfactory identification and control. For example, in all cases the plant was assumed to have bounded outputs for the class of inputs specified. An obvious and important extension of the methods in the future will be to the control of unstable systems in some compact domain in the state space. All the plants were also assumed to be of relative degree unity (i.e., input at k affects the output at $k + 1$), minimum phase (i.e., no unbounded input lies in the null space of the operator representing the plant) and Models II and III used in control problems assumed that inverses of operators existed and could be approximated. Future work will attempt to relax some or all of these assumptions. Further, in all cases the gradient method is used exclusively for the adjustment of the parameters of the plant. Since it is well known that such methods can

lead to instability for large values of the step size η, it is essential that efforts be directed towards determining stable adaptive laws for adjusting the parameters. Such work is currently in progress.

A number of assumptions were made throughout the paper regarding the plant to be controlled for the methods to prove successful. These include stability properties of recurrent networks with multilayer neural networks in the forward path, controllability, observability, and identifiability of the models suggested as well as the existence of nonlinear controllers to match the response of the reference model. At the present stage of development of nonlinear control theory, few constructive methods exist for checking the validity of these assumptions in the context of general nonlinear systems. However, the fact that we are dealing with special classes of systems represented by generalized neural networks should make the development of such methods more tractable. Hence, concurrent theoretical research in these areas is needed to justify the models suggested in this paper.

ACKNOWLEDGMENT

REFERENCES

[1] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[2] D. J. Burr, "Experiments on neural net recognition of spoken and written text," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 7, pp. 1162–1168, July 1988.
[3] R. P. Gorman and T. J. Sejnowski, "Learned classification of sonar targets using a massively parallel network," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 7, pp. 1135–1140, July 1988.
[4] T. J. Sejnowski and C. R. Rosenberg, "Parallel networks that learn to pronounce English text," *Complex Syst.*, vol. 1, pp. 145–168, 1987.
[5] B. Widrow, R. G. Winter, and R. A. Baxter, "Layered neural nets for pattern recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 7, pp. 1109–1118, July 1988.
[6] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci., U.S.*, vol. 79, pp. 2554–2558, Apr. 1982.
[7] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biolog. Cybern.*, vol. 52, pp. 141–152, 1985.
[8] D. W. Tank and J. J. Hopfield, "Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Syst., Man, Cybern.*, vol. CAS-33, no. 5, pp. 533–541, May 1986.
[9] H. Rauch and T. Winarske, "Neural networks for routing communication traffic," *IEEE Control Syst. Mag.*, vol. 8, no. 2, pp. 26–30, Apr. 1988.
[10] N. B. Haaser and J. A. Sullivan, *Real Analysis*. New York: Van Nostrand Reinhold, 1971.
[11] P. G. Gallman and K. S. Narendra, "Identification of nonlinear systems using a Uryson model," Becton Center, Yale University, New Haven, CT, tech. rep. CT-38, Apr. 1971; also *Automatica*, Nov. 1976.
[12] L. Ljung and T. Söderstrom, *Theory and Practice of Recursive Identification*. Cambridge, MA: M.I.T. Press, 1985.
[13] S. N. Singh and W. J. Rugh, "Decoupling in a class of nonlinear systems by state variable feedback," *Trans. ASME*, vol. 94, pp. 323–324, 1972.
[14] E. Freund, "The structure of decoupled nonlinear systems," *Int. J. Contr.*, vol. 21, pp. 651–654, 1975.

[15] A. Isidori, A. J. Krener, C. Gori Giorgi, and S. Monaco, "Nonlinear decoupling via feedback: A differential geometric approach," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 331–345, 1981.
[16] S. S. Sastry and A. Isidori, "Adaptive control of linearizable systems," *IEEE Trans. Automat. Contr.*, vol. 34, no. 11, pp. 1123–1131, Nov. 1989.
[17] F. J. Pineda, "Generalization of back propagation to recurrent networks," *Phys. Rev. Lett.*, vol. 59, no. 19, pp. 2229–2232, Nov. 1987.
[18] J. H. Li, A. N. Michel and W. Porod, "Qualitative analysis and synthesis of a class of neural networks," *IEEE Trans. Circuits Syst.*, vol. 35, no. 8, pp. 976–986, Aug. 1988.
[19] B. Kosko, "Bidirectional associative memories," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, no. 1, pp. 49–60, Jan./Feb. 1988.
[20] K. S. Narendra and K. Parthasarathy, "Neural networks and dynamical systems. Part I: A gradient approach to Hopfield networks," Center Syst. Sci., Dept. Electrical Eng., Yale University, New Haven, CT, tech. rep. 8820, Oct. 1988.
[21] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Dept. Economics, University of California, San Diego, CA, discussion pap., Dept. Economics, June 1988.
[22] K. S. Narendra and L. E. McBride, Jr., "Multiparameter self-optimizing system using correlation techniques," *IEEE Trans. Automat. Contr.*, vol. AC-9, pp. 31–38, 1964.
[23] P. V. Kokotovic, "Method of sensitivity points in the investigation and optimization of linear control systems," *Automat. Remote Contr.*, vol. 25, pp. 1512–1518, 1964.
[24] L. E. McBridge, Jr. and K. S. Narendra, "Optimization of time-varying systems," *IEEE Trans. Automat. Contr.*, vol. AC-10, no. 3, pp. 289–294, 1965.
[25] J. B. Cruz, Jr., Ed., *System Sensitivity Analysis, Benchmark Papers in Electrical Engineering and Computer Science*. Stroudsburg, PA: Dowden, Hutchinson and Ross, 1973.
[26] K. S. Narendra and K. Parthasarathy, "A diagrammatic representation of back propagation," Center for Syst. Sci., Dept. of Electrical Eng., Yale University, New Haven, CT, tech. rep. 8815, Aug. 1988.
[27] K. S. Narendra and K. Parthasarathy, "Back propagation in dynamical systems containing neural networks," Center Syst. Sci., Dept. Electrical Eng., Yale University, New Haven, CT, tech. rep. 8905, Mar. 1989.
[28] K. S. Narendra and K. Parthasarathy, "Neural networks and dynamical systems. Part II: Identification," Center for Syst. Sci., Dept. of Electrical Eng., Yale University, New Haven, CT, tech. rep. 8902, Feb. 1989.
[29] K. S. Narendra and K. Parthasarathy, "Neural networks and dynamical systems. Part III: Control," Center Syst. Sci., Dept. Electrical Eng., Yale University, New Haven, CT, tech. rep. 8909, May 1989.
[30] K. S. Narendra, Y. H. Lin, and L. S. Valavani, "Stable adaptive controller design—Part II: Proof of stability," *IEEE Trans. Automat. Contr.*, vol. 25, pp. 440–448, June 1980.
[31] A. S. Morse, "Global stability of parameter adaptive systems," *IEEE Trans. Automat. Contr.*, vol. 25, pp. 433–439, June 1980.
[32] G. C. Goodwin, P. J. Ramadge, and P. E. Caines, "Discrete time multivariable adaptive control," *IEEE Trans. Automat. Contr.*, vol. 25, pp. 449–456, June 1980.
[33] K. S. Narendra and Y. H. Lin, "Stable discrete adaptive control," *IEEE Trans. Automat. Contr.*, vol. 25, vol. 456–461, June 1980.

*

Kumpati S. Narendra (S'55-M'60-SM'63-F'79) received the B.S. degree with honors in electrical engineering from Madras University in 1954 and the M.S. and Ph.D. degrees from Harvard University in 1955 and 1959, respectively.

From 1961 to 1965 he was an Assistant Professor in the Division of Applied Physics at Harvard. He joined the Department of Engineering and Applied Science at Yale in 1965 and was made Professor in 1968. He was the Chairman of the Department of Electrical Engineering from 1984

to 1987 and currently he is the Director of the Center for Systems Science at Yale. He is the author of numerous technical publications in the area of systems theory. He is the author of the books *Frequency Domain Criteria for Absolute Stability* (coauthor J. H. Taylor) published by Academic Press in 1973; *Stable Adaptive Systems* (coauthor A. M. Annaswamy); and *Learning Automata—An Introduction* (coauthor M. A. L. Thathachar) published by Prentice-Hall in 1989. He is also the editor of the books *Applications of Adaptive Control* (coeditor R. V. Monopoli) published by Academic Press in 1980 and *Adaptive and Learning Systems* (Plenum Press, 1986). He is currently editing a reprint volume entitled *Recent Advances in Adaptive Control* (coeditors R. Ortega and P. Dorato) which will be published by the IEEE Press. His research interests are in the areas of stability theory, adaptive control, learning automata and the control of complex systems using neural networks. He has served on various national and international technical committees as well as several editorial boards of technical journals.

Dr. Narendra is a member of Sigma Xi and the American Mathematical Society and a Fellow of the American Association for the Advancement of Science and the IEE (U.K.). He was the recipient of the 1972 Franklin V. Taylor Award of the IEEE Systems, Man, and Cybernetics Society and the George S. Axelby Best Paper Award of the IEEE Control Systems Society in 1988.

*

Kannan Parthasarathy was born in India on August 31, 1964. He received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, in 1986 and the M.S. degree and M.Phil. degree in electrical engineering from Yale University in 1987 and 1988, respectively. At present he is a doctoral candidate at Yale University. His research interests include adaptive and learning systems, neural networks and their application to adaptive control problems.

Mr. Parthasarathy is the recipient of the Connecticut High Technology Graduate Scholarship for the year 1989–1990.